

Univerzita Karlova

Pedagogická fakulta

Katedra informačních technologií a technické výchovy

DIPLOMOVÁ PRÁCE

Metodika výuky PHP na střední škole

Methodology of teaching PHP in high school

Libor Češka

Vedoucí práce: PhDr. Josef Procházka, Ph.D.

Studijní program: Učitelství informačních a komunikačních technologií pro 2. stupeň
základní školy a střední školy

Odevzdáním této diplomové práce na téma Metodika výuky PHP na střední škole potvrzuji, že jsem ji vypracoval pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Prohlašuji, že jsem při její tvorbě nepoužil nástrojů umělé inteligence jiným způsobem, než je uvedeno ve vyjádření, které je součástí textu práce. Dále potvrzuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

Ostrov 2.12.2024

Rád bych vyjádřil poděkování vedoucímu mé diplomové práce PhDr. Josefu Procházkovi, Ph.D., za jeho odborné vedení, cenné rady a neocenitelnou podporu, kterou mi poskytoval během procesu tvorby této práce.

Děkuji mu za jeho čas, který mi věnoval, a za jeho neocenitelnou pomoc, bez níž by tato práce nebyla možná.

ABSTRAKT

Diplomová práce se zabývá zpracováním metodiky výuky skriptovacího jazyka PHP na střední škole, vycházející z teoretických poznatků obecných a oborových didaktik.

V teoretické části obsahuje rozbor skriptovacích jazyků, možné použití vybraného skriptovacího jazyka na straně serveru pro výuku na střední škole. Dále jsou popsány opory výuky skriptování v kurikulárních dokumentech, jako jsou rámcový vzdělávací program pro střední odborné vzdělávání, včetně jeho poslední aktualizace z roku 2023, a Strategie 2030+. Práce také zahrnuje skutečné zařazení výuky skriptovacího jazyka PHP do školního vzdělávacího programu na vybrané střední škole.

Praktickou část diplomové práce tvoří návrh metodiky zaměřené na výuku skriptovacího jazyka PHP, která je začleněna do příslušného předmětu dle školního vzdělávacího programu na vybrané střední škole. Součástí metodiky je i série graduujících navazujících úloh, které podporují postupný rozvoj dovedností studentů. Tato metodika je následně ověřena v reálném vzdělávacím procesu.

Práce dále popisuje ověření metodiky na Střední průmyslové škole Ostrov, p.o., v období od prosince 2022 do března 2023. Metodika byla nasazena a testována ve dvou paralelních skupinách žáků třetího ročníku oboru 18-20-M/01 Informační technologie.

V závěru práce je provedeno vyhodnocení nasazení metodiky v praxi, shrnutí získaných poznatků z jejího nasazení v reálném vzdělávacím procesu a formulování závěrů a doporučení pro budoucí nasazení navržené metodiky v praxi.

KLÍČOVÁ SLOVA

Skriptovací jazyky, PHP, internetové stránky, střední škola, programování, metodika

ABSTRACT

The thesis deals with the elaboration of the methodology of teaching the PHP scripting language at secondary school, based on the theoretical knowledge of general and specialized didactics.

In the theoretical part, it contains an analysis of scripting languages, possible use of the selected server-side scripting language for teaching in high school. Furthermore, the underpinnings of scripting instruction in curricular documents such as the Framework Curriculum for Secondary Vocational Education, including its latest update in 2023, and the 2030+ Strategy are described. The paper also covers the actual inclusion of PHP scripting language teaching in the school curriculum in a selected secondary school.

The practical part of the thesis consists of the proposal of a methodology aimed at teaching the PHP scripting language, which is incorporated into the relevant subject according to the school curriculum at the selected secondary school. The methodology includes a series of gradual follow-up tasks that support the gradual development of students' skills. The methodology is then tested in a real educational process.

The thesis further describes the validation of the methodology at the Secondary Industrial School Ostrov, p.o., in the period from December 2022 to March 2023. The methodology was deployed and tested in two parallel groups of third year students in the field 18-20-M/01 Information Technology.

The paper concludes with an evaluation of the deployment of the methodology in practice, a summary of the findings obtained from its deployment in a real educational process and the formulation of conclusions and recommendations for future deployment of the proposed methodology in practice.

KEYWORDS

Scripting languages, PHP, websites, high school, programming, methodology

Obsah

Úvod	8
1 Skriptování a jeho opora v kurikulárních dokumentech.....	9
1.1 Opora výuky skriptování v kurikulárních dokumentech	10
1.1.1 Kurikulární dokumenty v ČR	10
2 Zařazení do ŠVP na konkrétní škole	17
2.1 RVP pro obor 18-20-M/01 Informační technologie	17
2.2 ŠVP pro obor 18-20M/01 Informační technologie na SPŠ Ostrov, p.o.....	18
2.2.1 Prostor pro značkovací a skriptovací jazyky	18
3 Výběr konkrétní platformy pro výuku.....	21
3.1 Důvody pro výuku PHP ve školách	21
3.2 Jazyk PHP	21
3.3 Nutné technické vybavení.....	23
3.4 Technické vybavení pro výuku PHP na školách.....	23
3.4.1 Řešení pomocí školního serveru.....	24
3.4.2 Řešení pomocí poskytovatele hostingových služeb	24
3.4.3 Řešení pomocí lokálních webových serverů	25
3.4.4 Online prostředí pro interpretaci skriptů – W3Schools.com	28
3.4.5 Výuka na Střední průmyslové škole Ostrov, p.o.	28
4 Metodika výuky skriptovacího jazyka PHP	31
4.1 Požadavky na začátku výuky PHP.....	31
4.2 Očekávané výstupy	31
4.3 Výběr vhodných tematických celků	32
5 Metodické listy	34
5.1 Materiální a technické zabezpečení	34

5.2	Tematický celek III. Programování v PHP	34
5.2.1	Prostředí PHP, základní příkazy, proměnné, datové typy	35
5.2.2	Příkazy větvení a cykly.....	43
5.2.3	Funkce a globální proměnné.....	49
5.2.4	Pole	60
5.2.5	Textové soubory, řetězce	67
5.2.6	Formuláře	77
5.2.7	Hlavičky	87
5.2.8	Relace – session.....	90
6	Ověření metodiky	95
6.1	Očekávané výsledky ověření	95
6.2	Průběh výuky pomocí navržené metodiky.....	97
6.2.1	Prostředí PHP, základní příkazy, proměnné, datové typy	97
6.2.2	Příkazy větvení a cykly.....	98
6.2.3	Funkce a globální proměnné.....	98
6.2.4	Pole	99
6.2.5	Textové soubory, řetězce	100
6.2.6	Formuláře	101
6.2.7	Hlavičky	102
6.2.8	Session.....	103
6.2.9	Závěrečná hodina a dotazníkové šetření.....	103
6.3	Závěrečný dotazník.....	104
6.4	Hodnocení výuky na SPŠ Ostrov.....	108
6.5	Vyhodnocení ověření.....	111
	Závěr.....	113

Seznam použitých zkratek	115
Seznam použitých informačních zdrojů	117
Vyjádření k využití nástrojů umělé inteligence	120
Seznam příloh	121

Úvod

Téma své diplomové práce jsem si zvolil nejen na základě mé dlouholeté praxe v oblasti tvorby dynamických webových stránek pomocí jazyka PHP, ve které se pohybuji od roku 2002, ale také díky mému pedagogickému působení na střední škole. Od roku 2009 jsem začal pracovat jako učitel na částečný úvazek na nyní již zaniklé Střední průmyslové škole Loket, kde jsem se v následujícím roce stal kmenovým učitelem a od roku 2010 jsem zahájil výuku předmětu spojeným s tvorbou dynamických webových stránek. V roce 2018 jsem po zániku této školy přešel na Střední průmyslovou školu Ostrov, p.o., kde pokračuji ve výuce této problematiky.

Diplomová práce si klade za cíl navrhnout metodiku výuky jazyka PHP na střední škole a ověřit tuto navrženou metodiku spolu se sérií navazujících úloh v příslušném předmětu dle platného školního vzdělávacího programu na vybrané střední škole.

Dílními cíli je zmapovat prostor výuky skriptovacích jazyků na střední škole v rámci platných kurikulárních dokumentů ČR a na základě vybraného školního vzdělávacího programu vybrané školy provést analýzu nasazení výuky jazyka PHP a stanovit témata výuky.

Bude provedena analýza klíčových kurikulárních dokumentů, jako jsou rámcový vzdělávací program pro střední odborné vzdělávání, Strategie 2030+, rámcový vzdělávací program pro obor 18-20 M/01 Informační technologie a školní vzdělávací program pro tento obor na vybrané střední škole.

Na základě analýzy školního vzdělávacího programu bude navržena metodika výuky jazyka PHP, jejíž součástí bude i série graduujících navazujících úloh.

Navržená metodika, včetně série navazujících úloh, bude následně ověřena praktickým nasazením ve výuce. Součástí bude také vyhodnocení průběhu výuky a závěrečná zpětná vazba od žáků prostřednictvím dotazníkového šetření.

Získané informace budou analyzovány a využity k formulování závěrů a doporučení pro případné rozšíření metodiky do praxe.

1 Skriptování a jeho opora v kurikulárních dokumentech

Skripty jsou soubory instrukcí, které řídí provádění určitých akcí nebo úkolů v softwaru nebo operačním systému. Skripty se často používají pro automatizaci úloh, jednoduché úpravy dat, správu souborů a další činnosti.

Skriptování označuje samotný proces psaní skriptů ve skriptovacím jazyce.

„Skriptovací jazyk je interpretovaný programovací jazyk, který je navržen s ohledem na snadné ovládnutí jazyka a rychlý a pohodlný vývoj programů. Obecně přijímaná definice, jaké vlastnosti musí mít skriptovací jazyk, neexistuje.

Skutečný rozmach je spojen především s rozšířením internetu. Základní důvody jsou v zásadě dva:

- nezávislost na platformě a
- relativně jednoduchá konstrukce interpretu.“ (Durdilová, 2012)

Skriptovací jazyky typicky jsou:

- Interpretované – skriptovací jazyky jsou obvykle interpretovány, což znamená, že instrukce jsou vykonávány postupně interpretem, aniž by byly předem kompilovány do spustitelného souboru. Tím se zjednodušuje jejich vývoj a úpravy.
- Dynamicky typované – skriptovací jazyky obvykle nemají přísné definice typů proměnných, což umožňuje flexibilnější práci s daty. Proměnné mohou mít různé typy a jejich typ se určuje až za běhu programu.
- Snadná syntaxe – skriptovací jazyky mají často jednoduchou a intuitivní syntaxi, která je snadno srozumitelná i pro začátečníky. Tím se zjednodušuje jejich psaní a čtení.
- Silná podpora pro manipulaci se soubory, textem a daty – skriptovací jazyky nabízejí bohatou sadu funkcí a knihoven pro práci se soubory, textovými řetězci, regulárními výrazy a dalšími datovými strukturami.
- Rychlý vývoj – skriptovací jazyky umožňují rychlý vývoj a testování kódu, protože nemusí být kompilovány před každým spuštěním. To usnadňuje iterativní vývoj a experimentování s různými přístupy.

Skriptovací jazyky neslouží jen pro vývoj webových aplikací, jejich použitelnost je v různých oblastech od operačních systémů, správy a analýzy dat či automatizaci. (Beneš, 2024)

Nejznámější skriptovací jazyky jsou například Python, Perl, Ruby, JavaScript, PHP, ASP, PowerShell a také procedurální nástavba jazyka SQL – PL/SQL. (Stack Overflow, 2024)

1.1 Opora výuky skriptování v kurikulárních dokumentech

1.1.1 Kurikulární dokumenty v ČR

V České republice jsou hlavními kurikulárními dokumenty pro středoškolské vzdělávání:

- Rámcové vzdělávací programy (RVP)
- Strategie 2030+
- Školní vzdělávací programy (ŠVP)

Rámcové vzdělávací programy

RVP byly do vzdělávání v České republice zavedeny zákonem č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon).

Tyto programy stanovují základní požadavky na vzdělávání v různých oborech a jsou závazné pro všechny školy. Existují různé RVP pro různé typy středních škol, například:

- RVP pro gymnázia
- RVP pro střední odborné školy (SOV)
- RVP pro lycea

Rámcové vzdělávací programy tvoří obecně závazný rámec pro tvorbu školních vzdělávacích programů škol všech oborů vzdělávání v předškolním, základním, základním uměleckém, jazykovém a středním vzdělávání.

Hlavní cíle RVP SOV

Rozvoj klíčových kompetencí:

- RVP se zaměřují na rozvoj kompetencí, které jsou nezbytné pro osobní a profesní život, jako jsou kompetence k učení, řešení problémů, komunikativní, sociální a personální kompetence.

Propojení teorie s praxí:

- Praktické využití získaných vědomostí a dovedností v reálném životě a na trhu práce.

Podpora celoživotního učení:

- RVP podporují koncept celoživotního učení, což znamená, že vzdělávání by mělo být kontinuální proces, který pokračuje i po ukončení formálního vzdělávání.

Přizpůsobení aktuálním potřebám trhu práce:

- Programy jsou pravidelně aktualizovány (poslední aktualizace je z roku 2023), aby odpovídaly nejnovějším poznatkům a potřebám trhu práce. Školy však mají možnost si své ŠVP blíže přizpůsobit aktuálním potřebám na trhu práce ve svém regionu.

Struktura RVP SOV

RVP SOV jsou rozděleny do několika částí, z nichž hlavní jsou:

- Kompetence absolventa – popisuje, jaké znalosti, dovednosti a kompetence by měl absolvent daného oboru mít.
- Kurikulární rámce pro jednotlivé oblasti – obsahuje povinné vzdělávací oblasti, jako jsou jazykové vzdělávání a komunikace, matematické vzdělávání, Informatické vzdělávání, odborné vzdělávání a další specifické oblasti podle oboru včetně minimálního počtu hodin, které musí být věnovány jednotlivým vzdělávacím oblastem během studia.
- Zásady tvorby školního vzdělávacího programu (ŠVP) – obecné zásady tvorby ŠVP, tvorba ŠVP pro různé formy studia a další potřebné informace pro případné změny ŠVP.
- Základní podmínky pro uskutečnění vzdělávacího programu – určuje materiální, personální, organizační podmínky a také podmínky bezpečnosti práce a ochrany zdraví při vzdělávacích činnostech, které musí škola splňovat.
- Vzdělávání žáků se speciálními vzdělávacími potřebami (SVP) a žáků nadaných – specifikace podpůrných opatření, vytváření individuálních vzdělávacích plánů (IVP)

pro žáky se SVP, ale také pro nadané žáky včetně systému péče o žáky se SVP a nadané žáky.

(Národní pedagogický institut České republiky, 2024)

Aktualizace a změny

Poslední významné změny v RVP SOV byly provedeny v roce 2023, kdy byla aktualizována oblast vzdělávání v informačních a komunikačních technologiích (ICT). Tyto změny reflektují dynamický vývoj v oblasti ICT a zahrnují nové pojetí informatického vzdělávání.

Školy mají povinnost upravit své ŠVP v souladu s aktualizovanými RVP plošně pro všechny ročníky oborů vzdělání kategorie dosaženého vzdělání J, E, H, M včetně konzervatoří a lyceí, L0, L5 a zahájit podle nich vzdělávání nejpozději od 1. září 2025. Tento proces zahrnuje nejen úpravu obsahu vzdělávání, ale také zajištění odpovídajících podmínek pro jeho realizaci. (MŠMT a NPI České republiky, 2024)

Výuka značkovacích a skriptovacích jazyků v RVP

Výuku značkovacích jazyků lze zařadit do výuky několika povinných či volitelných předmětů.

Informatika

Nově v rámci předmětu Informatika mohou školy zahrnout výuku značkovacích jazyků, jako je HTML, a skriptovacích jazyků, jako je Python nebo JavaScript. Tyto dovednosti jsou součástí digitálních kompetencí, které jsou nyní povinné.

Projektová výuka

Školy mohou využít projektovou výuku, kde žáci pracují na konkrétních projektech, které vyžadují použití značkovacích a skriptovacích jazyků. Takové projekty mohou být součástí přímo předmětu Informatika, případně jiných předmětů, kde je možné integrovat digitální technologie.

Volitelné předměty

Školy mohou nabízet volitelné předměty zaměřené na programování a webový vývoj, kde se žáci mohou hlouběji věnovat značkovacím a skriptovacím jazykům.

Možnosti aplikace výuky skriptovacích a značkovacích jazyků

Algoritmizace a programování

Tento tematický okruh zahrnuje výuku základních i pokročilých programovacích technik. Skriptovací jazyky, jako je Python či JavaScript, mohou být pro svou jednodušší syntaxi využity k výuce základů algoritmizace, dále také k tvorbě skriptů pro automatizaci úloh a v neposlední řadě k vývoji webových aplikací.

Webové technologie

V rámci tohoto okruhu se žáci učí značkové jazyky jako HTML a CSS, které jsou základem pro tvorbu webových stránek. Dále se mohou učit JavaScript pro interaktivitu a dynamiku webových stránek či skriptovací jazyky PHP, ASP, nebo ASP.NET, vykonávané na straně serveru.

Databázové systémy

Skriptovací jazyky mohou být využity pro práci s databázemi, například pomocí SQL skriptů pro správu a manipulaci s daty. Žáci se mohou učit, jak integrovat databáze s webovými aplikacemi.

Projektová výuka

Žáci mohou pracovat na reálných projektech, které vyžadují použití značkovacích a skriptovacích jazyků. Projekty mohou zahrnovat vývoj webových aplikací, automatizaci procesů či tvorbu interaktivních prezentací.

(MŠMT, 2020)

Strategie 2030+

Strategie 2030+ je další klíčový dokument pro rozvoj vzdělávací soustavy České republiky na období 2020–2030. Jeho cílem je modernizovat vzdělávací systém, aby lépe odpovídal potřebám 21. století. Dokument se zaměřuje na několik strategických cílů, včetně zlepšení kvality vzdělávání, snížení nerovností v přístupu ke vzdělání a podpory pedagogických pracovníků.

Strategie se zaměřuje na několik hlavních oblastí:

- Získávání kompetencí – zaměřuje se na rozvoj dovedností potřebných pro aktivní občanský, profesní i osobní život.
- Rovný přístup ke vzdělání – snaží se snížit nerovnosti v přístupu ke kvalitnímu vzdělávání a umožnit maximální rozvoj potenciálu všech dětí, žáků a studentů.
- Podpora pedagogických pracovníků – poskytuje podporu učitelům a dalším pedagogickým pracovníkům, aby mohli efektivněji vykonávat svou práci.
- Inovace ve vzdělávání – zavádí moderní technologie a metody do výuky, včetně digitálního vzdělávání a programování.
- Financování a stabilita – zajišťuje stabilní financování vzdělávacího systému a podporuje jeho dlouhodobou udržitelnost.

Strategie 2030+ klade důraz na digitální vzdělávání a inovace ve vzdělávání. Zahrnuje zavádění moderních technologií a metod do výuky, což zahrnuje i programování. Cílem je, aby žáci získali dovednosti potřebné pro budoucí profesní život, včetně znalostí v oblasti informačních technologií a programování. (MŠMT, 2020)

Školní vzdělávací program (ŠVP)

Školní vzdělávací program (ŠVP) byl zaveden v České republice na základě nového školského zákona (zákon č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání), který vstoupil v platnost 1. ledna 2005. Tento zákon přinesl zásadní změny ve vzdělávacím systému, včetně přechodu na rámcové vzdělávací programy (RVP) a následně i na školní vzdělávací programy.

Při vzniku nového RVP má škola povinnost do dvou let vytvořit nové ŠVP či jej aktualizovat a zahájit dle něj výuku.

Každá škola si na základě RVP vytváří vlastní ŠVP, které konkretizují a přizpůsobují rámcové požadavky specifickým podmínkám a potřebám školy, jejím žákům a lokalitě.

Jsou schvalovány a vydávány ředitelem školy a musí být veřejně přístupné.

Tvorba ŠVP

ŠVP na školách zpravidla tvoří předem pověřený tým pedagogů, který má za úkol provést nejprve analýzu RVP pro daný obor.

Následuje samotná tvorba obsahu. Na základě RVP pedagogové vytvářejí konkrétní obsah vzdělávání, který může být uspořádán do předmětů nebo modulů. Je důležité zajistit správné mezipředmětové přesahy a vazby, aby učivo na sebe co nejlépe i mezipředmětově navazovalo, například předmět Programování a vývoj aplikací má vazbu na předmět Algoritmizace a tedy není potřeba v tomto předmětu vysvětlovat některé, byť důležité fakty, protože s nimi byli žáci již seznámeni v jiném předmětu.

Finální verze ŠVP poté přejde do procesu schválení ředitelem školy a zveřejnění.

Obsah ŠVP

Každý školní vzdělávací program by měl obsahovat povinné a další klíčové prvky, kterými jsou:

- Identifikační údaje: Název školy, název programu, obor vzdělání.
- Charakteristika školy a ŠVP: Popis školy, její vize, mise a specifika vzdělávacího programu.
- Charakteristika ŠVP: Organizace výuky, způsob a kritéria hodnocení žáků, podmínky přijímání ke vzdělávání, způsob ukončení studia, vzdělávání žáků se speciálními vzdělávacími potřebami a další informace.
- Učební plán: Přehled předmětů a jejich časová dotace.
- Učební osnovy: Detailní popis obsahu jednotlivých předmětů.
- Materiální a personální zajištění výuky: Materiální a personální podmínky a také podmínky bezpečnosti práce a ochrany zdraví při vzdělávacích činnostech.
- Hodnocení žáků a autoevaluace školy: Kritéria a metody hodnocení žáků, postupy pro autoevaluaci školy. (Kašparová, 2012)

Národní program rozvoje vzdělávání (Bílá kniha)

Národní program rozvoje vzdělávání v České republice, známý také jako Bílá kniha, byl schválen vládou České republiky dne 7. dubna 1999. Tento dokument formoval vládní strategii v oblasti vzdělávání a obsahoval myšlenková východiska, obecné záměry

a rozvojové programy, které měly být směrodatné pro vývoj vzdělávací soustavy. Bílá kniha byla platná až do schválení Strategie vzdělávací politiky České republiky do roku 2020, tedy částečně i v době přípravy této diplomové práce.

V roce 2020 však definitivně pozbyla platnosti. (MŠMT, 2020)

2 Zařazení do ŠVP na konkrétní škole

Na Střední průmyslové škole Ostrov, p.o., kde vyučuji a tuto metodiku jsem se pokusil nasadit a ověřit v praxi, je vyučován obor vzdělávání **18-20M/01 Informační technologie**, pro který je v této diplomové práci zpracována a ozkoušena metodika výuky skriptovacího jazyka PHP na straně serveru.

2.1 RVP pro obor 18-20-M/01 Informační technologie

Tento program je navržen tak, aby připravil žáky na různé role v oblasti informačních technologií, jako jsou programátoři, správci sítí, technici IT a další.

Hlavní cíle RVP pro obor Informační technologie

- Rozvoj klíčových kompetencí – důraz na rozvoj kritického myšlení, týmové spolupráce a schopnost řešit problémy, které jsou nezbytné pro úspěch v oblasti IT
- Rozvoj odborných znalostí a dovedností – žáci by měli získat teoretické i praktické znalosti v oblasti informačních technologií, včetně programování, správy sítí, databází a dalších technologií.
- Přípravenost na trh práce – program klade důraz na praktické dovednosti a aplikaci teoretických znalostí v reálných situacích.
- Podpora celoživotního učení – RVP podporuje myšlenku, že vzdělání v oblasti IT by mělo být dynamické a mělo by připravit žáky na neustálý profesní rozvoj a přizpůsobování se novým technologiím.
- Flexibilita a adaptabilita – umožňuje školám přizpůsobit své školní vzdělávací programy (ŠVP) specifickým potřebám regionu a trhu práce.

(MŠMT, 2020)

Aktualizace z roku 2023

V roce 2023 došlo k významné aktualizaci RVP v oblasti informatického vzdělávání. Tyto změny zahrnují zejména:

- Modernizaci obsahu – aktualizace reflektují nejnovější poznatky v oblasti informačních a komunikačních technologií (ICT) a zahrnují nové pojetí informatického vzdělávání.

- Nové pojetí informatického vzdělávání – aktualizace zavádí nový koncept informatického vzdělávání, který zahrnuje rozvoj digitálních kompetencí a průřezové téma "Člověk a digitální svět"
- Zvýšení důrazu na digitální kompetence – nové požadavky kladou větší důraz na rozvoj digitálních dovedností, které jsou nezbytné pro úspěšné uplatnění na trhu práce.
- Integrace nových technologií – aktualizace zahrnují i nové technologie, jako jsou umělá inteligence, kybernetická bezpečnost a internet věcí (IoT).
- Interdisciplinární přístup – podpora propojení s jinými obory, jako je například využití IT v oblasti zdravotnictví, průmyslu nebo umění, čímž se zvyšuje relevance výuky.

(MŠMT, 2023)

2.2 ŠVP pro obor 18-20M/01 Informační technologie na SPŠ Ostrov, p.o.

Aktuální verze tohoto dokumentu je na SPŠ Ostrov, p.o. zpracována dle RVP ze dne 29.5.2008, č. j. 6 907/2008-23 ve znění aktualizovaného RVP vydaného MŠMT dne 1.9.2020 pod číslem jednacím 1269/2022/SPS. Jeho platnost je od 1. 9. 2022 počínaje I. ročníkem.

Aktualizovaný dokument na základě poslední aktualizace RVP ze strany MŠMT z roku 2023 není zatím zaveden a výuka probíhá dle platných ŠVP.

Třída, na které byla metodika ověřovaná, nastoupila do prvního ročníku od 1.9.2020, tedy její průběh studia řešilo starší ŠVP zpracované dle RVP ze dne 29. 5. 2008, č. j. 6 907/2008-23 pod číslem jednacím 360/2018/SPS-3, platné od 1. 9. 2018 počínaje I. ročníkem. Z hlediska předmětu Programové vybavení, do kterého je tato metodika zařazena, nedošlo aktualizací ŠVP k 1.9.2022 k žádným změnám. (Střední průmyslová škola Ostrov, příspěvková organizace, 2018)

2.2.1 Prostor pro značkovací a skriptovací jazyky

V prvním ročníku je zařazen do učebního plánu předmět Základy algoritmizace s jednohodinovou týdenní dotací. Předmět je zařazen do prvního ročníku jako teoretická příprava k předmětu Programové vybavení, vyučovaný ve 2. ročníku. V tomto předmětu je zařazena výuka skriptovacího jazyka Python.

Od druhého ročníku probíhá výuka dvou klíčových předmětů, zaměřených na aplikaci programovacích, skriptovacích a značkovacích jazyků. Prvním z nich je předmět Programové vybavení a druhý Programování a vývoj aplikací.

Programové vybavení (PGV)

Předmět je vyučován ve druhém, třetím a čtvrtém ročníku vždy s tříhodinovou týdenní dotací.

Na začátku druhého ročníku se žáci seznámí se základními typy algoritmů a jejich zápisem pomocí vývojových diagramů. Následují základy programovacího jazyka C#, jako je práce s proměnnými, podmíněné příkazy a cykly.

Ve třetím ročníku se žáci naučí používat složitější struktury – pole, soubory, indexované seznamy, využívají v programech animaci a pohyb objektů, používají ovládání objektů pomocí klávesnice i myši.

Ve čtvrtém ročníku žáci využívají možnosti objektově orientovaného programování, tvoří vlastní moduly a pracují na složitějších programech.

Programování a vývoj aplikací (PVA)

Předmět je taktéž vyučován ve druhém, třetím a čtvrtém ročníku vždy s tříhodinovou týdenní dotací.

Cílem předmětu je naučit žáky znalostem a dovednostem z oblasti tvorby moderních webových stránek a webových aplikací.

Na začátku druhého ročníku se žáci učí vytvářet layout webové stránky s použitím HTML, CSS. Ve druhé části tohoto ročníku se učí používat skriptovací jazyk na straně klienta. Pro výuku je použito skriptování nejprve jazykem JavaScript, později doplněn o framework jQuery.

Třetí ročník na začátku obohacuje dovednosti žáků o responzivní webdesign s použitím frameworků. Větší část třetího ročníku je poté věnována skriptovacímu jazyku PHP na straně serveru, který je obohacen o práci s databází.

Ve čtvrtém ročníku je učivo rozšířeno o práci s objekty. Výuka je později směřována spíše k samostatné tvorbě různých typových webových aplikací, jako jsou například návštěvní knihy, diskuse, redakční systémy atd.

Při výuce jazyka PHP je využita vazba na předmět Programové vybavení a Algoritmizace, díky kterému mají žáci již osvojené základy programování a psaní základních programovacích konceptů v jiných jazycích. Není tedy nutné vysvětlovat žákům principy například podmíněného větvení, cyklů apod., ale pouze zdůraznit případné rozdíly od jazyků, se kterými se doposud setkali.

Profilová část maturitní zkoušky

Žáci skládají profilovou maturitní zkoušku z Českého jazyka, z cizího jazyka, pokud si žák nezvolil ve společné části předmět matematika, dále skládá ústní zkoušku z předmětů Soubor předmětů z informatiky a Počítačové sítě a systémy.

Dále žáci vykonávají praktickou zkoušku z předmětu Řízení strojů a procesů, anebo mohou volitelně tuto zkoušku nahradit předmětem Programování aplikací.

Programování aplikací je tzv. dlouhodobá maturitní práce, jejíž výstupem je funkční aplikace, tvořená nejčastěji programovacím jazykem C# či webová aplikace, která musí splňovat několik povinných kritérií, kde jedno z důležitých je objektově orientované programování v jazyce PHP. Samotná práce je poté hodnocena vedoucím a oponentem a v závěru prezentovaná a obhajovaná před maturitní komisí.

Žáci mohou využít volby vlastního téma, které musí splňovat alespoň minimální školou stanovené požadavky a projít procesem schválení, nebo mohou zvolit jedno z nabízených témat školou. V minulosti žáci vytvářeli například kompletní webovou aplikaci pro školní knihovnu s evidencí knih, s možností rezervace, správy výpůjček a dalších důležitých funkcí, dále například burzu knih, učebních pomůcek či doučování pro školní potřeby, nebo také redakční systém pro spolek zahrádkářů ve městě Ostrov, který o tuto aplikaci zažádal.

Z výše zmíněného plyne, že značkovací a skriptovací jazyky pro tvorbu webových aplikací jsou u tohoto oboru klíčovými jazyky, které mohou žáci využít pro úspěšné splnění části maturitní zkoušky.

(Střední průmyslová škola Ostrov, příspěvková organizace, 2022)

3 Výběr konkrétní platformy pro výuku

3.1 Důvody pro výuku PHP ve školách

Oproti tradičním programovacím jazykům jako je C, C#, JAVA a další, jsou skriptovací jazyky globálně ideálním začátkem pro výuku programování. Jazyk Python je uváděn jako ideální skriptovací jazyk pro začátečníky díky své jednoduchosti. (Lutz, 2013)

Skriptovací jazyk PHP má potenciál pro žáky, protože nabízí možnosti, jak převést statické HTML stránky, se kterými mohou mít žáci již zkušenosti, do dynamické podoby na straně serveru.

Hlavní důvody pro zařazení výuky z předchozích informací jsou tedy:

- Snadná integrace do HTML – skripty v PHP je možno psát přímo do HTML souborů, není tedy nutné oddělovat kód do samostatných souborů.
- Snadná instalace – PHP je open-source, snadno se instaluje na většině serverů (např. Apache nebo Nginx) a je zdarma ke stažení.
- Rychlý vývoj – nevyžaduje složité nastavení a kompilaci.
- Snadná práce s databázemi – PHP má vestavěné metody či třídy pro snadnou práci s databázemi, jako jsou MySQL či MariaDB, taktéž dostupné zdarma.
- Široká komunitní podpora – existuje mnoho webových stránek nabízející podporu, výuku či diskusní fóra s řešením různé problematiky.

Výše zmíněné důvody a aktuální popularita PHP s podílem 75,5% na světových serverech je jistě důvodem, proč tento jazyk vyučovat, neboť poskytuje absolventům prostor uplatnění na trhu práce v roli případných tvůrců webových stránek.

3.2 Jazyk PHP

PHP je skriptovací programovací jazyk používaný zejména pro vývoj webových aplikací. Byl vyvinut v roce 1994 Rasmussem Lerdorfem a původně sloužil k tvorbě dynamických webových stránek. Od té doby prošel PHP významným vývojem a stal se jedním z nejpoblárnějších jazyků pro webový vývoj. (The PHP Documentation Group, 2020)

Vlastnosti PHP:

- Server-Side Scripting – PHP běží na straně serveru, což znamená, že kód je zpracován na serveru a výsledný výstup HTML tagů s daty je odeslán klientovi.
- Dynamicky typovaný – datový typ je vázaný na hodnotu v proměnné.
- Objektově orientovaný programovací jazyk – kód je přidružen k datům a metody jsou zapouzdřeny v objektech. (Wikipedie: Otevřená encyklopedie., 2024)

Použití PHP

PHP lze použít k tvorbě konzolových, desktopových či webových aplikací.

Jedná se o jeden z nejpoužívanějších skriptovacích jazyků pro tvorbu webových dynamických stránek skriptovaných na straně serveru. V roce 2024 dosahuje popularity 75,5%. (W3Techs, 2024)

Jazyk PHP se rychle rozšířil mezi vývojáře webových stránek, protože nabízí jednoduchou syntaxi, snadnou integraci do HTML a podporu pro interakci se serverem.

PHP je skriptovací jazyk s otevřeným zdrojovým kódem, což znamená, že jeho zdrojový kód je volně dostupný a může být upravován a rozšiřován komunitou vývojářů.

V roce 2000 byla vydána verze PHP 4, která přinesla výrazné změny a zlepšení výkonu.

V roce 2004 byla vydána verze PHP 5, která přinesla mnoho nových vlastností, včetně podpory objektově orientovaného programování.

Poslední stabilní verzi PHP je PHP 8, která byla vydána v roce 2020, přesněji tedy verze 8.4.1 vydána aktuálně v době finalizace této práce, přesněji dne 21. listopadu 2024 (The PHP Group, 2024). PHP 8 přináší další významné vylepšení výkonu a nové funkce, jako je zlepšená podpora pro typování, nové datové typy, nové operátory a mnoho dalšího. (The PHP Documentation Group, 2020)

PHP je široce používán vývojáři po celém světě pro tvorbu dynamických webových stránek, správu databází, manipulaci se soubory a mnoho dalších úkolů. Existuje také mnoho rozsáhlých frameworků a knihoven postavených na PHP, jako je například Laravel, Symfony, CodeIgniter, CakePHP, či české Nette (Grud, 2024) a další, které usnadňují vývoj webových aplikací. (Radix, 2024)

3.3 Nutné technické vybavení

Pro práci s PHP je potřeba několik základních technických prvků:

- **Webový server** – PHP je skriptovací jazyk, který je interpretován na straně serveru, takže je nezbytné mít nainstalovaný webový server, který podporuje PHP. Nejpoužívanějšími webovými servery pro PHP jsou Apache, Nginx a Microsoft IIS. (netcraft, 2024)
- **PHP interpret** – PHP soubory musí být interpretovány pomocí PHP interpretu. PHP je možno nainstalovat jako samostatný software nebo jako součást webového serveru. PHP interpret zpracovává PHP kód a generuje výstup, který je odeslán do prohlížeče.
- **Databáze** – pro základní práci s PHP není databáze nutná, ovšem mnoho webových aplikací napsaných v PHP využívá databáze pro práci s daty. Populární databázové systémy, které se často používají s PHP, zahrnují MySQL a její větve, jako například MariaDB, dále PostgreSQL či SQLite. (Medium, 2022)
- **Vývojové prostředí** – pro pohodlný vývoj skriptů v PHP je užitečné mít vhodné vývojové prostředí. Existuje mnoho vývojových prostředí IDE (Integrated Development Environment) a textových editorů s funkcemi specifickými pro PHP, jako je syntaxové zvýraznění, automatické doplňování kódu a ladění aplikací. Pro komerční účely mohou být některá prostředí zpoplatněna. Mezi zdarma dostupné IDE lze zařadit Visual Studio Code (VS Code), Sublime Text, Atom či NetBeans. Zpoplatněné, ale velice oblíbené prostředí může být například PhpStorm od české společnosti JetBrains s.r.o., které je dle statistik velice používané napříč celým světem. (JetBrains: Developer Tools for Professionals and Teams, 2023)

3.4 Technické vybavení pro výuku PHP na školách

Pro výuku na školách, oproti výše zmíněnému technickému vybavení pro běh a tvorbu skriptů v PHP, je nutné koncipovat výuku v počítačových učebnách. Zde je potřeba zvážit, zda výuka bude probíhat formou nahrání skriptů na školní server s podporou pro PHP, pokud jí škola disponuje, registrací žáků u poskytovatelů webového hostingu, tedy poskytovatelů prostoru pro soubory k internetové prezentaci, nebo zda bude výuka probíhat formou výuky

na tzv. lokálních serverech, tedy každý počítač bude mít v sobě nainstalovaný webový server s podporou PHP. Tím se samozřejmě liší požadavky na technické vybavení v učebnách.

3.4.1 Řešení pomocí školního serveru

V případě volby zpracování skriptů pomocí školního serveru je nutno zajistit pro každého žáka vlastní složku na serveru s patřičným oprávněním pro nahrávání, editaci a spuštění souborů, dále vytvořit databázový účet a vytvořit například pouze jednu databázi pro žáka s možností vlastní správy tabulek v této databázi.

Je nutné si uvědomit, že každý skript je nutné v průběhu vývoje neustále nahrávat na školní server pomocí připojeného síťového disku k serverové složce ve Windows či přes šifrovaný protokol sFTP, SCP apod. V prvotní fázi seznámení se s jazykem PHP je toto opravdu velké zdržování a vzniká mnoho problémů s tím, že žáci nenahrají soubory správným způsobem do správných složek a často bývá celá vyučovací hodina věnována řešení takto vznikajících problémů než samotné výuce skriptovacího jazyka.

3.4.2 Řešení pomocí poskytovatele hostingových služeb

Volba externího poskytovatele hostingových služeb je často placená a skrývá mnoho omezení, jako je limit datového přenosu, nutnost mít po celou dobu funkční připojení k internetu, tedy nelze žákům odepřít přístup k internetu během výuky, nutnost vlastnit doménu nebo zřídit subdoménu pro tento účel, přístup k těmto aplikacím ze strany celého světa, pokud není tato možnost zakázána a další.

Tuto možnost jsem osobně testoval ve školním roce 2011-2012 na již zaniklé Střední průmyslové škole Locket, p.o. s myšlenkou, že žáci budou vytvářet na svých zdarma vytvořených subdoménách a diskových prostorech u poskytovatelů funkční webové prezentace. Praxe však byla špatná, sám jsem musel mít udělený od každého žáka přístup k jeho prostoru přes protokol FTP, tedy přibližně 30 samostatných FTP účtů, abych mohl nahlížet do souborů na serveru. Zároveň vznikl problém se vkládáním reklamy, neboť společnosti měly tuto možnost vyhrazenou smluvně atd. Výuka již dále touto formou nepokračovala.

3.4.3 Řešení pomocí lokálních webových serverů

Apache

Apache HTTP Server, označovaný jako Apache, je zdarma dostupný open-source software, používaný k tvorbě webových serverů.

Apache poskytuje prostředí pro hostování webových stránek a aplikací. Podporuje mnoho operačních systémů, včetně Unixu, Linuxu, Windows a dalších. Jeho hlavními vlastnostmi jsou spolehlivost, výkon a široká škála konfiguračních možností.

Apache poskytuje také funkce pro správu a konfiguraci webového serveru, jako jsou přístupové kontroly, logování, SSL/TLS šifrování, zálohování a další. Může být také rozšířen pomocí modulů a rozšíření pro podporu specifických funkcí a technologií.

Lze jej nainstalovat i v podobě předkompilovaných balíků XAMPP, AMPPS a další.

(The Apache Software Foundation, 2024)

PHP

Pro práci s PHP na serveru Apache je potřeba mít nainstalovaný modul PHP pro Apache, který umožňuje interpretaci PHP kódu. Existují různé způsoby, jak nakonfigurovat Apache pro podporu PHP, včetně integrace modulu do konfiguračního souboru Apache. Jedna ze snadných možností instalace a implementace bude popsána níže.

Databáze

MySQL a MariaDB jsou relační databázové systémy, které mají společný původ.

MariaDB vznikla jako fork (větev) MySQL v roce 2009, kdy se objevily obavy ohledně budoucího směřování MySQL pod novým vlastníkem Oracle Corporation. MariaDB byla vyvinuta s cílem zachovat kompatibilitu s MySQL a zároveň přinést několik vylepšení a nových funkcí. (MariaDB Foundation, 2024)

Oba systémy se podobají v mnoha ohledech a používají podobnou syntaxi SQL pro manipulaci s daty.

Některé z hlavních rysů a funkcí MySQL a MariaDB jsou:

- Relační databáze – oba systémy jsou relační databázové systémy, které podporují tabulkovou strukturu pro ukládání dat.
- Výkon – MySQL i MariaDB jsou známé svým vysokým výkonem a rychlostí zpracování.
- Bezpečnost – oba systémy nabízejí mechanismy pro zabezpečení dat, včetně přístupových práv, šifrování a autentizace.
- Podpora pro standardy – MySQL i MariaDB jsou dobře podporované systémy s velkou uživatelskou základnou a rozsáhlou dokumentací. Podporují mnoho standardů a technologií, jako je například ACID (Atomicity, Consistency, Isolation, Durability).
- Replikace a clusterování – oba databázové systémy podporují replikaci a umožňují vytváření databázových clusterů pro vyšší dostupnost a odolnost.

Hlavním rozdílem mezi MySQL a MariaDB je vývojová cesta a vlastnictví. MySQL je v současnosti vlastněná a spravovaná společností Oracle Corporation, zatímco MariaDB je open-source projekt, který je nezávislý a spravovaný komunitou vývojářů.

Další rozdíly zahrnují některé specifické funkce a vylepšení, které byly přidány do MariaDB po odštěpení od MySQL. MariaDB nabízí například některé nové úložné enginy, jako je XtraDB (založený na InnoDB), nebo ColumnStore pro analytické dotazy.

Pro výuku skriptovacího jazyka PHP s databází na střední škole není nutné věnovat těmto rozdílům velkou pozornost, protože ve většině případů se při výuce jedná o provádění základních DDL a DML příkazů, které jsou pro obě varianty totožné. (websupport, 2023)

Balíček XAMPP

XAMPP je softwarový balíček, který kombinuje několik často používaných technologií pro webový vývoj a umožňuje jednoduchou instalaci a konfiguraci webového serveru na operačním systému Windows, Linux či OS X.

Pro jeho jednoduchost jsme jej zvolili na škole jako výchozí lokální webový server na všech počítačích, kde probíhá výuka skriptovacího jazyka PHP. Díky integraci databáze se tento balíček osvědčil i kolegům při výuce předmětu Programové vybavení, kde s použitím patřičných knihoven vytvářejí desktopové aplikace v jazyce C#, propojené se stejnou

databází. Je tedy možno k desktopovým aplikacím vytvořit například i webovou prezentaci či vzdálenou správu a výstupy dat přes internet.

Výhodou tohoto komplexního balíčku je, že instalace je prováděna opravdu automaticky na pár kliknutí a uživatel může při instalaci volit jen určité komponenty, které skutečně využije.

XAMPP zahrnuje následující komponenty:

- Apache – webový server Apache poskytuje prostředí pro hostování webových stránek a aplikací.
- MySQL – databázový systém MySQL či MariaDB pro ukládání a získávání dat.
- PHP – PHP interpret, který umožňuje spouštění PHP skriptů na serveru.
- phpMyAdmin – nástroj pro uživatelsky přívětivou správu databáze MySQL přes webové rozhraní.
- OpenSSL – poskytuje podporu pro šifrování SSL/TLS, což je důležité pro zabezpečené přenosy dat přes internet, ovšem při výuce skriptování na škole není nutné.
- cUrl – slouží k provádění požadavků HTTP z PHP skriptů. Umožňuje stahovat data z jiných serverů, odesílat data na servery, pracovat s API a mnoho dalšího.
- Tomcat – slouží k nasazení a správě aplikací napsaných v jazyce Java. Tomcat je servletový kontejner, což znamená, že umožňuje spouštění Java servlets a JavaServer Pages (JSP).

XAMPP balíček je snadno dostupný ke stažení na oficiálních stránkách projektu¹. Po stažení a instalaci XAMPPu je možné jednoduše spustit potřebné komponenty pro vývoj a testování webových aplikací v prostředí Windows přes ovládací prvek XAMPP Control Panel, nebo jednotlivé komponenty zavést jako službu operačního systému a spouštět je automaticky po spuštění systému. Pokud je v počítačové učebně vyučován i jiný předmět, je na zvážení, zda je nutné, aby i v tomto případě běžel celý webový server na každém stroji automaticky formou služby a zbytečně si vyhrazoval určitou část výkonu počítače i když s ním není pracováno. (XAMPP, 2024)

¹ <https://www.apachefriends.org/>

3.4.4 Online prostředí pro interpretaci skriptů – W3Schools.com

W3Schools umožňuje uživatelům testovat a spouštět PHP skripty přímo na jejich stránkách pomocí nástroje zvaného Tryit Editor². Tento editor poskytuje integrované prostředí, kde je možno napsat a interpretovat PHP skript bez nutnosti nastavovat místní server nebo instalovat další software.

Výhody tohoto editoru jsou:

- Okamžité vykonání kódu – uživatelé mohou okamžitě vidět výstup svého PHP skriptu přímo v prohlížeči.
- Rozdělená obrazovka – umožňuje demonstrovat skript a zároveň interpretované výstupy
- Bez instalace – nepotřeba instalace softwaru a další nastavení, vše probíhá v okně internetového prohlížeče

Nevýhody:

- V základu bez práce se soubory a adresáři
- Některé běžně používané funkce nemusí být podporovány

(Refsnes Data, 2024)

3.4.5 Výuka na Střední průmyslové škole Ostrov, p.o.

Žáci na Střední průmyslové škole Ostrov, p.o. mají možnost si zřídit na školním serveru vlastní prostor pro své prezentace, do svých složek nahrávat i PHP skripty, požádat o zřízení databáze a jejich práce mohou být i veřejně dostupné z internetu pod školní doménou. Učitelé mají právo do těchto složek nahlížet. Přístup je ve školní síti realizován pomocí připojeného síťového disku a z domova přes protokol SCP. (Střední průmyslová škola Ostrov, p.o., 2024)

Pro začátky výuky jazyka PHP, bez nutnosti prezentace svých prací veřejnosti, se velice osvědčila tvorba skriptů pomocí lokálních serverů na jednotlivých počítačích.

² https://www.w3schools.com/php/php_install.asp

Žáci takto mohou na každém počítači v předem připravené učebně spustit lokální server a poté v lokální složce pracují napřímo se skripty, které mohou okamžitě po uložení vykonávat svým vlastním serverem, bez nutnosti nahrávání do vzdálených složek.

V případě lokálních serverů pomocí balíčku XAMPP využíváme pro výuku pouze komponenty Apache, PHP a MySQL spolu s phpMyAdmin.

Problém ovšem nastává v případě migrace dat, kdy tato složka není součástí cestovního profilu žáka. Je tedy nutné, aby každý své práce na konci hodiny zálohoval jednorázově na svůj diskový prostor a v ideálním případě seděl pokaždé u stejného počítače i následující vyučovací hodiny.

Druhý problém nastává, že ke skriptům mají ve výchozím nastavení přístup všichni uživatelé na tomto počítači, je tedy nutno brát na tuto skutečnost zřetel a nevyplňovat v tomto případě do skriptů citlivé údaje. Ovšem jak bylo zmíněno, pro začátky výuky se tato varianta velice osvědčila pro svou jednoduchost a také rychlou možnost instalace knihoven, povolení dalších funkcí atd. V případě školního serveru je nutné vždy žádat správce a často nejsou požadavky zcela vyřízené, neboť některé změny by mohly mít za následek chybné zabezpečení serveru nebo chybný chod pro zbylé školní aplikace na něm.

Pro samotnou výuku je tedy potřeba:

- Počítače s operačním systémem, nainstalovaným internetovým prohlížečem, webovým serverem s podporou PHP a případně také databázový systém.
- Vývojové prostředí (IDE) - počítače je vhodné doplnit o vhodné vývojové prostředí nebo textový editor pro psaní PHP kódu. Existuje několik IDE a editorů, které jsou zdarma, jako například Visual Studio Code, Atom, NetBeans či pro účely výuky je zdarma i samotné prostředí PhpStorm (JetBrains s.r.o., 2024), ovšem tyto IDE jsou skvělým pomocníkem pro praktickou práci s PHP díky syntaxovému zvýraznění, automatickému doplnění kódu a dalším možnostem. Při výuce se ovšem možnost automatického doplnění kódu neosvědčila. Žáci si potřebují osvojit jednotlivé příkazy v jejich plném znění, nikoliv napsat počáteční písmena a vybrat tag z nabídky. Při výuce jsme zvolili bezplatný český textový editor PSPad, obsahující zvýraznění syntaxe, obsahující doplňky jako je Lorem ipsum generátor pro vyplnění

obsahu fiktivním textem a další užitečné nástroje, ovšem bez našeptávání a automatickému doplnění kódu. (Jan Fiala PSPad, 2023)

4 Metodika výuky skriptovacího jazyka PHP

4.1 Požadavky na začátku výuky PHP

Při výuce skriptovacího jazyka PHP se uvažuje, že žáci mají základní znalosti značkovacího jazyka HTML, CSS, rozšířeného o JavaScript či framework pro front-end stránek. Kromě webových stránek se předpokládá, že žáci již mají základní znalost programování, porozumí základním programovacím konceptům, jako jsou proměnné, podmínky, smyčky a funkce. Pokud žáci i na jiných školách mají zkušenosti s jinými programovacími jazyky, může jim to pomoci při pochopení základních principů PHP.

Nad rámec těchto předpokladů je pro výuku PHP vhodné mít a dále se pokoušet rozvíjet:

- Logické myšlení – programování vyžaduje schopnost logického myšlení a řešení problémů. Žáci by měli být schopni analyzovat úlohy, identifikovat kroky řešení a implementovat je pomocí PHP kódu.
- Samostatnost a sebeorganizace – výuka PHP vyžaduje určitou míru samostatnosti a schopnost sebeorganizace. Žáci by měli být schopni studovat a porozumět dokumentaci PHP, experimentovat s kódem, hledat a řešit problémy a rozvíjet své dovednosti samostatně mimo vyučování.
- Praktická cvičení a projekty – výuka PHP často zahrnuje praktická cvičení a projekty, ve kterých žáci vytvářejí reálné webové aplikace. Žáci by měli být připraveni aktivně se zapojit do praktických úkolů, spolupracovat v týmu a aplikovat své znalosti v praxi.

Má poznámka, kterou rád žákům sděluji:

Jakýkoliv programovací jazyk je pro většinu z nás jako cizí jazyk, kterým se snažíme naučit plynulé konverzace. Je zřejmé, že například anglickým jazykem se také nenaučíme za tři vyučovací hodiny týdně ve škole, ale je potřeba tomuto jazyku věnovat i určitou část samostudia. Stejně tak je tomu u programovacích jazyků.

4.2 Očekávané výstupy

Očekávané výstupy žáků na konci úspěšného studia střední školy:

- Základní znalosti syntaxe PHP – žáci by měli být schopni porozumět a používat základní syntaxi jazyka PHP, včetně proměnných, datových typů, podmínek, smyček a funkcí.
- Vytváření a manipulace s databázemi – žáci by měli být schopni vytvářet a manipulovat s databázemi pomocí PHP. To zahrnuje propojení s databázovými systémy, vytváření dotazů SQL, získávání a zpracovávání dat z databází.
- Webové formuláře a validace – žáci by měli být schopni vytvářet webové formuláře pomocí PHP a provádět základní validaci vstupních dat z formulářů, jako je kontrola správného formátu e-mailu, data a času, nebo kontroly povinných polí a jejich délky a obsahu.
- Správa souborů a obrázků – žáci by měli být schopni pracovat se soubory a obrázky pomocí PHP, včetně jejich nahrávání, ukládání, mazání a manipulace s jejich vlastnostmi.
- Systémová integrace – žáci by měli být schopni integrovat PHP do webových stránek a aplikací, například přidáním PHP kódu do HTML šablon, manipulací s URL adresami.
- Základy bezpečnosti – žáci by měli být seznámeni se základy bezpečnosti webového vývoje v PHP, jako je správné ošetřování vstupních dat (ochrana proti SQL injection, XSS útokům), práce s uživatelskými oprávněními a řízením přístupu.
- Debugování a testování – žáci by měli být schopni ladit a testovat svůj PHP kód pomocí nástrojů pro debugování, hledání chyb a optimalizaci výkonu.
- Projektová práce – žáci by měli být schopni pracovat na projektové práci, ve které budou využívat své dovednosti a znalosti PHP k vytvoření kompletní webové aplikace. Důležitým prvkem je i dodržování jednotné “kultury” programování, dostatečného komentování kódu a dalších prvků, umožňující kooperativní činnosti.

4.3 Výběr vhodných tematických celků

Metodika byla navržena na základě výuky předmětu Programování a vývoj aplikací na Střední průmyslové škole Ostrov, p.o. ve třetím ročníku u tematického celku III.

Tématický plán pro třetí ročník:

Pořadové číslo	Tématický celek a téma	Počet vyučovacích hodin
I.	Opakování a rozšíření učiva 2. ročníku HTML, CSS, layout webu JavaScript, knihovna jQuery, pluginy	Celkem 15 9 6
II.	Responzivní design webu Zásady responzivního webu, mobilní zařízení Základní třídy css-knihovny (např. Bootstrap, Foundation apod.) a jejich typické použití Tvorba vlastního responzivního webu	Celkem 18 3 9 6
III.	Programování v PHP Prostředí PHP, základní příkazy, proměnné, datové typy Příkazy větvení a cykly Funkce, globální proměnné a pole Textové soubory, řetězce Formuláře Hlavičky, session	Celkem 36 6 3 9 6 9 3
IV.	MySQL databáze Základní příkazy SQL, práce v MySQL Admineru Výběr a filtrování dat z databázové tabulky Použití dat z databáze ve webové aplikaci Tvorba vlastní databázové aplikace	Celkem 33 6 6 6 15

Tabulka 1 - Tématický plán PVA 3. ročník

5 Metodické listy

Metodické listy vychází z tematického celku ŠVP zaměřeného na jazyk PHP ve třetím ročníku, a to celek číslo:

III. Programování v PHP s celkovou hodinovou dotací 36 hodin, obsahující témata:

- Prostředí PHP, základní příkazy, proměnné, datové typy – 6 hodin
- Příkazy větvení a cykly – 3 hodiny
- Funkce, globální proměnné a pole – 9 hodin
- Textové soubory, řetězce – 6 hodin
- Formuláře – 6 hodin
- Hlavičky, session – 3 hodiny

5.1 Materiální a technické zabezpečení

Žáci: osobní počítače s operačním systémem, přístupu k interpretu PHP ve formě vzdáleného či lokálního serveru, textový editor, internetový prohlížeč.

Učitel: Stejně vybavení jako žáci, projektor, tabule a psací potřeby, scénář aktivit.

5.2 Tématický celek III. Programování v PHP

Zaměření metodických listů:

Obor: 18-20-M/01 Informační technologie

Oblast RVP: PROGRAMOVÁNÍ A VÝVOJ APLIKACÍ

Učivo RVP: 2 Strukturované programování a skriptování

Oblast ŠVP: Programování a vývoj aplikací – Programování v PHP

Věk žáků: 3. ročník SŠ

Vhodné zařazení: metodické listy je vhodné zařadit po probrání předchozích tematických celků, zaměřených na tvorbu webových stránek v jazyce HTML spolu s CSS, doplněné o případné skripty na straně uživatele v jazyce JavaScript. Výhodou je znalost případných frameworků pro front-end webových stránek pro snadnější formátování a zpracování výstupů ze skriptu.

Vstupní požadavky: Žák ovládá tvorbu statických webových stránek v jazyce HTML se stylováním pomocí CSS, zná základní tvorbu skriptů na straně klienta v jazyce JavaScript.

Podpůrné zdroje:

Online stránka v anglickém jazyce W3Schools.com, která obsahuje nejen vysvětlení základních principů, konstrukcí algoritmů a funkcí, ale také jedna z možností, jak vykonávat první PHP skripty bez nutnosti instalace serverů nebo alternativní možnost demonstrace částí skriptů a jejich výsledku interpretace při výuce.

<https://www.w3schools.com/php/default.asp>

Oficiální dokumentace k jazyku PHP je k dispozici na stránce:

<https://www.php.net/>

5.2.1 Prostředí PHP, základní příkazy, proměnné, datové typy

Zaměření

Učivo ŠVP: Prostředí PHP, základní příkazy, proměnné, datové typy

Časová dotace: 220–295 minut

Cíle

Výuka dle tohoto metodického listu má za cíl seznámit žáky se základy skriptovacího jazyka PHP, jeho principem činnosti, práci se serverem pro interpretaci skriptů v PHP, tvorbou skriptů, základními příkazy, knihovními funkcemi, zejména pro výstup dat ze skriptu, s datovými typy, proměnnými, konstantami a operátory.

Kompetence

Kognitivní: žák umí analyzovat problémy při práci s datovými typy, proměnnými či konstantami a rozumí základním operacím v jazyce PHP.

Praktické: žák ovládá používání základních funkcí a konstruktů pro výstup dat ze skriptu, používá základní operátory v jazyce PHP, pracuje s datovými typy, proměnnými a konstantami.

Obsah výuky:

- Princip činnosti PHP a serveru
- Princip psaní PHP skriptu do HTML souborů
- Základní příkazy PHP
- Datové typy
- Proměnné a konstanty v PHP
- Superglobální proměnné
- Operátory v jazyce PHP

Organizační formy a metody výuky

Celý materiál je rozdělen do šesti oblastí, které na sebe navzájem navazují.

1. Seznámení se jazykem PHP, jeho principem činnosti a s funkcí serveru

Tato oblast je zaměřena na základní seznámení žáků se skriptovacím jazykem PHP, jeho principem činnosti a rolí serveru při interpretování PHP skriptů. Žáci získají základní povědomí o tom, jak PHP funguje a jak je interpretováno serverem při jeho volání z internetového prohlížeče.

Rozsah: 35–45 minut

1. **Prezentace** se stručným představením jazyka PHP, jeho využitím v praxi, vysvětlení, jak PHP funguje a jeho základní principy³. (10 minut)
2. **Diskuze** o možnostech provozování PHP – lokální server, webový server⁴. (10 minut)
3. **Praktická ukázka** práce s adresářovou strukturou, umístění PHP skriptů, princip volání skriptů přes URL adresu a jejich interpretace serverem⁵. (5 minut)
4. **Otevřená diskuze** s žáky o jejich zkušenostech a otázkách ohledně PHP. (10 minut)

³ <https://www.jakpsatweb.cz/php/jak-zacit.html>

⁴ Například použití informací na stránce <https://cs.wikipedia.org/wiki/PHP>

⁵ Ukázka formou vykonání úvodního skriptu index.php v rootu adresáře htdocs, který je součástí instalace

2. Implementace PHP kódu do HTML stránky a jeho vykonání

Tato oblast je zaměřena na seznámení žáků s implementací PHP kódu do HTML stránek a jeho vykonáním. Žáci se naučí, jak správně začínat a ukončovat PHP skripty, umisťovat PHP kód do HTML stránek a vytvoří svůj první funkční PHP program (skript)⁶.

Rozsah: 30–45 minut

1. **Prezentace** na téma, jak začínat a ukončovat PHP skripty uvnitř HTML souborů⁷. (10 minut)
2. **Demonstrace** různých možností umístění PHP kódu uvnitř HTML stránek⁸. (10 minut)
3. **Aktivita** - žáci vytvoří svůj první PHP program (Úloha 1.1), který zobrazuje informace o PHP konfiguraci serveru pomocí funkce `phpinfo()`. Vysvětlení fungování tohoto skriptu. (10 minut)

3. Výstup dat ze skriptu

Tento okruh je zaměřen na seznámení žáků s různými způsoby výstupu dat ze skriptu v PHP. Žáci se naučí používat základní funkce či konstrukty pro výstup dat, porovnávat jejich použití a efektivně zobrazovat obsah proměnných a dat ve svých PHP skriptech.

Rozsah: 30–45 minut

1. **Praktická ukázka** funkce⁹ či jazykového konstruktu¹⁰ `echo`, dále jen funkce `echo`. Ukázka použití funkce `echo` pro výstup dat do HTML stránky¹¹. (10 minut)
2. **Praktická ukázka** funkce či jazykového konstruktu `print`, dále jen funkce `print` a její srovnání s `echo`. (10 minut)
3. **Prezentace** použití funkce `var_dump` pro podrobné zobrazení obsahu proměnných¹². (5 minut)

⁶ Soubor `info.php` umístěný ve složce `list1` v příloze

⁷ Například použití informací na stránce https://www.w3schools.com/php/php_syntax.asp

⁸ Například použití informací na stránce https://www.w3schools.com/php/php_syntax.asp

⁹ https://www.w3schools.com/php/func_string_echo.asp

¹⁰ <https://www.php.net/manual/en/function.echo.php>

¹¹ https://www.w3schools.com/php/php_echo_print.asp

¹² https://www.w3schools.com/php/phptryit.asp?filename=tryphp_var_dump

4. **Demonstrace** skládání výstupů a použití zkráceného zápisu pro výstup dat¹³. (5 minut)

4. Datové typy, proměnné a konstanty, superglobální proměnné

Tento okruh je zaměřen na seznámení žáků s nejpoužívanějšími datovými typy v PHP, způsobem práce s proměnnými a konstantami a významem superglobálních proměnných.

Rozsah: 40–50 minut

1. **Prezentace** a vysvětlení různých datových typů v PHP (int, float, string, bool)¹⁴. (15 minut)
2. **Demonstrace**, jak přiřadit hodnoty do proměnných a jak přetypovat datové typy¹⁵. (5 minut)
3. **Praktická ukázka**, jak definovat a používat konstanty v PHP¹⁶. (5 minut)
4. **Diskuze** o superglobálních proměnných (5 minut)
5. **Aktivita** – tvorba výpisu obsahu superglobální proměnné \$_SERVER¹⁷ pomocí var_dump – Úloha 1.2¹⁸. (10 minut)

5. Operátory v jazyce PHP, praktické příklady

Tento okruh je zaměřen na seznámení žáků s různými typy operátorů v jazyce PHP a jejich praktickým použitím. Žáci se naučí používat přiřazovací, aritmetické, porovnávací a logické operátory, stejně jako inkrementaci, dekrementaci, operátor sloučení a ternární operátor.

Rozsah: 40–50 minut

1. **Demonstrace** použití přiřazovacího operátoru a **demonstrace** použití aritmetických operátorů a operátoru sloučení¹⁹. (20 minut)
2. **Ukázka** použití porovnávacích operátorů a demonstrace použití logických operátorů. (10 minut)
3. **Ukázky** inkrementace a dekrementace. (5 minut)

¹³ <https://www.php.net/manual/en/function.echo.php>

¹⁴ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_var_dump_more

¹⁵ https://www.w3schools.com/php/php_datatypes.asp

¹⁶ https://www.w3schools.com/php/php_constants.asp

¹⁷ https://www.w3schools.com/php/php_superglobals_server.asp

¹⁸ Soubor info.php umístěný ve složce list1 v příloze

¹⁹ https://www.w3schools.com/php/php_operators.asp

4. Ukázky použití ternárního operátoru. (5 minut)

6. Praktické využití práce s proměnnými, operátory a jejich výstupem

Tato oblast je zaměřena na seznámení žáků s tvorbou vlastního funkčního programu pro matematický výpočet pomocí PHP. Žáci se naučí definovat konstanty, provádět výpočty a formátovat výstup dat.

Rozsah: 45–60 minut

1. **Prezentace** na téma funkce printf a k čemu slouží²⁰. (10 minut)
2. **Demonstrace**, jak formátovat celá čísla, desetinná čísla a zaokrouhlování pomocí printf. (5 minut)
3. **Demonstrace**, jak formátovat a vypsát řetězce pomocí printf²¹. (5 minut)
4. **Diskuze** o praktickém použití proměnných a konstant. (10 minut)
5. **Aktivita** – praktický příklad výpočtu obvodu kružnice na základě použití proměnných, konstant, operátorů a formátovaného výstupu – Úloha 1.3²². (15 minut)

Úlohy:

Úloha 1.1 – první program v PHP a zkouška interpretace skriptu serverem

Tato úloha pomůže žákům seznámit se s funkcí phpinfo(), která poskytuje detailní informace o konfiguraci PHP a serveru. Žáci budou schopni vytvořit jednoduchý PHP skript, který tuto funkci využívá, a budou schopni interpretovat zobrazené informace. Tato znalost je základním krokem k pochopení prostředí, ve kterém PHP skripty běží, jejich nahráváním do složek na serveru a volání jejich interpretace z URL prohlížeče.

Zadání úlohy 1.1:

Vytvořte skript v jazyce PHP, který provede informativní výpis konfigurace serveru pomocí knihovní funkce phpinfo.

Pracovní postup:

²⁰ https://www.w3schools.com/php/func_string_printf.asp

²¹ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_string_printf2

²² Soubor obvod.php umístěný ve složce list1 v příloze

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP. Do souboru napište následující kód:

```
<?php  
phpinfo();  
?>
```

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem info.php.

Otevřete webový prohlížeč a zadejte do adresního řádku URL, která odpovídá vašemu lokálnímu serveru a umístění souboru info.php, v našem případě <http://localhost/info.php> a zavolejte vykonání tohoto skriptu stiskem klávesy Enter.

Metoda ověření:

Zobrazení informací o PHP: Po otevření URL v prohlížeči by se měly zobrazit detailní informace o konfiguraci PHP a serveru. Výstup by měl zahrnovat verzi PHP, nastavení konfigurace, načtené rozšíření, a další informace.

Kontrola správnosti skriptu: Ověřte, že skript neobsahuje žádné chyby a že funkce phpinfo() správně funguje.

Interpretace zobrazených informací: Zkuste porozumět zobrazeným informacím, jako je verze PHP, nastavení direktiv, načtené moduly a další detaily. Diskutujte s učitelem nebo spolužáky o významu těchto informací.

Úloha 1.2 – Tvorba PHP skriptu obsahujícího výpis superglobální proměnné \$_SERVER pomocí var_dump

Tato úloha pomůže žákům seznámit se s použitím superglobálních proměnných v PHP a s funkcí var_dump pro zobrazení jejich obsahu. Žáci se naučí, jak efektivně vypisovat podrobné informace o serveru a prostředí, ve kterém jejich PHP skripty běží.

Zadání úlohy 1.2:

Vytvořte PHP skript, který vypíše obsah superglobální proměnné \$_SERVER pomocí funkcí var_dump.

Postup provedení úlohy:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP. Do souboru vytvořte základní strukturu HTML souboru a do těla stránky napište následující kód:

```
<?php
    var_dump($_SERVER);
?>
```

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem server.php. Tento soubor opět otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení informací: Po otevření URL v prohlížeči by se měly zobrazit detailní informace o serveru a prostředí.

Kontrola správnosti skriptu: Ověřte, že skript neobsahuje žádné chyby a že funkce var_dump funguje správně a vypisuje informace o serveru a prostředí.

Úloha 1.3 – Vytvoření funkčního programu pro výpočet obvodu kružnice

Tato úloha pomůže žákům procvičit si definování konstant, provádění základních matematických výpočtů a formátování výstupu v PHP. Žáci se naučí, jak efektivně definovat a používat konstanty a proměnné, provádět matematické operace a zobrazovat výsledky na obrazovce.

Zadání úlohy 1.3:

Vytvořte PHP skript, obsahující definování vlastní konstanty PI o hodnotě 3,1415 (konstanta PI samozřejmě v jazyce PHP existuje²³ a její přesnost je mnohem lepší, ale pro demonstraci tvorby konstant si vytvoříme vlastní) a proměnné s hodnotou poloměru kružnice. Do proměnné obvod provedte výpočet obvodu kružnice. Výsledný obvod vypište pomocí funkce printf se zaokrouhlením na dvě desetinná místa.

²³ <https://www.php.net/manual/en/math.constants.php#constant.m-pi>

Postup provedení úlohy:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP. Do souboru vytvořte základní strukturu HTML souboru a do těla stránky napište následující kód:

```
<?php
    define("PI", 3.1415);
    $polomer = 5; // centimetry
    $obvod = 2 * PI * $polomer;
    printf("Obvod kruhu je: %.2f cm", $obvod);
?>
```

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem obvod.php. Tento soubor opět otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit výsledek obvodu kruhu.

Kontrola správnosti: Ověřte, že výstup obvodu kruhu je správně zaokrouhlený a zobrazený pomocí funkce printf.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci. Doporučený čas je 6 vyučovacích hodin.

Očekávané problémy:

Nesprávná syntaxe PHP kódu, například zapomenuté ukončení příkazů středníkem či zapomenuté uzavírací značky ?>

Špatné umístění PHP kódu v HTML.

Chybné umístění souborů do složky na serveru.

Chybné volání adresy serveru a názvu skriptu.

Chybná koncovka souboru .html místo .php

5.2.2 Příkazy větvení a cykly

Zaměření

Učivo ŠVP: Příkazy větvení a cykly

Časová dotace: 115–180 minut

Cíle

Tento metodický list je zaměřen na seznámení žáků s podmínkovými konstrukcemi, přepínači a cykly v programovacím jazyce PHP. Žáci se naučí, jak efektivně řídit tok programu pomocí podmínek if, větvi else a elseif, přepínače switch a cyklů while, do-while a for, včetně řídicích příkazů break a continue. Celé téma zahrnuje teoretické i praktické části, které žákům umožní získat základní dovednosti v používání těchto struktur a jejich aplikací v praxi.

Kompetence

Kognitivní: žák umí vytvářet podmínky na základě logických porovnávání a vykonávat tak různé sekvence kódu v závislosti na podmínce, umí větvit program pomocí rozcestníku switch, zná základní cykly v jazyce PHP a umí s nimi opakovat části kódu.

Praktické: žák využívá podmíněné větvení programu, rozcestník switch a základní druhy cyklů při tvorbě skriptů.

Obsah výuky:

- Podmínky
- Větve else
- Větve elseif
- Rozcestník switch
- Cyklus while
- Cyklus do-while
- Cyklus for
- Řídicí příkazy break a continue

Organizační formy a metody výuky

Celý materiál je rozdělen do tří oblastí, které na sebe navzájem navazují.

1. Seznámení s podmínkou větví else a elseif

Tato oblast je zaměřena na seznámení žáků s podmíněnými výrazy v PHP. Žáci se naučí používat podmínky if, volitelnou větev else, vnořené podmínky a větev elseif k vyhodnocování různých situací ve skriptech.

Rozsah: 35–60 minut

1. **Prezentace** na princip činnosti podmínky, vyhodnocení logických operací, negace, konjunkce a disjunkce^{24 25}. (10 minut)
2. **Demonstrace** použití větve else a příklady jejího použití. (5 minut)
3. **Praktická ukázka** vnořených podmínek a jejich efektivní použití²⁶. (5 minut)
4. **Praktická ukázka** použití větve elseif pro zpracování více podmínek. (5 minut)
5. **Aktivita** – žáci vytvoří skript v jazyce PHP, který převede číselnou hodnotu známky na slovní hodnocení – Úloha 2.1²⁷ (10 minut)

2. Přepínač switch

Tato oblast je zaměřena na seznámení žáků s přepínačem switch v PHP, který slouží k vyhodnocování různých případů na základě vstupní hodnoty. Žáci se naučí, jak používat switch, kdy je vhodné jej použít, jak zadávat vstupní parametry, vytvářet a ukončovat větve, slučovat větve a definovat výchozí větev.

Rozsah: 40–60 minut

1. **Prezentace** na téma přepínač switch, jeho funkce a kdy je vhodné jej použít²⁸. (10 minut)
2. **Demonstrace**, jak zadávat vstupní parametry do switch. (5 minut)
3. **Praktické ukázky**, jak vytvářet větve uvnitř switch a jak je správně ukončovat pomocí break²⁹. (5 minut)
4. **Demonstrace**, jak sloučit více větví a definovat výchozí větev default. (5 minut)

²⁴ https://www.w3schools.com/php/php_if_else.asp

²⁵ https://www.w3schools.com/php/php_if_operators.asp

²⁶ https://www.w3schools.com/php/php_if_nested.asp

²⁷ Soubor znamka-if.php umístěný ve složce list2 v příloze

²⁸ https://www.w3schools.com/php/php_switch.asp

²⁹ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_switch

5. **Aktivita** – cvičení, kde žáci převedou podmínkový kód z minulé úlohy 2.1 do přepínače switch – Úloha 2.2³⁰ (15 minut)

3. Cyklus while, do-while a for a řídicí příkazy break a continue

Tato oblast je zaměřena na seznámení žáků s cykly while, do-while a for v PHP, a s řídicími příkazy break a continue.

Rozsah: 40–60 minut

1. **Prezentace** a ukázky použití cyklu while³¹. (5 minut)
2. **Prezentace** a ukázky použití cyklu do-while, který proběhne vždy alespoň jednou³². (5 minut)
3. **Prezentace** a ukázky použití cyklu for³³. (5 minut)
4. **Vysvětlení a praktické ukázky** použití příkazů break a continue pro řízení toku v cyklech^{34 35}. (5 minut)
5. **Aktivita** – cvičení, kde žáci vytvoří program s podmínkami uvnitř cyklů, který využívá break a continue – Úloha 2.3³⁶ (20 minut)

Úlohy:

Úloha 2.1 – převod číselné hodnoty známky na slovní hodnocení

Tato úloha pomůže žákům procvičit si používání podmínek, větví elseif a else v PHP. Žáci se naučí, jak efektivně zpracovávat více podmínek a přiřazovat odpovídající slovní hodnocení na základě číselné známky.

Zadání úlohy 2.1:

Vytvořte skript v jazyce PHP, který převede hodnotu známky z proměnné na slovní hodnocení. V případě zadání jiné hodnoty než 1-5 vypište hodnotu „Neznámá známka“.

Pracovní postup:

³⁰ Soubor znamka-switch.php umístěný ve složce list2 v příloze

³¹ https://www.w3schools.com/php/php_looping_while.asp

³² https://www.w3schools.com/php/php_looping_do_while.asp

³³ https://www.w3schools.com/php/php_looping_for.asp

³⁴ https://www.w3schools.com/php/php_looping_break.asp

³⁵ https://www.w3schools.com/php/php_looping_continue.asp

³⁶ Soubor delitelnost.php umístěný ve složce list2 v příloze

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou `znamka`, do které nejprve uložte hodnotu 1. Vytvořte sérii podmínek `if` a `elseif`, které budou testovat, zda proměnná `znamka` obsahuje hodnotu 1 až 5 a v každém bloku vypíše její slovní podobu. Vytvořte závěrečnou větev `else`, která vypíše „Neznámá známka“, pokud v proměnné bude jiná hodnota než 1-5.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. `C:\xampp\htdocs`) pod názvem `znamka-if.php`. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se mělo zobrazit slovní hodnocení.

Správnost podmínek: Ověřte, že podmínky jsou správně nastavené a že větev `else` a `elseif` fungují podle očekávání pro různé hodnoty v proměnné `znamka`.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají zadaným známkám.

Úloha 2.2 - Tvorba přepínače switch v PHP pro převod číselné hodnoty známky na slovní hodnocení

Tato úloha navazuje na Úlohu 2.1 a pomůže žákům procvičit si používání přepínače `switch` v PHP. Žáci se naučí, jak efektivně zpracovávat různé hodnoty vstupní proměnné pomocí přepínače `switch` a přiřazovat odpovídající slovní hodnocení v jednotlivých větvích na základě číselné známky.

Zadání úlohy 2.2:

Vytvořte skript v jazyce PHP, který převede hodnotu známky z proměnné na slovní hodnocení pomocí přepínače `switch` a jednotlivých větví odpovídajícím hodnotě známky. V případě zadání jiné hodnoty než 1-5 vypíše hodnotu „Neznámá známka“.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou `znamka`, do které nejprve uložte hodnotu 1. Vytvořte přepínač `switch` a jednotlivé větve s hodnotami známky 1 až 5, které budou vypíšu její slovní podobu.

Vytvořte závěrečnou větev default, která vypíše „Neznámá známka“, pokud v proměnné bude jiná hodnota než 1-5.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem znamka-switch.php. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se mělo zobrazit slovní hodnocení.

Správnost přepínače: Ověřte, že větve ve switch jsou správně nastavené a že větev default funguje podle očekávání pro různé hodnoty v proměnné znamka.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají zadaným známkám.

Úloha 2.3 - Použití cyklu while pro výpis hodnoty celého kladného čísla, dělitelného číslem 3

Tato úloha pomůže žákům procvičit si používání cyklu while a řídicích příkazů break a continue uvnitř podmínek v PHP. Žáci se naučí, jak efektivně opakovat bloky kódu a řídit tok programu pomocí těchto příkazů.

Zadání úlohy 2.3:

Vytvořte skript v jazyce PHP, který postupně otestuje čísla v proměnné cislo 1-9, zda jsou dělitelné číslem 3. Pokud ne, navyšte hodnotu testované proměnné cislo o 1 a přeskočte zbytek iterace příkazem continue. Pokud je číslo dělitelné, vypište jej na obrazovku a navyšte hodnotu proměnné cislo o 1. Pokud testovaná proměnná dosáhne hodnoty 10, ukončete běh cyklu pomocí příkazu break. Použijte nekonečný cyklus while(true).

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou cislo, do které uložte hodnotu 1. Vytvořte nekonečný cyklus while(true) ve kterém pomocí podmínky otestujete, zda cislo je větší nebo rovno 10. Pokud ano, ukončete běh cyklu příkazem break. Pokud není, program bude pokračovat pod podmínkou. Dále otestujte dělitelnost čísla hodnotou 3 operátorem modulo. Pokud číslo není dělitelné číslem 3, navyšte jeho hodnotu o 1 operátorem inkrementace a přeskočte danou

iteraci cyklu pomocí continue. Pokud je číslo dělitelné hodnotou 3, pokračujte pod podmínkou výpisem hodnoty čísla, doplňte výpis o příkaz `<br>` a navyšte hodnotu proměnné číslo operátorem inkrementace.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem delitelnost.php. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měly zobrazit hodnoty kladného celého čísla dělitelného číslem 3 až po hodnotu 9.

Přeskočení iterace: Ověřte, že hodnoty, které nejsou dělitelné číslem 3, jsou správně přeskočené pomocí příkazu continue.

Ukončení cyklu: Ověřte, že cyklus je ukončený, pokud je hodnota čísla vyšší nebo rovna 10, pomocí příkazu break.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci. Doporučený čas jsou 3 vyučovací hodiny.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL

Možný chybný zápis operátoru rovnosti v podmínce pouze jedním znakem rovná se, tedy operátorem přiřazení.

Možný chybný konec bloku podmínky.

Možné chybné ukončení větví přepínače chybějícím příkazem break.

Možný vznik nekonečného cyklu a tím provádění skriptu do maximální doby vykonání serverem v případě chybně nastavených ukončujících podmínek.

5.2.3 Funkce a globální proměnné

Zaměření

Učivo ŠVP: Funkce, globální proměnné a pole

Časová dotace: 200–270 minut

Cíle

Obsah tohoto metodického listu je zaměřen na seznámení žáků s tvorbou a použitím funkcí a konceptem rekurze v jazyce PHP. Žáci se naučí vytvářet vlastní funkce, pracovat s lokálními a globálními proměnnými a implementovat rekurzi. Metodický list zahrnuje teoretické i praktické části, které umožní žákům získat základní znalosti ze zmíněné problematiky a aplikovat tyto znalosti v praxi.

Kompetence

Kognitivní: žák umí vytvářet vlastní funkce, zná rozdíly mezi lokálními a globálními proměnnými a umí vytvořit rekurzivní volání funkcí v jazyce PHP.

Praktické: žák používá vlastní funkce pro efektivní tvorbu opakujících se částí kódu, používá lokální i globální proměnné ve funkcích a vytváří rekurzivní funkce v případě potřeby jejich použití.

Obsah výuky:

- Tvorba vlastních funkcí
- Vstupní parametry funkcí
- Návrátové hodnoty
- Lokální a globální proměnné
- Rekurze

Organizační formy a metody výuky

Celý materiál je rozdělen do pěti oblastí, které na sebe navzájem navazují.

1. Tvorba funkcí, vstupní parametry a návratové hodnoty funkcí

Tato oblast je zaměřena na seznámení žáků s tvorbou a používáním funkcí v PHP. Žáci se naučí vytvářet vlastní funkce, přidávat a používat vstupní parametry, definovat výchozí hodnoty, vracet hodnoty a pracovat s více návratovými hodnotami.

Rozsah: 35–50 minut

1. **Prezentace** na vytváření funkcí v PHP, ukázky vzorových skriptů z vhodných stránek³⁷. (15 minut)
2. **Demonstrace**, jak přidat jeden vstupní parametr do funkce³⁸. (5 minut)
3. **Demonstrace**, jak přidat více vstupních parametrů a pracovat s nimi ve funkci³⁹. (5 minut)
4. **Praktická ukázka** použití výchozích hodnot pro parametry funkcí⁴⁰. (5 minut)
5. **Demonstrace**, jak vracet hodnoty z funkcí⁴¹. (5 minut)

2. Lokální a globální proměnné, praktický příklad

Tato oblast je zaměřena na pochopení rozdílu mezi lokálními a globálními proměnnými v PHP a jejich použití uvnitř funkcí. Žáci se naučí, jak lokální a globální proměnné vznikají, jaký je jejich rozsah platnosti a jak k nim přistupovat ve funkcích.

Rozsah: 35–50 minut

1. **Prezentace** na vznik a platnost lokálních proměnných, ukázky kódu⁴². (10 minut)
2. **Demonstrace**, jak přistupovat ke globálním proměnným uvnitř funkcí⁴³. (5 minut)
3. **Aktivita** – cvičení, kde žáci vytvoří funkci využívající jak lokální, tak globální proměnné – Úloha 3.1⁴⁴. (20 minut)

³⁷ https://www.w3schools.com/php/php_functions.asp

³⁸ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_function2

³⁹ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_function_return

⁴⁰ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_function_default_arg

⁴¹ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_function_return

⁴² <https://www.php.net/manual/en/language.variables.scope.php>

⁴³ https://www.w3schools.com/php/php_superglobals_globals.asp

⁴⁴ Soubor pozdrav.php umístěný ve složce list3 v příloze

3. Praktická tvorba vlastní funkce na základě předchozích příkladů

Tato oblast je zaměřena na praktickou tvorbu funkcí v PHP, které zjišťují základní vlastnosti celých čísel. Žáci se naučí vytvářet funkce, které určují, zda je číslo kladné, záporné či rovno nule, zda je číslo dělitelné jiným číslem, zda je sudé nebo liché, a zda je prvočíslem. Zároveň si procvičí návratové hodnoty funkcí.

Rozsah: 45–60 minut

1. **Aktivita 1** – žáci vytvoří funkce, které zjistí, zda je číslo kladné, záporné či rovno nule – Úloha 3.2⁴⁵. (15 minut)
2. **Aktivita 2** – žáci vytvoří funkce pro zjištění dělitelnosti – Úloha 3.3⁴⁶. (5 minut)
3. **Aktivita 3** – žáci vytvoří funkci pro určení, zda je číslo sudé nebo liché – Úloha 3.4⁴⁷. (5 minut)
4. **Aktivita 4** – žáci vytvoří funkci pro kontrolu, zda je číslo prvočíslem – Úloha 3.5⁴⁸. (20 minut)

4. Rekurze, příklad výpočtu funkce faktoriál

Tato oblast je zaměřena na pochopení konceptu rekurze a jejího použití v programování. Žáci se naučí, co je rekurze, jak funguje a jaké jsou její principy. Prakticky si vyzkouší použití rekurze při výpočtu faktoriálu čísla.

Rozsah: 40–50 minut

1. **Prezentace** konceptu rekurze a její použití v programování⁴⁹. (5 minut)
2. **Demonstrace**, jak funguje rekurze, ukázky kódu. (5 minut)
3. **Praktická ukázka** rekurzivní funkce pro výpočet faktoriálu a vysvětlení ukončujících podmínek. (10 minut)
4. **Aktivita** – žáci vytvoří rekurzivní volání funkce pro výpočet faktoriálu – Úloha 3.6⁵⁰. (20 minut)

⁴⁵ Funkce kladne v souboru funkce.php umístěný ve složce list3 v příloze

⁴⁶ Funkce delitelne v souboru funkce.php umístěný ve složce list3 v příloze

⁴⁷ Funkce sude v souboru funkce.php umístěný ve složce list3 v příloze

⁴⁸ Funkce prvočíslo v souboru funkce.php umístěný ve složce list3 v příloze

⁴⁹ [https://cs.wikipedia.org/wiki/Rekurze_\(programov%C3%A1n%C3%AD\)](https://cs.wikipedia.org/wiki/Rekurze_(programov%C3%A1n%C3%AD))

⁵⁰ Soubor faktorial.php umístěný ve složce list3 v příloze

5. Rekurze, příklad výpočtu Fibonacciho posloupnosti

Tato závěrečná oblast je zaměřena na představení Fibonacciho posloupnosti a její implementace pomocí rekurzivní funkce v PHP. Žáci se naučí, co je Fibonacciho posloupnost, jak zjistit hodnotu n -tého členu posloupnosti pomocí funkce, jak rekurze funguje a jaké jsou principy a složitosti rekurzivních algoritmů.

Rozsah: 45–60 minut

1. **Prezentace** na vysvětlení principu Fibonacciho posloupnosti⁵¹. (10 minut)
2. **Aktivita** – implementace Fibonacciho posloupnosti pomocí rekurzivní funkce – Úloha 3.7⁵². (10 minut)
3. **Diskuze** o tom, jak funkce funguje a proč nemusí být rekurze vhodná. (5 minut)
4. **Demonstrace** výpočtu vyšších členů Fibonacciho posloupnosti pomocí rekurzivní funkce z Úlohy 3.7. (10 minut)
5. **Diskuze** o složitosti rekurzivních algoritmů a jak optimalizovat výpočty⁵³. (10 minut)

Úlohy:

Úloha 3.1 – tvorba funkce pro vrácení řetězce s globálním jménem ze skriptu

Tato úloha pomůže žákům procvičit si tvorbu vlastních funkcí v PHP, práci s globálními proměnnými uvnitř funkcí a vrácení hodnot z funkcí. Žáci se naučí, jak efektivně číst a používat globální proměnné ve svých skriptech a vracet výsledky hodnoty z funkcí.

Zadání úlohy 3.1:

Vytvořte PHP skript, který obsahuje funkci pozdrav, která čte globální proměnnou jmeno ze skriptu a při zavolání vytvoří uvnitř řetězec pozdrav, který vrátí zpět do skriptu. Obsah návratové hodnoty vypište.

Pracovní postup:

⁵¹ https://cs.wikipedia.org/wiki/Fibonacciho_posloupnost

⁵² Soubor fibonacci.php umístěný ve složce list3 v příloze

⁵³ https://cw.fel.cvut.cz/b182/_media/courses/b6b36dsa/dsa-3-slozitostalgoritmu.pdf

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou `jmeno`, do které napište své jméno. Vytvořte funkci `pozdrav` bez vstupního parametru. Uvnitř funkce vytvořte lokální proměnnou `pozdrav`, do které napište text "Dobrý den, Vaše jméno je " a slučte tento řetězec s globální proměnnou `jmeno`. Tuto proměnnou použijte jako návratovou hodnotu funkce. Ve skriptu vypište návrat z funkce.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. `C:\xampp\htdocs`) pod názvem `pozdrav.php`. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit řetězec Dobrý den, Vaše jméno je Jan, pokud bylo zadáno jméno Jan do proměnné `jmeno`.

Správnost funkce: Ověřte, že funkce `pozdrav` správně čte globální proměnnou `jmeno` a vrací řetězec `$pozdrav`.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 3.2 – tvorba funkce pro zjištění, zda je číslo kladné, záporné či rovno nule

Tato úloha pomůže žákům procvičit si tvorbu vlastních funkcí v PHP, práci s podmínkami uvnitř funkcí a vracení hodnot z funkcí.

Zadání úlohy 3.2:

Vytvořte PHP skript, který obsahuje funkci `kladne` s jedním vstupním parametrem `cislo`. Funkce pomocí podmínky zjistí, zda je číslo kladné. Pokud ano, vrátí řetězec "kladné", pokud je záporné, vrátí řetězec "záporné" a pokud je rovno nule, vrátí řetězec "nula".

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte funkci `kladne` se vstupním parametrem `cislo`. Uvnitř funkce vytvořte podmínku, která zjistí, zda je proměnná `cislo` větší než nula, pokud ano, vraťte řetězec "kladné". Dále vytvořte větev `elseif`, která zjistí, zda je proměnná `cislo` menší než nula. Pokud ano, vraťte

řetězec "záporné". Vytvořte větev else, která vrátí řetězec "nula". Vyzkoušejte volání této funkce s různými hodnotami čísel, např. -10, -5, 0, 5, 10.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem funkce.php. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měly zobrazit výsledky, které ukazují, zda je číslo kladné, záporné nebo rovno nule.

Správnost funkce: Ověřte, že funkce kladně správně vyhodnocuje podmínky a vrací odpovídající řetězec.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 3.3 – tvorba funkce pro zjištění dělitelnosti celého čísla jiným

Tato úloha pomůže žákům procvičit si tvorbu vlastních funkcí v PHP, vyhodnocování dělitelnosti čísla a vracení hodnot true nebo false z funkcí.

Zadání úlohy 3.3:

Vytvořte PHP skript, který obsahuje funkci delitelne se dvěma vstupními parametry cislo a delitel. Funkce pomocí podmínky zjistí, zda je celé číslo dělitelné dělitelem beze zbytku. Pokud ano, vrátí řetězec hodnotu true, pokud ne, vrátí hodnotu false.

Pracovní postup:

Otevřete textový editor pro práci s PHP a otevřete předchozí soubor funkce.php.

Vytvořte další funkci delitelne se vstupními parametry cislo a delitel. Uvnitř funkce vytvořte podmínku, která zjistí, zda je proměnná cislo dělitelná proměnnou delitel pomocí operátoru modulo beze zbytku. Pokud ano, vraťte hodnotu true, jinak hodnotu false. Vyzkoušejte volání této funkce s různými hodnotami čísel a dělitelů pomocí funkce var_dump, např. (10, 2), (10, 3), (10, 5).

Uložte soubor a otevřete v URL prohlížeče pro ověření nové funkce delitelne.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měly zobrazit výsledky, které ukazují, zda je číslo dělitelné jiným číslem (true nebo false).

Správnost funkce: Ověřte, že funkce `delitelne` správně vyhodnocuje dělitelnost a vrátí hodnotu true nebo false.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 3.4 – tvorba funkce pro zjištění, zda je celé kladné číslo sudé či liché s využitím předchozích funkcí `kladne` a `delitelne`

Tato úloha pomůže žákům procvičit si tvorbu vlastních funkcí využívající návratové hodnoty z již dříve vytvořených vlastních funkcí v PHP.

Zadání úlohy 3.4:

Vytvořte PHP skript, který obsahuje funkci `sude` s jedním vstupním parametrem `cislo`. Funkce pomocí podmínky ověří kladnost čísla pomocí návratové hodnoty předchozí funkce `kladne`. V případě, že je číslo kladné, ověří, zda je číslo sudé pomocí předchozí metody `delitelne` s dělitelem 2. Pokud ano, vrátí řetězec s hodnotu true, pokud ne, vrátí hodnotu false. V případě jiného než kladného čísla vrátí text "chyba".

Pracovní postup:

Otevřete textový editor pro práci s PHP a otevřete předchozí soubor `funkce.php`.

Vytvořte další funkci `sude` se vstupním parametrem `cislo`. Uvnitř funkce vytvořte podmínku, která zjistí, zda je proměnná `cislo` kladná pomocí předchozí funkce `kladne`. Pokud ano, vytvořte vnořenou podmínku pro zjištění, zda je číslo sudé na základě předchozí metody `delitelne` s parametrem dělitele 2. Pokud ano, vrátí hodnotu true. Pokud není sudé, vrátí ve větvi `else` u vnořené podmínky hodnotu false. Pokud se nejedná o kladné číslo, vytvořte větev `else`, která vrátí řetězec "chyba". Vyzkoušejte volání této funkce s různými hodnotami čísel, např. -5, 1, 4, 7, 8.

Uložte soubor a otevřete v URL prohlížeče pro ověření nové funkce `sude`.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měly zobrazit výsledky, které ukazují, zda je číslo sudé nebo liché (true nebo false).

Správnost funkce: Ověřte, že funkce sude správně vyhodnocuje sudost a lichost čísla a vrací odpovídající hodnotu true, false či chybový řetězec.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 3.5 – tvorba funkce pro zjištění, zda je celé kladné číslo prvočíslem s využitím některých předchozích metod

Tato úloha pomůže žákům procvičit si tvorbu vlastních funkcí v PHP, práci s podmínkami a cykly uvnitř funkcí, volání vlastních funkcí uvnitř funkce a vracení hodnot z funkcí.

Zadání úlohy 3.5:

Vytvořte PHP skript, který obsahuje funkci prvocislo s jedním vstupním parametrem cislo. Funkce pomocí podmínky ověří kladnost čísla pomocí návratové hodnoty předchozí funkce kladne. Pokud není kladné, vrátí řetězec "chyba". V případě, že je číslo kladné, pomocí cyklu for ověří dělitelnost tohoto čísla všemi čísly od 2 až do druhé odmocniny hodnoty čísla. Pokud zjistí, že některý z dělitelů vrací hodnotu true, vrátí hodnotu false a tím zastaví další vyhodnocení. Pokud doběhne for do konce a nenajde dělitele, vrátíme true neboť toto číslo je prvočíslem.

Pracovní postup:

Otevřete textový editor pro práci s PHP a otevřete předchozí soubor funkce.php.

Vytvořte další funkci prvocislo se vstupním parametrem cislo. Uvnitř funkce vytvořte podmínku, která zjistí, zda je proměnná cislo jiná než kladná pomocí předchozí funkce kladne. Pokud ano, zastavte vyhodnocení návratovou hodnotou "chyba". Pokud je kladné, vytvořte cyklus for s počáteční hodnotou iterované proměnné i od 2. Cyklus půjde s krokem +1 až do hodnoty druhé odmocniny proměnné cislo. Použijte knihovní funkci sqrt. Pomocí podmínky testujte, zda proměnná cislo je dělitelná proměnnou i. Pokud ano, vraťte hodnotu false pro zastavení dalšího vyhodnocení. Pokud skončí cyklus bez návratové hodnoty

funkce, vraťte hodnotu true, protože zadané číslo nemá dělitele, tedy je prvočíslem. Vyzkoušejte volání této funkce s různými hodnotami čísel, např. -5, 1, 2, 4, 7, 9.

Uložte soubor a otevřete jej v URL prohlížeče pro ověření nové funkce sude.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měly zobrazit výsledky, které ukazují, zda je číslo prvočíslem (true nebo false).

Správnost funkce: Ověřte, že funkce prvocislo správně vyhodnocuje prvočísla a vrací hodnotu true nebo false.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 3.6 – výpočet faktoriálu čísla pomocí rekurze

Tato úloha pomůže žákům procvičit si tvorbu rekurzivních funkcí v PHP, práci s podmínkami uvnitř funkcí a vracení hodnot z funkcí. Žáci se naučí, vytvářet a používat rekurzivní funkce pro výpočet faktoriálu čísel.

Zadání úlohy 3.6:

Vytvořte PHP skript, který obsahuje funkci faktorial s jedním vstupním parametrem cislo. Funkce zjistí pomocí podmínky, zda hodnota proměnné cislo není záporná. Pokud ano, vrátí řetězec "chyba". Pokud je číslo 0 nebo 1, vrátí nám hodnotu 1. V každém dalším případě vrátí hodnotu čísla násobenou voláním funkce sebe samotné s hodnotou o jedno vstupního parametru o jedno menším.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte funkci faktorial s jedním vstupním parametrem cislo. Vytvořte podmínku, která ověří, zda je hodnota proměnné cislo menší než nula. Pokud ano, vraťte řetězec "chyba". Dále vytvořte další podmínku, která ověří, zda je hodnota proměnné cislo menší nebo rovna jedné, tedy je 0 nebo 1. V obou případech vrátí hodnotu 1. Pokud je číslo větší než 1, vrátí aktuální hodnotu proměnné cislo vynásobenou o zavolání funkce sebe samotné se vstupním

parametrem `cislo - 1`. Proveďte výpis návratové hodnoty po zavolání funkce faktoriál s různými hodnotami vstupní proměnné `cislo`.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit výsledek, který ukazuje faktoriál zadaného čísla.

Správnost funkce: Ověřte, že funkce `faktorial` správně vyhodnocuje faktoriál čísla pomocí rekurze a vrací správný výsledek.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 3.7 – výpočet n-tého členu Fibonacciho posloupnosti pomocí rekurze

Tato úloha taktéž pomůže žákům procvičit si tvorbu rekurzivních funkcí v PHP, práci s podmínkami uvnitř funkcí a vracení hodnot z funkcí. Oproti funkci faktoriál si na této funkci později mohou ukázat složitost výpočtu například 40. či 50. členu posloupnosti.

Zadání úlohy 3.7:

Vytvořte PHP skript, který obsahuje funkci `fibonacci` s jedním vstupním parametrem `cislo`. Funkce zjistí pomocí podmínky, zda hodnota proměnné `cislo` není záporná. Pokud ano, vrátí řetězec "chyba". Pokud je hodnota proměnné `cislo` rovna 0 vrátí hodnotu 0, pokud je hodnota rovna 1, vrátí hodnotu 1. V každém dalším případě vrátí návratovou hodnotu po výsledku volání sama sebe s hodnotou vstupního parametru `cislo - 1` se součtem návratové hodnoty po výsledku volání sama sebe s hodnotou vstupního parametru `cislo - 2`. Vyzkoušejte tuto funkci pro různé hodnoty vstupních čísel, nejvýše do hodnoty 40.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte funkci `fibonacci` s jedním vstupním parametrem `cislo`. Vytvořte podmínku, která ověří, zda je hodnota proměnné `cislo` menší než nula. Pokud ano, vraťte řetězec "chyba". Dále vytvořte další podmínku, která ověří, zda je hodnota proměnné `cislo` rovna nule, pokud ano, vraťte hodnotu 0. Vytvořte další podmínku, která ověří, zda je hodnota proměnné `cislo` rovna jedné, pokud ano, vraťte hodnotu 1. V každém dalším případě vrátí návratovou

hodnotu po výsledku volání sama sebe s hodnotou vstupního parametru $cislo - 1$ se součtem návratové hodnoty po výsledku volání sama sebe s hodnotou vstupního parametru $cislo - 2$ ho. Proveďte výpis návratové hodnoty po zavolání funkce fibonacci s různými hodnotami vstupní proměnné $cislo$, nejvýše však hodnoty 40.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit výsledek, který ukazuje n -tý člen Fibonacciho posloupnosti.

Správnost funkce: Ověřte, že funkce fibonacci správně generuje Fibonacciho posloupnost pomocí rekurze a vrací správný výsledek.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci. Doporučený čas je 5 vyučovacích hodin.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL

Možný chybný zápis funkce, vstupních parametrů oddělených jiným znakem než čárkou.

Možný chybný zápis návratových hodnot, případné ukončení dříve, než bylo očekáváno.

Možný chybný konec bloku kódu funkce.

Možné chybné ukončující podmínky při rekurzivním volání funkcí vedoucí v nekonečné volání funkce.

5.2.4 Pole

Zaměření

Učivo ŠVP: Funkce, globální proměnné a pole

Časová dotace: 145–205 minut

Cíle

Obsah tohoto metodického listu je zaměřen na seznámení žáků s poli v jazyce PHP. Žáci se naučí vytvářet indexované a asociativní pole, vybírat prvky, modifikovat prvky v poli, cyklicky procházet oba druhy polí a také pracovat s vícerozměrnými poli. Metodický list zahrnuje teoretické i praktické části, které umožní žákům získat základní znalosti ze zmíněné problematiky a aplikovat tyto znalosti v praxi.

Kompetence

Kognitivní: žák umí vytvářet indexované a asociativní pole, umí vybírat prvky z pole, modifikovat hodnoty prvků v poli a vkládat nové prvky do existujícího pole. Zná tvorbu a práci s vícerozměrnými poli v jazyce PHP.

Praktické: žák používá indexované, asociativní a vícerozměrné pole pro praktickou práci s daty.

Obsah výuky:

- Druhy polí v PHP
- Vytváření pole
- Výběr hodnot
- Vkládání a modifikace
- Cyklický výpis obsahu pole

Organizační formy a metody výuky

Celý materiál je rozdělen do čtyř oblastí, které na sebe navzájem navazují.

1. Indexované a asociativní pole v PHP, jejich vytváření, zjištění velikosti, výpis pomocí print_r

Tato hodina je zaměřena na pochopení různých druhů polí v PHP a jejich praktické použití. Žáci se naučí rozlišovat mezi indexovanými a asociativními poli, vytvářet pole různými způsoby, zjistit jejich velikost a vypisovat jejich obsah.

Rozsah: 40–55 minut

1. **Demonstrace** rozdílu mezi indexovanými a asociativními poli^{54 55}. (5 minut)
2. **Praktické ukázky**, jak vytvářet pole pomocí různých způsobů⁵⁶. (10 minut)
3. **Praktické ukázky**, jak zjistit velikost pole pomocí funkce count⁵⁷. (5 minut)
4. **Praktické ukázky**, jak vypsát obsah pole funkcí print_r⁵⁸ a vysvětlení rozdílů oproti funkci var_dump⁵⁹. (5 minut)
5. **Aktivita** – praktické cvičení pro žáky na tvorbu pole – Úloha 4.1⁶⁰. (15 minut)

2. Výběr prvků z pole, modifikace prvků v poli a vkládání prvků do pole

Tato oblast je zaměřena na praktickou práci s poli v PHP. Žáci se naučí vybírat a měnit hodnoty v polích, vkládat nové prvky do indexovaných i asociativních polí a pracovat s indexy a klíči.

Rozsah: 40–55 minut

1. **Aktivita** – praktické cvičení, kde žáci vyberou a vypíší konkrétní hodnoty z polí – Úloha 4.2⁶¹. (15 minut)
2. **Praktické ukázky**, jak změnit hodnotu na specifické pozici v indexovaném poli⁶². (5 minut)
3. **Praktické ukázky**, jak změnit hodnotu na určitém klíči v asociativním poli. (5 minut)

⁵⁴ https://www.w3schools.com/php/php_arrays_indexed.asp

⁵⁵ https://www.w3schools.com/php/php_arrays_associative.asp

⁵⁶ https://www.w3schools.com/php/php_arrays_create.asp

⁵⁷ https://www.w3schools.com/php/func_array_count.asp

⁵⁸ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_var_print_r

⁵⁹ https://www.w3schools.com/php/func_var_var_dump.asp

⁶⁰ Soubor pole.php umístěný ve složce list4 v příloze

⁶¹ Soubor pole.php umístěný ve složce list4 v příloze doplněný o výpis hodnot z pole

⁶² https://www.w3schools.com/php/php_arrays_update.asp

4. **Praktické ukázky**, jak přidávat nové prvky do indexovaného pole⁶³. (5 minut)
5. **Praktické ukázky**, jak přidat prvek s konkrétním indexem. (5 minut)
6. **Praktické ukázky**, jak přidat prvek s konkrétním klíčem do asociativního pole⁶⁴. (5 minut)

3. Cyklický výpis polí pomocí cyklu foreach

Tato oblast je zaměřena na práci s cyklem foreach v PHP, který je klíčový pro efektivní výpis prvků z polí, zejména asociativních. Žáci se naučí, jak správně vypisovat obsah různých typů polí.

Rozsah: 30-45 minut

1. **Diskuze** o problémech spojených s výpisem polí a jak je řešit. (10 minut)
2. **Praktická ukázka** syntaxe a použití cyklu foreach⁶⁵.
3. **Demonstrace**, jak vypsat všechny prvky indexovaného pole pomocí foreach⁶⁶.
4. **Praktické ukázky**, jak vypsat klíče a hodnoty asociativního pole pomocí foreach⁶⁷.
5. **Aktivita – praktické cvičení**, kde žáci vypisují klíče/indexy a hodnoty z různých typů polí – Úloha 4.3⁶⁸.

4. Vícerozměrné pole

Tato oblast je zaměřena na pochopení konceptu vícerozměrných (vícedimenzionálních) polí v PHP a jejich praktické využití. Žáci se naučí vytvářet vícerozměrná pole, přistupovat k jejich prvkům, zjistit jejich velikost a manipulovat s jejich obsahem.

Rozsah: 35–50 minut

1. **Prezentace** konceptu vícerozměrných polí a jejich využití⁶⁹. (5 minut)
2. **Praktická ukázka**, jak vytvořit vícerozměrné pole a vkládat do něj další pole⁷⁰. (5 minut)

⁶³ https://www.w3schools.com/php/php_arrays_add.asp

⁶⁴ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_arrays_add_assoc

⁶⁵ https://www.w3schools.com/php/php_looping_foreach.asp

⁶⁶ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_loop_foreach

⁶⁷ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_foreach_key_value

⁶⁸ Soubor foreach.php umístěný ve složce list4 v příloze

⁶⁹ https://www.w3schools.com/php/php_arrays_multidimensional.asp

⁷⁰ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_multi

3. **Praktické ukázky**, jak přistupovat k jednotlivým prvkům ve vícerozměrném poli⁷¹. (5 minut)
4. **Demonstrace**, jak zjistit velikost hlavního pole i vnořených polí pomocí funkce count⁷². (5 minut)
5. **Aktivita** – praktické cvičení, kde žáci vytvoří vícerozměrné pole a manipulují s jeho obsahem – Úloha 4.4⁷³. (15 minut)

Úlohy:

Úloha 4.1 – tvorba pole

Tato úloha pomůže žákům procvičit si tvorbu indexovaných a asociativních polí v PHP a výpis jejich obsahu pomocí funkce print_r.

Zadání úlohy 4.1:

Vytvořte PHP skript, ve kterém dle předchozích ukázek vytvořte indexované pole a asociativní pole pomocí funkce array s alespoň třemi prvky. Obě pole vypište pomocí funkce print_r.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou indexovane do které uložíte nové indexované pole pomocí funkce array. Vložte do tohoto pole alespoň tři prvky, například názvy barev – červená, zelená a modrá.

Vytvořte proměnnou asociativni do které uložíte nové asociativní pole pomocí funkce array. Vložte do tohoto pole alespoň tři prvky, například názvy barev – červená s klíčem red, zelená s klíčem green a modrá s klíčem blue.

Obě pole postupně vypište funkcí print_r. Za každý výpis vypište značku
.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem pole.php. Tento soubor otevřete v URL prohlížeče.

⁷¹ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_multi

⁷² https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_count2

⁷³ Soubor vicerozmerne-pole.php umístěný ve složce list4 v příloze

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit obsah indexovaného a asociativního pole.

Správnost funkce: Ověřte, že funkce `print_r` správně vypisuje obsah obou typů polí.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 4.2 – rozšíření předchozí úlohy

Tato úloha pomůže žákům rozšířit znalosti z práce s indexovanými a asociativními poli v PHP a výpis specifických prvků z těchto polí.

Zadání úlohy 4.2:

Doplňte předchozí skript z úlohy 4.1 o výpis nadpisu úrovně h2 s nápisem Úloha 2 a doplňte výpis hodnoty na indexu 1 v indexovaném poli a o výpis hodnoty v klíči blue v asociativním poli. Výpisy oddělte tagem `<br>`.

Pracovní postup:

Otevřete textový editor pro práci s PHP a otevřete předchozí soubor `pole.php`.

Vypište nadpis pomocí tagu `<h2>` s textem Úloha 2. Poté pomocí funkce `echo` vypište hodnotu na indexu 1 v indexovaném poli z předchozí úlohy. Vypište značku `<br>`. Poté vypište hodnotu s klíčem blue z asociativního pole z předchozí úlohy. Opět vypište značku `<br>`.

Uložte soubor a otevřete jej v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit oproti předchozí úloze navíc nadpis Úloha 2 a prvek na pozici 1 z indexovaného pole a prvek s klíčem "blue" z asociativního pole.

Správnost přístupu: Ověřte, že přístup k prvkům v obou typech polí je správně proveden.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 4.3 – cyklický výpis pole včetně výpisu indexu či klíče

Tato úloha pomůže žákům procvičit si práci s cyklem foreach v PHP a výpis prvků indexovaných i asociativních polí. Žáci se naučí, jak efektivně iterovat přes prvky polí a jak je vypsat včetně hodnot indexů či klíčů.

Zadání úlohy 4.3:

Doplňte předchozí skript z úlohy 4.1 a úlohy 4.2 o výpis nadpisu druhé úrovně s textem Úloha 3 a o cyklický výpis indexovaného a asociativního pole včetně hodnot indexů či klíčů pro každou položku. Mezi jednotlivé výpisy umístěte značku
.

Pracovní postup:

Otevřete textový editor pro práci s PHP a otevřete předchozí soubor pole.php.

Vypište nadpis pomocí tagu <h2> s textem Úloha 3. Poté pomocí cyklu foreach vypište jednotlivé položky v indexovaném poli ve tvaru index – hodnota a doplňte výpis o značku
. Stejným způsobem proveďte cyklický výpis obsahu asociativního pole cyklem foreach ve tvaru klíč – hodnota a doplňte o značku
.

Uložte soubor a otevřete jej v URL prohlížeče.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měl zobrazit obsah indexovaného a asociativního pole ve tvaru index – hodnota a klíč – hodnota vždy na nové řádce.

Správnost cyklu foreach: Ověřte, že cyklus foreach správně iteruje přes prvky obou typů polí a vypisuje jejich obsah.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 4.4 – vícerozměrné pole

Tato úloha pomůže žákům procvičit si tvorbu asociativních a indexovaných polí v PHP, práci s vícerozměrnými poli, výpis struktury pole pomocí funkce print_r, manipulaci s prvky pole a zjištění velikosti pole pomocí funkce count.

Zadání úlohy 4.4:

Vytvořte asociativní pole jménem barvy, ve kterém vytvořte klíč red a do něj vložte indexované pole s hexa kódem světlé a tmavé červené barvy. Vytvořte klíč green a do něj vložte indexované pole se hexa kódem světlé a tmavě zelené barvy. Stejným způsobem vytvořte klíč blue a do něj uložte indexované pole s hexakódem světlé a tmavě modré barvy.

Vypište strukturu pole barvy pomocí funkce print_r.

Vypište kód barvy na pozici \$barvy["green"][0] a tuto barvu použijte pro obarvení tohoto textu.

Vypište velikost pole barvy a celkovou velikost všech prvků pomocí funkce count.

Výpisy proveďte vždy na novou řádku.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem vicerozmerne-pole.php. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte pole barvy funkcí array, do klíče red vytvořte nové pole funkcí array a uložte do něj hexa kódy pro světlé červenou použijte například #FF6666, pro tmavě červenou například #8B0000. Poté vytvořte klíč green a do něj nové pole funkcí array s hodnotami #90EE90 pro světlé zelenou a #006400 pro tmavě zelenou. Poslední klíč bude blue a do něj vytvořte nové pole funkcí array s hodnotami #ADD8E6 pro světlé modrou a #00008B pro tmavě modrou. Vypište pole barvy funkcí print_r. Proveďte výpis bloku
 a poté bloku a nastavte mu styl barvy písma dle hodnoty na pozici hexakódu \$barvy["green"][0] a do tohoto bloku vypište i samotný hexa kód.

Na nové řádky vypište poté velikost buněk v poli barvy a celkovou velikost všech buněk včetně vnořených polí pomocí funkce count.

Metoda ověření:

Zobrazení struktury pole: Po otevření URL v prohlížeči by se měla zobrazit struktura pole barvy pomocí funkce print_r.

Obarvený text: Ověřte, že se zobrazí text obarvený světle zelenou barvou z pole `$barvy["green"][0]`.

Velikost pole: Ověřte, že se správně zobrazí velikost pole barvy a celková velikost všech prvků pomocí funkce `count`.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci.

Doporučený čas jsou 4 vyučovací hodiny.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL

Možný chybný zápis funkce pro tvorbu pole.

Možný chybný zápis indexů či klíčů pole.

5.2.5 Textové soubory, řetězce

Zaměření

Učivo ŠVP: Textové soubory, řetězce

Časová dotace: 215–285 minut

Cíle

Obsah tohoto metodického listu je zaměřen na seznámení žáků s manipulací s externími textovými soubory a zpracování řetězců. Žáci se naučí číst a zapisovat data do textových souborů a manipulovat s textovými řetězci.

Kompetence

Kognitivní: žák umí pracovat s knihovními funkcemi pro práci s textovými soubory a umí pracovat s řetězci.

Praktické: žák používá knihovní funkce k práci s textovými soubory pro načítání a ukládání dat, skládá složitější řetězce z vícero proměnných a textu, vytváří podřetězce a převádí znaky v řetězcích na malé či velké

Obsah výuky:

- Funkce pro čtení obsahu textových souborů
- Funkce pro zápis do textových souborů
- Připojení externích souborů
- Skládání a dělení řetězců
- Vytváření podřetězců
- Převod malých a velkých znaků v řetězci
- Praktické příklady na práci s řetězci a jejich ukládání a načítání do externích souborů

Organizační formy a metody výuky

Celý materiál je rozdělen do pěti oblastí, které na sebe navzájem navazují.

1. Knihovní funkce fopen, fread a fclose

Tato oblast je zaměřena na práci se soubory v PHP. Žáci se naučí, jak otevřít, číst, zjistit velikost a zavřít soubor pomocí funkcí jako fopen, fread, filesize a fclose.

Rozsah: 40–50 minut

1. **Prezentace**, jak použít funkci fopen k otevření souboru pro čtení⁷⁴. (10 minut)
2. **Demonstrace** použití funkce fread a její parametry pro čtení dat ze souboru⁷⁵. (5 minut)
3. **Demonstrace**, jak použít funkci filesize pro získání velikosti souboru⁷⁶. (5 minut)
4. **Demonstrace**, jak zavřít soubor pomocí funkce fclose⁷⁷. (5 minut)
5. **Aktivita** – praktické cvičení, kde žáci načtou obsah textového souboru a vypíší jej na obrazovku – Úloha 5.1⁷⁸. (10 minut)

⁷⁴ https://www.w3schools.com/php/func_filesystem_fopen.asp

⁷⁵ https://tryphp.w3schools.com/showphp.php?filename=demo_func_filesystem_fread2

⁷⁶ https://www.w3schools.com/php/func_filesystem_filesize.asp

⁷⁷ https://www.w3schools.com/php/func_filesystem_fclose.asp

⁷⁸ Soubor soubor-cteni.php umístěný ve složce list5 v příloze

2. Parametry funkce fopen pro různé způsoby otevření souborů a zápis do souboru funkcí fwrite

Tato oblast je zaměřena na práci se soubory v PHP, konkrétně na otevírání souborů pro čtení a zápis, použití funkce fwrite pro zápis dat, a na rozdíly mezi zápisem na začátek a na konec souboru.

Rozsah: 40–55 minut

1. **Prezentace** na parametry funkce fopen pro otevření souboru pro čtení nebo zápis⁷⁹. (10 minut)
2. **Demonstrace** použití funkce fwrite pro zápis dat do souboru⁸⁰. (5 minut)
3. **Demonstrace**, jak zapisovat data na začátek souboru a přepisovat existující obsah⁸¹. (5 minut)
4. **Demonstrace**, jak zapisovat data na konec souboru. (5 minut)
5. **Aktivita** – cvičení, kde žáci zapíší data do souboru a zkontrolují výsledky – Úloha 5.2⁸². (15 minut)

3. Řetězce, jejich skládání a dělení

Tato oblast je zaměřena na práci s řetězci v PHP. Žáci se naučí, jak spojovat více řetězců dohromady, vkládat proměnné do řetězců, používat funkci explode pro rozdělení řetězce na pole, dělit řetězce na pole podle určitých znaků s praktickým příkladem dělení odstavce textu na jednotlivé věty.

Rozsah: 40–55 minut

1. **Demonstrace**, jak spojovat více řetězců dohromady⁸³. (5 minut)
2. **Demonstrace**, jak vkládat proměnné do řetězců. (5 minut)
3. **Prezentace** funkce explode pro rozdělení řetězce na pole⁸⁴. (10 minut)

⁷⁹ <https://www.php.net/manual/en/function.fopen.php>

⁸⁰ https://www.w3schools.com/php/func_filesystem_fwrite.asp

⁸¹ <https://www.php.net/manual/en/function.fwrite.php>

⁸² Soubor soubor-zapis.php umístěný ve složce list5 v příloze

⁸³ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_oper_string1

⁸⁴ https://www.w3schools.com/php/func_string_explode.asp

4. **Demonstrace**, jak rozdělit řetězec na pole podle určitého znaku⁸⁵. (5 minut)
5. **Aktivita** – praktické cvičení, kde žáci rozdělí odstavec textu na jednotlivé věty – Úloha 5.3⁸⁶. (15 minut)

4. Podřetězce a převod znaků

Tato oblast je zaměřena na práci s řetězci v PHP, zejména na práci s funkcemi pro manipulaci s podřetězci, převod znaků a problematiku diakritiky. Žáci se naučí používat funkce `substr` a `mb_substr`, převádět řetězce na malá a velká písmena a pracovat s kapitálkami.

Rozsah: 40–55 minut

1. **Prezentace** funkce `substr` pro výběr části řetězce a její parametry⁸⁷. (5 minut)
2. **Diskuze** o problémech s diakritikou při práci s řetězci. (5 minut)
3. **Demonstrace** použití funkce `mb_substr` pro práci s řetězci s diakritikou⁸⁸. (5 minut)
4. **Demonstrace**, jak převádět řetězce na malá a velká písmena a diskuze o problémech s diakritikou^{89 90 91 92}. (5 minut)
5. **Demonstrace**, jak převést text na kapitálky⁹³. (5 minut)
6. **Aktivita** – úloha zaměřená na použití funkcí pro práci s podřetězci a převodem znaků – Úloha 5.4⁹⁴. (15 minut)

5. Praktické příklady na práci s řetězci a jejich ukládání a čtení z textových souborů

Tato závěrečná oblast je zaměřena na vytváření vlastních funkcí pro čtení a zápis řetězců ze souborů v PHP. Žáci se naučí, jak ověřit existenci souboru, číst a vracet obsah souboru, a zapisovat data na začátek či konec souboru. Navíc implementují návratové hodnoty indikující úspěšnost operace.

⁸⁵ https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_string_explode

⁸⁶ Soubor `deleni-taxtu.php` umístěný ve složce `list5` v příloze

⁸⁷ https://www.w3schools.com/php/func_string_substr.asp

⁸⁸ <https://www.php.net/manual/en/function.mb-substr.php>

⁸⁹ <https://www.php.net/manual/en/function.strtolower.php>

⁹⁰ <https://www.php.net/manual/en/function.strtoupper.php>

⁹¹ <https://www.php.net/manual/en/function.mb-strtolower.php>

⁹² <https://www.php.net/manual/en/function.mb-strtoupper.php>

⁹³ <https://www.php.net/manual/en/function.ucfirst.php>

⁹⁴ Soubor `textove-funkce.php` umístěný ve složce `list5` v příloze

Rozsah: 55–70 minut

Během následujících aktivit bude postupně vytvářen soubor obsahující funkce pro čtení a zápis do textových souborů. Celá oblast je formou aktivity, vedoucí k postupnému rozšíření této souhrnné úlohy – Úloha 5.5⁹⁵.

1. **Aktivita 1** – žáci vytvoří funkci pro čtení řetězce ze souboru, která zahrnuje vstupní parametr, kterým je název souboru. (10 minut)
2. **Aktivita 2** – žáci implementují kontroly existence souboru a návrat chyby v případě, že soubor neexistuje. (5 minut)
3. **Aktivita 3** – žáci rozšíří funkci obsahující načtení souboru o vrácení celého obsahu souboru na základě parametru. (5 minut)
4. **Aktivita 4** – žáci vytvoří funkci pro zápis řetězce do souboru, která zahrnuje tři vstupní parametry, kterými jsou název souboru, řetězec a parametr pro zápis na konec nebo začátek souboru. (15 minut)
5. **Aktivita 5** – žáci implementují logiku pro zápis na začátek nebo konec souboru podle hodnoty parametru. (5 minut)
6. **Aktivita 6** – žáci implementují návratovou hodnotu, která indikuje úspěšnost nebo neúspěšnost operace (true nebo false). (5 minut)

Úlohy v tomto metodickém listu:

Úloha 5.1 – načtení a výpis textového souboru

Tato úloha pomůže žákům procvičit si práci se soubory v PHP pomocí funkcí fopen, fread, filesize a fclose. Žáci se naučí, jak otevřít, načíst, zjistit velikost a zavřít soubor, a jak vypsat jeho obsah.

Zadání úlohy 5.1:

Vytvořte skript pro načtení externího textového souboru data.txt. K načtení použijte knihovní funkci fopen, fread, filesize a fclose. Pro řádkování je možno použít funkci nl2br. Obsah souboru vypište na obrazovku.

Pracovní postup:

⁹⁵ Soubor funkce-soubory.php umístěný ve složce list5 v příloze

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou `nazevSouboru` pro název souboru. Vytvořte referenci na soubor otevřený funkcí `fopen` s parametrem `"r"` pro čtení. Do proměnné obsah uložte celý obsah pomocí funkce `fread`. Velikost souboru zjistěte funkcí `filesize`. Soubor zavřete funkcí `fclose`. Při provádění výpisu funkce obsah použijte funkci `nl2br` s parametrem `false` pro automatické vložení tagu `<br>` na konec řádku.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. `C:\xampp\htdocs`) pod názvem `soubor-cteni.php`. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení obsahu souboru: Po otevření URL v prohlížeči by se měl zobrazit obsah textového souboru `data.txt`.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 5.2 – zápis řetězce do textového souboru

Tato úloha pomůže žákům procvičit si práci se soubory v PHP pomocí funkcí `fopen`, `fwrite` a `fclose`. Žáci se naučí, jak otevřít soubor pro zápis, zapsat do něj řetězec a zavřít soubor.

Zadání úlohy 5.2:

Proveďte zápis proměnné text obsahující libovolný text (možno lorem ipsum) do souboru `data2.txt` pomocí funkce `fwrite`. Použijte parametr `"w"` pro otevření souboru. Výsledek po zápisu vypište na obrazovku funkcí `var_dump`.

Pracovní postup:

Otevřete textový editor pro práci s PHP a otevřete předchozí soubor `pole.php`.

Vytvořte proměnnou `nazevSouboru` pro název soubor `data2.txt`. Vytvořte referenci na soubor otevřený funkcí `fopen` s parametrem `"w"` pro zápis. Do proměnné text uložte například jeden odstavec lorem ipsum. Zapište text do souboru funkcí `fwrite` a výsledek zápisu uložte do proměnné `vysledek`. Soubor zavřete funkcí `fclose`. Proveďte výpis proměnné `vysledek` pomocí funkce `var_dump`.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem soubor-zapis.php. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Zobrazení výsledku: Po otevření URL v prohlížeči by se měl zobrazit obsah proměnné výsledek, který obsahuje počet znaků, které funkce zapsala v případě úspěšného zápisu do souboru.

Kontrola obsahu souboru: Otevřete soubor data2.txt, který měl být vytvořen ve stejné složce, jako je PHP soubor a ověřte, že obsahuje zapsaný text z proměnné text.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a výsledky odpovídají očekávaným hodnotám.

Úloha 5.3 – dělení odstavce

Tato úloha pomůže žákům procvičit si práci s funkcí explode v PHP pro rozdělení textového obsahu na jednotlivé části. Žáci se naučí, jak efektivně rozdělovat text na věty a pracovat s oddělovači.

Zadání úlohy 5.3:

Vytvořte proměnnou text obsahující několik vět textu. Věty oddělte tečkou. Pomocí funkce explode rozdělte tento odstavec na jednotlivé věty do pole. Toto pole poté cyklicky vypište, doplňte výpis o tečky na konci vět a tag
. Nevypisujte prázdná data.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem deleni-textu.php. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou text s několika věty v odstavci. Pomocí funkce explode s parametrem dělitele "." rozdělte proměnnou text. Výsledek uložte do pole vety. Pomocí cyklu foreach načtěte jednotlivé hodnoty z tohoto pole a proveďte jejich výpis doplněný o tečku a tag
. Jelikož se očekává, že poslední věta končí tečkou, funkce explode provede dělení i na konci a tím vytvoří další hodnotu v poli, která bude prázdná. Pomocí funkce empty lze zjistit, zda je hodnota prázdná. Vypíšeme tedy jen neprázdné věty. Zároveň mezi větami je vždy tečka

a mezera. Mezera bude přiřazena tímto způsobem na začátek další věty. Tyto „bílé znaky“ lze odstranit pomocí funkce trim.

Metoda ověření:

Zobrazení rozdělených vět: Po otevření URL v prohlížeči by se měly zobrazit jednotlivé věty zadaného odstavce textu, každá věta na novém řádku zakončená tečkou.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a správně rozděluje odstavec textu na věty.

Úloha 5.4 – funkce na vytváření podřetězce a převod znaků

Tato úloha pomůže žákům procvičit si práci s funkcemi substr a mb_substr pro výběr části řetězce, převádět řetězce na malá a velká písmena pomocí funkcí strtolower a strtoupper a funkcí mb_strtolower a mb_strtoupper.

Zadání úlohy 5.4:

Vytvořte proměnnou text obsahující text například „Řetězec s diakritikou“. Vypište podřetězec pomocí funkce substr s prvními pěti znaky a odřádkujte. Stejný výpis proveďte funkcí mb_substr odřádkujte a výsledky porovnejte.

Vypište obdobným způsobem řetězec převedený na malé znaky funkcí strtolower a na novou řádku pomocí mb_strtolower. Opět porovnejte rozdíly.

Na závěr vypište obdobným způsobem řetězec převedený na velké znaky funkcí strtoupper a na novou řádku pomocí mb_strtoupper. Opět porovnejte rozdíly.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem deleni-textu.php. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte proměnnou text obsahující text například „Řetězec s diakritikou“.

Vypište výsledek funkce substr(\$text, 0, 5) a doplňte o znak
.

Vypište výsledek funkce mb_substr(\$text, 0, 5) a doplňte o znak
.

Vypište výsledek funkce `strtolower($text)` a dopňte o znak `<br>`.

Vypište výsledek funkce `mb_strtolower($text)` a dopňte o znak `<br>`.

Vypište výsledek funkce `strtoupper($text)` a dopňte o znak `<br>`.

Vypište výsledek funkce `mb_strtoupper($text)` a dopňte o znak `<br>`.

Metoda ověření:

Zobrazení výsledků: Po otevření URL v prohlížeči by se měly zobrazit výsledky použití funkcí `substr` a `mb_substr`, převodu na malá a velká písmena funkcí `strtolower`, `mb_strtolower`, `strtoupper` a `mb_strtoupper`.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a správně provádí všechny operace s řetězcem.

Úloha 5.5 – vlastní funkce pro čtení a zápis do souboru

Tato úloha pomůže žákům procvičit si tvorbu vlastních funkcí pro čtení a zápis do souborů v PHP, včetně ověřování existence souboru, načítání obsahu, a zápisu dat na začátek nebo konec souboru.

Zadání úlohy 5.5:

Vytvořte vlastní funkci `cteni` s parametrem obsahující název textového souboru pro čtení. Pomocí funkce `file_exists` ověřte, že soubor existuje, pokud ne, vraťte hodnotu `false`. Pokud existuje, ověřte, že délka tohoto souboru není nulová, pokud ano, vraťte hodnotu `false`. Pokud má soubor zapsaná data, načtěte celý obsah na základě Úlohy 1 a vraťte jeho obsah v návratové hodnotě.

Vytvořte vlastní funkci pro zápis řetězce do textového souboru. Funkce bude mít dva povinné a jeden volitelný parametr. První bude název souboru, druhý bude řetězec pro zápis a třetí je způsob zápisu, kde výchozí hodnota je `false` a znamená způsob zápisu na konec souboru. Pokud uživatel použije hodnotu `true`, nový řetězec bude zapsán před existující data v souboru, pokud již existuje. V tomto případě použijte předchozí funkci pro čtení k zjištění existence a případného načtení obsahu souboru pro zápis. Čtení a zápis vyzkoušejte.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. `C:\xampp\htdocs`) pod názvem `funkce-soubory.php`. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte funkci čtení s parametrem `nazevSouboru`. Vytvořte podmínku s ověřením případné neexistence souboru přes negaci výsledku funkce `file_exists`. Pokud neexistuje, vraťte hodnotu `false`. Pokud existuje, načtěte délku souboru funkcí `filesize`. Pokud je délka nulová (prázdný soubor), vraťte opět hodnotu `false`. Pokud je délka nenulová, proveďte načtení souboru s parametrem pro čtení, načtěte celý obsah do proměnné `obsah`, zavřete soubor a vraťte hodnotu proměnné `obsah`.

V druhé části vytvořte funkci `zapis` se třemi parametry, první `nazevSouboru`, druhý `text` a třetí `způsob`, výchozí hodnota `false`.

Pokud je hodnota způsobu zápisu `false`, otevřete soubor parametrem `"a"` pro zápis na konec souboru, proveďte zápis proměnné `text` za původní obsah souboru a soubor uzavřete.

Pokud je hodnota způsobu zápisu `true`, načtěte obsah souboru již hotovou metodou čtení. Pokud hodnota nevrátí `false`, slučte vrácený obsah za proměnnou `text`. Pokud vrátí `false`, pokračujte s původním textem. Otevřete soubor pro zápis s parametrem `"w"` pro případné vytvoření či vyprázdnění souboru. Zapište do souboru `text` a soubor uzavřete. Ověřte funkčnost čtení a zápisu externích souborů.

Metoda ověření:

Zobrazení čtení souboru: Po otevření URL v prohlížeči by se měl zobrazit obsah textového souboru `pokus.txt`, pokud existuje.

Kontrola zápisu na konec a začátek souboru: Ověřte, že do souboru `pokus.txt` byla správně zapsána data na konec a na začátek souboru.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a správně provádí všechny operace se souborem.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci. Doporučený čas je 6 vyučovacích hodin.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL

Možný chybný zápis funkce pro práci se soubory, chybný parametr s cestou k souboru, použití chybného parametru.

Problém s diakritikou při práci s knihovními funkcemi pro vytváření substringu, převodu znaků na malé či velké.

5.2.6 Formuláře**Zaměření**

Učivo ŠVP: Formuláře

Časová dotace: 290–385 minut

Cíle

Obsah tohoto metodického listu je zaměřen na seznámení žáků s technikami zpracování dat z HTML formulářů pomocí metod GET a POST, jejich validací a ukládáním do textových souborů v programovacím jazyce PHP. Žáci se naučí efektivně pracovat s uživatelskými vstupy, zpracovávat je a ukládat.

Kompetence

Kognitivní: žák umí pracovat s daty z HTML formulářů a URL adres, zná jejich možné způsoby zpracování.

Praktické: žák používá superglobální proměnné GET a POST pro získání formulářových dat a dat z URL, které dále používá k tvorbě aplikace.

Obsah výuky:

- Metoda předání formuláře metodou GET a POST
- Superglobální proměnné GET a POST
- Zpracování a základní validace dat
- Tvorba funkční aplikace s formuláři a ukládáním dat

Organizační formy a metody výuky

Celý materiál je rozdělen do pěti oblastí, které na sebe navzájem navazují.

1. Metoda GET, parametry v URL a jejich získání v PHP

Tato oblast je zaměřena na práci s metodou GET v PHP, na použití superglobální proměnné `$_GET` pro získání dat z URL, na ověření existence a hodnot parametrů a na výpis těchto dat.

Rozsah: 45–60 minut

1. **Praktická ukázka**, jak metoda GET funguje a jak používat parametry v URL⁹⁶. (10 minut)
2. **Demonstrace**, jak získat data z URL pomocí `$_GET`⁹⁷. (5 minut)
3. **Demonstrace** použití funkce `isset` pro kontrolu existence parametrů. (5 minut)
4. **Praktická ukázka**, jak kontrolovat, zda jsou data prázdná pomocí funkce `empty`⁹⁸. (5 minut)
5. **Aktivita** – praktické cvičení, kde žáci načtou a vypíší data z URL – Úloha 6.1⁹⁹. (20 minut)

2. Metoda POST, získání dat z formuláře

Tato oblast je zaměřena na práci s daty v PHP, předanými HTML formuláři metodou POST. Žáci se naučí, jak metoda POST funguje, jak získat data z formuláře pomocí superglobální proměnné `$_POST`, a jak ověřit, zda byl formulář odeslán a zda existují potřebné parametry. V závěru si prakticky vyzkouší načítání a výpis získaných dat z formuláře.

Rozsah: 45–60 minut

1. **Praktická ukázka**, jak metoda POST funguje a jak používat `$_POST`¹⁰⁰. (10 minut)
2. **Demonstrace**, jak zkontrolovat, zda byl formulář odeslán¹⁰¹. (5 minut)
3. **Demonstrace**, jak ověřit existenci odeslaných dat. (5 minut)

⁹⁶ https://www.w3schools.com/tags/ref_httpmethods.asp

⁹⁷ https://www.w3schools.com/php/php_superglobals_get.asp

⁹⁸ https://www.w3schools.com/php/func_var_empty.asp

⁹⁹ Soubor `get.php` umístěný ve složce `list6` v příloze

¹⁰⁰ https://www.w3schools.com/php/php_superglobals_post.asp

¹⁰¹ https://tryphp.w3schools.com/showphp.php?filename=demo_global_post

4. **Demonstrace**, jak načítat data z formuláře. (5 minut)
5. **Aktivita** – praktické cvičení, kde žáci načtou a vypíší data z formuláře – Úloha 6.2¹⁰². (20 minut)

3. Validace, zpracování a výpis dat

Tato oblast je zaměřena na validaci a zpracování formulářových dat v PHP. Žáci se naučí, jak používat funkce trim a htmlspecialchars pro zpracování a zabezpečení vstupních dat. Také se naučí, jak zpracovávat různé druhy formulářových polí a jak validovat a vypisovat jejich hodnoty.

Rozsah 55–70 minut

1. **Demonstrace**, jak použít funkci trim k odebrání bílých znaků¹⁰³. (5 minut)
2. **Demonstrace**, jak použít funkci htmlspecialchars k ochraně proti XSS útokům¹⁰⁴. (5 minut)
3. **Otevřená diskuse** nad XSS útoky. (10 minut)
4. **Demonstrace**, jak zpracovávat různé typy formulářových polí (text, číslo, email)¹⁰⁵. (10 minut)
5. **Aktivita** – praktické cvičení, kde žáci validují a vypisují jednotlivé hodnoty – Úloha 6.3¹⁰⁶. (15 minut)

4. Praktické cvičení – tvorba návštěvní knihy pomocí textového souboru

Tato oblast je zaměřena na tvorbu HTML formuláře, získání dat z formuláře pomocí metody POST, zjištění aktuálního datumu a času, zápis získaných dat na začátek textového souboru a výpis obsahu textového souboru. Celá oblast je brána formou praktického cvičení (Úloha 6.4¹⁰⁷) na tvorbu návštěvní knihy, doplněna o vysvětlení jednotlivých částí a ukázky ze strany učitele.

Rozsah: 60–75 minut

¹⁰² Soubor post.php umístěný ve složce list6 v příloze

¹⁰³ https://www.w3schools.com/php/func_string_trim.asp

¹⁰⁴ https://www.w3schools.com/php/func_string_htmlspecialchars.asp

¹⁰⁵ https://tryphp.w3schools.com/showphp.php?filename=demo_form_validation_escapechar

¹⁰⁶ Soubor validate.php umístěný ve složce list6 v příloze

¹⁰⁷ Soubor kniha.php umístěný ve složce list6 v příloze

1. **Aktivita 1** - žáci vytvoří HTML formulář s poli pro jméno, nadpis a text. (10 minut)
2. **Demonstrace**, jak získat data z formuláře předaného pomocí metody POST. (5 minut)
3. **Demonstrace**, jak získat aktuální datum a čas pomocí PHP funkce date¹⁰⁸. (10 minut)
4. **Aktivita 2** - žáci zapíší získaná data na začátek textového souboru pomocí vlastní funkce pro zápis řetězce. (15 minut)
5. **Praktická ukázka**, jak načíst a vypsat obsah textového souboru pomocí vlastní metody pro načtení obsahu. (5 minut)
6. **Aktivita 3** – žáci vytvoří výpis článků z textového souboru pomocí vlastní metody. (15 minut)

5. Praktické cvičení – registrace uživatelů do CSV souboru

Tato oblast je zaměřena na tvorbu HTML formuláře, získání dat z formuláře pomocí metody POST, zápis získaných dat na konec souboru typu CSV. Celá oblast je brána formou praktického cvičení (Úloha 6.5) na tvorbu registračního formuláře a výpisu registrovaných uživatelů, doplněna o vysvětlení jednotlivých částí a ukázky ze strany učitele.

Rozsah: 85–120 minut

1. **Prezentace** na téma CSV soubory¹⁰⁹. (15 minut)
2. **Praktická ukázka** tvorby CSV souboru v textovém editoru. (10 minut)
3. **Aktivita 1** - žáci vytvoří HTML formulář s poli pro jméno, příjmení a e-mail. (10 minut)
4. **Demonstrace**, jak získat data z formuláře předaného pomocí metody POST. (5 minut)
5. **Aktivita 2** - žáci zapíší získaná data na konec CSV souboru pomocí vlastní funkce pro zápis řetězce. (15 minut)
6. **Demonstrace** funkce fgets¹¹⁰ a feof¹¹¹ pro čtení souboru po řádcích. (10 minut)

¹⁰⁸ https://www.w3schools.com/php/func_date_date.asp

¹⁰⁹ <https://cs.wikipedia.org/wiki/CSV>

¹¹⁰ https://www.w3schools.com/php/func_filesystem_fgets.asp

¹¹¹ https://www.w3schools.com/php/func_filesystem_feof.asp

7. **Praktická ukázka**, jak načíst, rozdělit a vypsát obsah CSV souboru pomocí vlastní metody pro načtení obsahu. (5 minut)
8. **Aktivita 3** – žáci vytvoří výpis registrovaných uživatelů ze souboru CSV. (15 minut)

Úlohy:

Úloha 6.1 – načtení a výpis textového souboru

Tato úloha pomůže žákům procvičit si práci s metodou GET v PHP, naučí je, jak získat data z URL, ověřit existenci parametrů pomocí funkcí `isset` a `empty`, a vypsát načtená data na obrazovku.

Zadání úlohy 6.1:

Vytvořte skript pro načtení parametrů `stranka` a `id` z URL. Ověřte existenci parametrů a zda jsou naplněny. Pokud ano, vypište jejich obsah. Pokud ne, informujte o tomto stavu, například parametr `id` existuje, ale je prázdný, nebo parametr `id` neexistuje. Volání provádějte například ve tvaru `get.php?stranka=aktuality&id=1`

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte podmínku, testující existenci parametru `stranka` v URL, v případě neexistence proveďte výpis s touto informací. V případě existence vytvořte další podmínku na ověření, zda je parametr prázdný. Pokud je prázdný, proveďte výpis s touto informací. Pokud není prázdný, proveďte výpis s informací o obsahu parametru `stranka`.

Stejný postup opakujte pro parametr `id` z URL.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. `C:\xampp\htdocs`) pod názvem `get.php`. Tento soubor otevřete v URL prohlížeče, poté s patřičnými parametry bez obsahu či naplněné daty.

Metoda ověření:

Zobrazení dat z URL: Po otevření URL v prohlížeči by se měly zobrazit data předaná v URL (stranka a id).

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a správně načítá a vypisuje data z URL.

Úloha 6.2 – výpis dat z HTML formuláře metodou POST

Tato úloha pomůže žákům procvičit si práci s metodou POST pro předání HTML formuláře a superglobální proměnnou `$_POST` v PHP. Žáci se naučí, jak ověřit odeslání formuláře, zkontrolovat existenci parametrů, načíst data z formuláře a vypsát je na obrazovku.

Zadání úlohy 6.2:

Vytvořte html formulář s textovými vstupními prvky jmeno a nadpis a s prvkem textarea jmenující se text. Formulář předejte metodou POST. Pod tento formulář umístěte PHP skript, který ověří odeslání formuláře, dále ověří existenci všech polí z formuláře a pokud existují, vypíše jejich obsah.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Vytvořte na začátek těla HTML stránky formulář s metodou předání dat POST. Do formuláře umístěte tři prvky a odesílací tlačítko. Prvním prvkem bude vstupní prvek typu text, jmenující se jmeno. Vytvořte pro něj label Jméno. Druhým prvkem bude vstupní prvek typu text jmenující se nadpis a vytvořte pro něj label Nadpis. Třetím prvkem bude textarea se jménem text, nastavte ji například velikost 80 znaků a 6 řádek. Doplňte label Text. Pod tyto prvky dále umístěte tlačítko pro odeslání formuláře.

Pod formulářem vytvořte PHP skript, ve kterém pomocí podmínky ověřte přijetí odeslaného formuláře. V případě splnění podmínky otestujte existenci všech polí pomocí funkce `isset` v jedné podmínce. V případě, že data existují, proveďte výpis jejich obsahu ve tvaru Jméno: + získaná data z tohoto prvku formuláře, Nadpis: + získaná data z tohoto prvku formuláře a na závěr Text: + získaná data z tohoto prvku formuláře s použitím funkce `nl2br` pro případné řádkování.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem post.php. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Ověření odeslání formuláře: Zkontrolujte, zda se zobrazují data po odeslání formuláře, v případě neodeslání musí stránka obsahovat jen formulář.

Zobrazení dat: Po správném odeslání formuláře zkontrolujte, zda se data odeslaná formulářem správně zobrazují.

Úloha 6.3 – validace a úprava dat z formuláře

Tato úloha navazuje na předchozí úlohu 6.2 a rozšiřuje jí o validaci dat. Žáci se naučí, jak načítat data z formuláře, ověřovat je a bezpečně je zobrazovat na obrazovce.

Zadání úlohy 6.3:

Nejprve spusťte znovu soubor post.php z úlohy 2 a například do pole jméno umístěte i HTML tagy, příklad `<strong>Pokus</strong>`, vyplňte další pole a formulář odešlete. Všimněte si, že výpis bude zároveň vykonávat tag `<strong>`. Takto by mohl uživatel podstrčit i případný kód v jazyce JavaScript a další.

Použijte formulář, ověření odeslání formuláře a existence hodnot z předchozího příkladu v úloze 6.2. Použijte funkci trim pro odstranění bílých znaků z jednotlivých hodnot a poté pomocí funkce htmlspecialchars ošetřete případné značky z těchto polí.

Výpis dat proveďte obdobně jako v úloze 6.2 a pokuste se o podobný příklad umístění HTML tagů do polí.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem validate.php. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Otevřete předchozí soubor z úlohy 2 a jeho obsah si zkopírujte.

Uvnitř vnořených podmínek načtěte hodnoty ze superglobální proměnné POST. Pomocí funkce trim odstraňte z těchto prvků bílé znaky. Dále pomocí funkce htmlspecialchars

odstraňte případné tagy z těchto polí. V závěru vytvořte obdobný výpis jako v úloze 2, ovšem již takto ošetřených dat.

Metoda ověření:

Ověření odeslání formuláře: Zkontrolujte, zda se zobrazují data po odeslání formuláře, v případě neodeslání musí stránka obsahovat jen formulář.

Zobrazení dat: Po správném odeslání formuláře zkontrolujte, zda se data odeslaná formulářem správně validují a zobrazují.

Úloha 6.4 – návštěvní kniha s externím souborem

Tato úloha pomůže žákům procvičit si práci s HTML formuláři, metodou POST, získáním aktuálního datumu a času, zápisem dat do textového souboru a jejich následným výpisem. V této úloze budou využity metody pro zápis a čtení souboru z metodického listu Textové soubory, řetězce a využít formulář a skript z předchozí úlohy 6.3.

Zadání úlohy 6.4:

Na základě formuláře z úlohy 6.3 a jeho zpracování vytvořte návštěvní knihu, která bude do textových souborů zapisovat jednotlivé vzkazy s nejnovějším vzkazem na začátku. Pro zápis použijte vlastní funkci zapis z metodického listu Textové soubory, řetězce. Zároveň použijte i funkci cteni, kterou funkce zapis využívá pro zápis na začátek souboru.

Vytvořte řetězec obsahující nadpis úrovně 2 s nadpisem vzkazu, dále do odstavce napište jméno autora vzkazu a pomocí funkce date informujte o datu a času přidání vzkazu. Řetězec doplňte o odstavec textu vzkazu s případným zalomením řádek pomocí funkce nl2br.

Řetězec zapište funkcí zapis do souboru kniha.txt.

Pod formulář proved'te výpis obsahu souboru kniha, získaný funkcí cteni.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem kniha.php. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Na základě úlohy 6.3 použijte formulář a skript pro zpracování dat.

Použijte funkci zapis a cteni ze souboru funkce-soubory.php z metodického listu Textové soubory, řetězce.

Vytvořte HTML kód pro jeden vzkaz ve formátu nadpis úrovně 2 obsahující nadpis z formuláře.

Dále vytvořte odstavec textu obsahující text Autor: a doplňte jméno autora z formuláře. Za jméno doplňte text datum přidání: a pomocí funkce date vypište den.měsíc.rok dle aktuálního času na serveru. Tento údaj doplňte i o hodinu:minutu:sekundu a odstavec ukončete.

Vytvořte další odstavec obsahující text z formuláře, doplněný o řádkování pomocí funkce nl2br.

Tento HTML kód uložte do řetězce vzkaz.

Zapište řetězec vzkaz do souboru kniha.txt funkcí zápis s parametrem zápisu na začátek souboru, tedy true.

Pod skript, který zpracovává data proveďte výpis celého obsahu souboru kniha.txt funkcí cteni.

Metoda ověření:

Ověření odeslání formuláře: Zkontrolujte, zda data z formuláře jsou odeslána, zpracována a zobrazena na stránce, společně s předchozími záznamy v návštěvní knize.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a vypisuje obsah souboru kniha.txt.

Úloha 6.5 – registrace uživatelů do CSV souboru

Tato úloha seznámí žáky s CSV soubory, pomůže žákům procvičit si práci s HTML formuláři, metodou POST, zápisem dat do CSV souboru a jejich následným výpisem. V této úloze budou využity metody pro zápis a čtení souboru z metodického listu Textové soubory, řetězce a využít formulář z předchozích úloh.

Zadání úlohy 6.5:

Na základě formuláře z úlohy 6.3 a jeho zpracování vytvořte registraci uživatelů, která bude do CSV souboru zapisovat jednotlivé uživatele na nové řádky, hodnoty budou uzavřeny v uvozovkách a odděleny čárkou.

Vstupní prvky jsou jméno, příjmení a e-mail.

Pro zápis použijte vlastní funkci zapis z metodického listu Textové soubory, řetězce. Zároveň použijte i funkci cteni, kterou funkce zapis využívá ke své funkčnosti.

Vytvořte řetězec z registračních údajů a zapište jej funkcí zapis do souboru registrace.csv na konec souboru.

Pod formulář proved'te výpis obsahu CSV souboru. Soubor čtete po řádcích metodou fgets, rozdělte dle čárek a vyjměte data z uvozovek.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem registrace.php. Tento soubor otevřete v URL prohlížeče.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Na základě úlohy 6.3 použijte formulář, který modifikujte pro vstupní prvky jméno, příjmení a e-mail, a skript pro zpracování dat.

Použijte funkci zapis a cteni ze souboru funkce-soubory.php z metodického listu Textové soubory, řetězce.

Vytvořte řetězec uzivatel ze vstupních hodnot. Hodnoty umístěte do uvozovek a oddělte je čárkou. Příklad: "jméno","příjmení","e-mail"

Zapište řetězec uzivatel do souboru registrace.csv funkcí zapis s parametrem zápisu na konec souboru, tedy bez parametru či parametr false.

Pod skript, který zpracovává data proved'te výpis registrovaných uživatelů.

Otevřete soubor funkcí fopen ke čtení, pomocí while postupně načítejte řádky funkcí fgets.

Ukončující podmínku v cyklu použijte jako negaci návratové hodnoty feof z tohoto souboru, tedy while(!feof(\$soubor)).

Pomocí funkce `explode`, `mb_substr` a dalších oddělte jednotlivá data.

Hodnoty vypíšte na nové řádky ve tvaru „jméno příjmení e-mail“.

Metoda ověření:

Ověření odeslání formuláře: Zkontrolujte, zda data z formuláře jsou odeslána, zpracována a zobrazena na stránce, společně s předchozími záznamy v registraci.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a vypisuje obsah dat ze souboru `registrace.csv`.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci. Doporučený čas je 9 vyučovacích hodin.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL

Možný chybný zápis superglobálních proměnných.

Chybný způsob odeslání formuláře.

Problém s dělením řádky CSV souboru.

Několikanásobné odeslání formuláře při obnovení stránky.

5.2.7 Hlavičky

Zaměření

Učivo ŠVP: Hlavičky a session

Časová dotace: 35–50 minut

Cíle

Tento metodický list je zaměřen na seznámení žáků s technikami práce s HTTP hlavičkami v PHP. Žáci se naučí, jak efektivně využívat HTTP hlavičky pro přesměrování, změnu stavových kódů, ovlivňování ukládání stránek do mezipaměti a nastavení typu souboru.

Kompetence

Kognitivní: žák umí modifikovat hlavičky webu, zná základní význam vybraných hlaviček a jejich použití v praxi.

Praktické: žák používá hlavičky k přesměrování na jiné adresy, mění typy souborů v hlavičkách a ovládá změny ukládání do mezipaměti prohlížeče (cache).

Obsah výuky:

- Funkce header a její zápis
- Nastavení stavového kódu
- Přesměrování na jinou adresu
- Nastavení typu souboru aktuální odpovědi
- Nastavení ukládání do mezipaměti prohlížeče

Organizační formy a metody výuky

Celý materiál je rozdělen do jedné oblasti, vysvětlující a demonstrující danou problematiku.

1. Hlavičky, jejich zápis a možné použití v praxi

Tato oblast je zaměřena na práci s HTTP hlavičkami v PHP. Žáci se naučí, co jsou HTTP hlavičky, jak fungují, a jak je používat k různým účelům, jako je přesměrování uživatelů, změna stavových kódů, nastavení typu souboru a zakázání ukládání do mezipaměti prohlížeče. Závěrem je praktické použití přesměrování v jednom z předchozích skriptů.

Rozsah 35–50 minut

1. **Prezentace** na téma HTTP hlavičky a jak fungují¹¹². (5 minut)
2. **Demonstrace**, jak vytvořit a použít hlavičku pro přesměrování uživatele. (5 minut)
3. **Demonstrace**, jak změnit stavový kód pomocí HTTP hlaviček. (5 minut)
4. **Praktická ukázka**, jak nastavit typ souboru pomocí hlavičky. (5 minut)
5. **Demonstrace**, jak použít hlavičku pro zakázání ukládání do mezipaměti. (5 minut)
6. **Aktivita** – praktické cvičení, kde žáci přesměrují stránku po úspěšném zpracování formuláře – Úloha 7.1¹¹³ (10 minut)

¹¹² <https://www.php.net/manual/en/function.header.php>

¹¹³ Soubor kniha-presmerovani.php umístěný ve složce list7 v příloze

Úlohy:

Úloha 7.1 – přesměrování uživatele po odeslání vzkazu

Tato úloha pomůže žákům vylepšit možnosti zpracování dat z formulářů o přesměrování na jinou stránku pomocí hlaviček či obnovení stránky. Tím lze ošetřit nechtěné znovu odeslání formuláře při stisku klávesy F5 či obnovení stránky uživatelem.

Zadání úlohy 7.1:

Na základě skriptu kniha.php z metodického listu Formuláře – úloha 6.4 vytvořte nový soubor, který bude obsahovat celý předchozí příklad tvorby návštěvní knihy s použitím textového souboru. Skript doplňte o přesměrování na sebe samotný funkcí header, čímž dojde k obnovení dat a zároveň k ochraně před znovu odesláním dat stiskem klávesy F5.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Otevřete soubor kniha.txt z úlohy 6.4 metodického listu Formuláře.

Překopírujte její obsah.

Pod zápis do souboru funkcí zapis proved'te přesměrování funkcí header s názvem aktuálního souboru, tedy kniha-presmerovani.php.

Po odeslání příspěvku zkuste stisknout klávesu F5 pro obnovení stránky. Prohlížeč by již neměl informovat o případném znovu odeslání dat.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. C:\xampp\htdocs) pod názvem kniha-presmerovani.php. Tento soubor otevřete v URL prohlížeče.

Metoda ověření:

Ověření odeslání formuláře: Zkontrolujte, zda data z formuláře jsou odeslána, zpracována a zobrazena na stránce, společně s předchozími záznamy v návštěvní knize a zda došlo k obnovení stránky stiskem klávesy F5, kdy neproběhne upozornění o znovu odeslání dat.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a vypisuje obsah souboru kniha.txt.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci.
Doporučený čas je 1 vyučovací hodina.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL.

Chybná definice hlaviček.

Chybné zadání názvu souboru či adresy pro přesměrování.

5.2.8 Relace – session**Zaměření**

Učivo ŠVP: Hlavičky a session

Časová dotace: 80–110 minut

Cíle

Tento metodický list zahrnuje práci s relacemi a superglobální proměnnou SESSION, ukládání dat do relací, jejich modifikaci a mazání. Cílem je seznámit žáky s možností vytváření relací mezi serverem a uživatelem.

Kompetence

Kognitivní: žák umí vytvářet, modifikovat a mazat relace mezi serverem a uživatelem.

Praktické: žák používá superglobální proměnné SESSION pro vytváření relace mezi serverem a uživatelem.

Obsah výuky:

- Relace a superglobální proměnná SESSION
- Vytváření relací
- Modifikace relací
- Odebrání hodnot relace
- Mazání celé relace

Organizační formy a metody výuky

Celý materiál je rozdělen do dvou oblastí, které na sebe navzájem navazují.

1. Relace a superglobální proměnná SESSION

Tato oblast je zaměřena na práci s relacemi (sessions) v PHP. Žáci se naučí, co jsou relace, jak fungují, kde se ukládají data, jak používat superglobální proměnnou `$_SESSION`, ukládat a číst hodnoty v relaci, a jak relace mazat a modifikovat.

Rozsah 35–50 minut

1. **Prezentace** konceptu relací, jejich funkce a umístění dat¹¹⁴. (10 minut)
2. **Demonstrace**, jak používat superglobální proměnnou `$_SESSION`¹¹⁵. (5 minut)
3. **Praktické ukázky**, jak ukládat data do relace pomocí `$_SESSION`¹¹⁶. (5 minut)
4. **Demonstrace**, jak číst a modifikovat data uložená v relaci¹¹⁷. (5 minut)
5. **Demonstrace**, jak odstranit konkrétní hodnotu z relace¹¹⁸. (5 minut)
6. **Demonstrace**, jak smazat celou relaci pomocí PHP funkcí¹¹⁹. (5 minut)

2. Praktický příklad vytváření relace, ukládání dat, modifikace, mazání a rušení

Tato oblast je zaměřena na praktickou práci s relacemi (sessions) v PHP, včetně vytvoření jednoduchého systému fiktivního přihlášení uživatele, uložení a zpracování uživatelských dat do relace a odhlášení uživatele. Žáci se naučí, jak používat relace k ukládání a předávání dat. Celý okruh je veden jako tvorba Úlohy 8.1¹²⁰ s postupným vývojem na základě ukázky klíčových částí ze strany učitele.

Rozsah: 45–60 minut

1. **Aktivita 1** - žáci vytvoří jednoduchý formulář pro zadání uživatelského jména (login) a hesla. (10 minut)
2. **Praktická ukázka**, jak vytvořit relaci a uložit data z formuláře do `$_SESSION`. (5 minut)

¹¹⁴ [https://cs.wikipedia.org/wiki/Relace_\(informatika\)](https://cs.wikipedia.org/wiki/Relace_(informatika))

¹¹⁵ <https://www.php.net/manual/en/reserved.variables.session.php>

¹¹⁶ https://tryphp.w3schools.com/showphp.php?filename=demo_session1

¹¹⁷ https://tryphp.w3schools.com/showphp.php?filename=demo_session2

¹¹⁸ <https://www.php.net/manual/en/function.session-unset.php>

¹¹⁹ <https://www.php.net/manual/en/function.session-destroy.php>

¹²⁰ Soubor `relace.php` a `relace2.php` umístěné ve složce `list8` v příloze

3. **Aktivita 2** – žáci vytvoří skript pro uložení a ověření existence relace s uživatelskými daty. (15 minut)
4. **Praktická ukázka**, jak vytvořit metodu pro odhlášení uživatele a odebrání relace. (5 minut)
5. **Aktivita 3** – praktické cvičení, kde žáci předají uložené hodnoty mezi více soubory a provedou odhlášení uživatele. (10 minut)

Úlohy:

Úloha 8.1 – relace a přihlášení

Tato úloha pomůže žákům procvičit si práci s HTML formuláři, metodou POST, relacemi (sessions) a přesměrováním uživatelů v PHP. Žáci se naučí, jak vytvořit jednoduchý systém přihlášení, ukládat uživatelské údaje do session, a jak implementovat odhlášení uživatele s přesměrováním.

Zadání úlohy 8.1:

Vytvořte skript a inicializujte práci s relacemi. Ověřte, že není přihlášen uživatel. Pokud není, zobrazte formulář pro zadání loginu a hesla a odešlete jej metodou post. V případě přijetí dat je zpracujte dle předchozích příkladů a login uložte do superglobální proměnné `$_SESSION` na hodnotu klíče login. Obnovte adresu funkcí header. Pokud proběhl zápis, podmínka na ověření přihlášení bude provedena. V tomto případě místo formuláře zobrazte informaci se jménem přihlášeného uživatele ve tvaru Login: + login ze SESSION. Pod text umístěte odkaz na tento soubor s GET parametrem odhlasit, který provede odhlášení uživatele.

Doplňte skript o podmínku, testující existenci tohoto GET parametru v URL. Pokud existuje parametr odhlasit, odeberte pouze hodnotu session s loginem, neprovádějte `session_destroy`.

Po odebrání přesměrujte uživatele zpět na stránku bez GET parametru v URL, kde bude opět zobrazen formulář pro přihlášení.

Vytvořte kopii tohoto souboru a pojmenujte jej `relace2.php` a po úspěšném přihlášení v původním souboru vykonajte i tento soubor v prohlížeči. Výsledkem bude aktivní přihlášení i v tomto skriptu.

Pracovní postup:

Otevřete textový editor pro práci s PHP a vytvořte nový soubor typu PHP.

Inicializujte práci s relacemi funkcí `session_start`. Ověřte, že není přihlášen uživatel přes podmínku a negaci funkce `isset`.

V případě splnění podmínky vytvořte HTML formulář s textovým vstupním prvkem login, vstupním prvkem pro heslo se jménem heslo a tlačítkem pro odeslání.

Nad tento formulář vytvořte zpracující příchozí data z formuláře metodou POST.

Do superglobální proměnné `$_SESSION["login"]` uložte hodnotu získanou z formulářového prvku login.

Pomocí funkce `header` přesměrujte uživatele na tento skript, čímž dojde k obnovení skriptu a tím nesplnění podmínky neexistence této relace.

Vytvořte větev `else`, která vypíše informace o přihlášeném uživateli ve tvaru Login: + login ze SESSION. Pod text umístěte odkaz na tento soubor s GET parametrem `odhlasit`, který provede odhlášení uživatele.

Nad celým skriptem vytvořte podmínku, testující existenci tohoto GET parametru v URL. Pokud existuje parametr `odhlasit`, odeberte pouze hodnotu session s loginem funkcí `unset`.

Po odebrání přesměrujte uživatele zpět na stránku bez GET parametru v URL funkcí `header`.

Uložte soubor do kořenového adresáře vašeho webového serveru (např. `C:\xampp\htdocs`) pod názvem `relace.php`. Tento soubor otevřete v URL prohlížeče.

Vytvořte kopii tohoto souboru a pojmenujte jej `relace2.php` a po úspěšném přihlášení v původním souboru vykonajte i tento soubor v prohlížeči pro ověření funkčnosti relace mezi serverem a uživatelem.

Metoda ověření:

Ověření odeslání formuláře: Zkontrolujte, zda data z formuláře jsou odeslána, zpracována a uložena do relace.

Bezchybnost skriptu: Ujistěte se, že skript neobsahuje žádné chyby a vypisuje obsah SESSION po úspěšném odeslání přihlášení.

Vhodný postup:

Celý metodický list je navržen lineárně a je potřeba zajistit dostatek času pro jeho realizaci.
Doporučený čas je 2 vyučovací hodiny.

Očekávané problémy:

Nesprávná syntaxe PHP kódu.

Chybné umístění souborů do složky na serveru či jeho chybné volání přes URL.

Zapomenuté zahájení práce s relací funkcí `session_start`.

Chybná definice session.

Chybný parametr v URL.

6 Ověření metodiky

Cílem ověření navržené metodiky je zjistit, zda je výuka dle jednotlivých listů pro žáky srozumitelná, pochopitelná, úlohy v nich obsažené odpovídají doposud získaným znalostem žáků a lze ji aplikovat při výuce skriptovacího jazyka PHP při dodržení doporučených hodinových dotací.

Metodika byla nasazena a ověřena na Střední průmyslové škole Ostrov, p.o. Nasazení proběhlo ve školním roce 2022/2023 u třetího ročníku oboru 18-20-M/01 Informační technologie v tematických celcích tomu určených dle platného ŠVP. (Střední průmyslová škola Ostrov, příspěvková organizace, 2018)

Pro výuku předmětu Programování a vývoj aplikací je třída dělená na dvě skupiny. Ověření tedy probíhalo paralelně se skupinou „PVAa“ i „PVAb“.

Třída I3 čítala 25 žáků a byla rozdělena do skupin po 11 a 14 žácích.

První hodina dle této metodiky proběhla u obou skupin třídy v týdnu od 5.12.2022.

Výuka probíhala u obou skupin paralelně dle navržených scénářů a lišila se převážně jen v obsahu diskusí a podnětů ze strany žáků.

6.1 Očekávané výsledky ověření

Před začátkem ověřování této metodiky měli žáci již ukončené tematické celky, věnované tvorbě statických stránek v jazyce HTML spolu s CSS, znali základní práci se skriptovacím jazykem JavaScript a knihovnami jQuery a také používání knihovny Bootstrap 4 pro tvorbu front-endu stránek.

Z výše vyjmenovaných technologií žáci před tímto tematickým celkem zaměřeným na PHP vytvářeli ročníkovou práci v podobě kompletní statické webové prezentace, tudíž problémy na straně tvorby HTML se neočekávaly a ani skutečně nenastaly.

Paralelně s výukou předmětu PVA probíhá i výuka PGV, ve kterém žáci aktivně vytvářejí desktopové aplikace v jazyce C#. Na základě této zkušenosti se nepředpokládá, že by základní konstrukce v jazyce PHP přinášely problémy na straně nepochopení těchto

konstrukcí. Očekává se naopak rychlé pochopení probírané látky a tím méně kolizní průběh výuky dle metodiky.

Některé předpokládané komplikace při výuce:

Očekávané problémy mohou nastat v případě odlišné syntaxe jazyka PHP oproti C#, kde žáci mohou chybně zapisovat konstrukty, či názvy funkcí tak, jak je zvykem u druhého jazyka, se kterým pracují od druhého ročníku.

Dle očekávaných problémů v metodických listech lze v průběhu výuky očekávat chyby v nesprávné syntaxi PHP kódu, chybné umístění souborů do složky na serveru, jeho chybné volání přes URL či nespustění lokálního serveru.

U metodických listů z kapitoly 5.2.1 *Prostředí PHP, základní příkazy, proměnné, datové typy*, 5.2.2 *Příkazy větvení a cykly*, 5.2.3 *Funkce a globální proměnné* a 5.2.4 *Pole* se nepředpokládá neovládání tématu či problémy s tvorbou úloh kromě specifických možných chyb při tvorbě či interpretaci skriptu.

Metodický list 5.2.5 *Textové soubory, řetězce* a úlohy 5.1–5.5 v něm obsažené mohou způsobit vznik chyb převážně na straně chybné syntaxe funkcí pro práci se soubory, chybný parametr s cestou k souboru, použití chybného parametru a také problém s diakritikou a kódováním znaků v textových souborech či řetězcích.

Asi nejsložitějším se může jevit obsah metodického listu 5.2.6 *Formuláře* zaměřený na práci s uživatelskými daty a úloha 6.4 na samostatnou tvorbu aplikace v podobě návštěvní knihy a úloha 6.5 na tvorbu registrace do CSV souborů. Očekávají se problémy s předáváním dat formuláři, problémy se zpracováním superglobálních proměnných či převodu obsahu CSV souboru na data.

Metodické listy 5.2.7 *Hlavičky* a 5.2.8 *Relace - session* a úlohy 7.1 a 8.1 v nich obsažené nepředpokládají neovládání tématu či problémy s tvorbou úloh kromě případných drobných a snadno opravitelných chyb.

Všechny úlohy je potřeba vykonávat postupně a rozdělit na několik dílčích cílů.

Cílem výuky je vést žáky k samostatnosti a pokusit se žáky naučit především samostatně pomocí dostupných zdrojů řešit vzniklé potíže.

6.2 Průběh výuky pomocí navržené metodiky

Výuka probíhala dle platného rozvrhu u obou skupin v počítačové učebně s maximální kapacitou 16 míst pro žáky a katedrou, vybavené osobními počítači na každém stanovišti, projektorem a bílou tabulí pro případné poznámky, schémata a diagramy.

Výuka probíhala od prosince 2022 do března 2023, převážně formou dvouhodinového bloku 2x45 minut. Tento celek zahrnoval 36 vyučovacích hodin paralelně u obou skupin. Skutečné ověření této metodiky trvalo 72 vyučovacích hodin.

Na začátku každého vyučovacího bloku nebo při zahájení výuky nového tématu či oblasti probíhalo krátké představení tématu, vyslovení cílů a krátká aktivace již získaných znalostí z předchozích hodin. Před koncem vyučování byl vždy krátký prostor pro shrnutí získaných znalostí a dovedností a prostor pro diskusi.

6.2.1 Prostředí PHP, základní příkazy, proměnné, datové typy

Tento metodický list byl rozložen do 6 vyučovacích hodin u každé skupiny.

Během prvních dvou hodin probíhalo teoretické seznámení s jazykem PHP na základě oblastí 1 a 2 v metodickém listu. Součástí byla samostatná tvorba úlohy 1.1, která byla dle žáků snadná. Očekávané problémy nenastaly.

V následujícím dvouhodinovém bloku byla probrána oblast 3 a 4 a zadána úloha 1.2. U některých žáků byla zobrazena chyba špatného názvu funkce či proměnné, ovšem tuto chybu samostatně opravili a úloha byla splněna.

Poslední dvouhodinový blok byl věnován náplni oblastem 5 a 6 a úloze 1.3 na tvorbu programu pro výpočet obvodu kružnice. Úloha byla splněna na základě demonstrace práce s funkcí printf, kterou implementovali pro zaokrouhlení výsledku. Úloha byla ověřena a splněna.

Cíle v tomto metodickém listu byly naplněny.

Doporučená doba 6 vyučovacích hodin je dostatečná, zejména pokud je možno vyučovat formou dvouhodinových bloků jako v tomto případě.

6.2.2 Příkazy větvení a cykly

Tento metodický list byl rozložen do 3 vyučovacích hodin.

Oblast 1 byla spolu s úlohou 2.1 probrána během první hodiny. Vzhledem k častému používání podmínek v rámci jiných programovacích jazyků nebyl ze strany žáků žádný problém s pochopením celé kapitoly a vypracováním úlohy. Na tuto hodinu plynule navázala hodina věnovaná oblasti 2 jejíž součástí byla úloha 2.2 na převedení podmínky z předchozí hodiny do podoby switche. Ani zde nenastal problém s jejím vypracováním. Za přínosné dle diskuse považovali žáci spojení těchto dvou okruhů do bloku, který plynule navázal.

Třetí hodina byla věnována oblasti 3 a úloze 2.3. Žáci měli menší problém s pochopením zadání a někteří navrhovali použít jiný druh cyklu. Ovšem bylo nutné držet se zadání, a tak tuto úlohu dokončili všichni bez problému. Na téma použití cyklu for jsme poté vedli krátkou diskusi.

Ověření všech úloh proběhlo v pořádku a cíle v tomto metodickém listu byly naplněny.

Navrhované nasazení do 3 vyučovacích hodin je v tomto případě dostačující. V případě menší hodinové dotace by jistě šlo oblast 1 a 2 spojit v jeden celek, neboť část teorie je obdobná a úlohy by šlo vypracovat najednou na konci tohoto bloku.

6.2.3 Funkce a globální proměnné

Tento metodický list byl rozložen do prvních 5 vyučovacích hodin během výuky tématu dle ŠVP Funkce, globální proměnné a pole.

První oblast s teorií funkcí byla probrána v druhé vyučovací hodině po úloze 2.1.

Oblast 2 a 3 byla nasazena v dalším bloku. Součástí bylo zadání úlohy 3.1 na tvorbu funkce se čtením globální proměnné, která se obešla bez problémů a poté ihned byla žákům postupně zadána úloha 3.2, 3.3, 3.4 a 3.5 na tvorbu krátkých funkcí. Na tyto aktivity byla vymezená doba 45 minut, ovšem žáci byli hotovi před tímto limitem. Někteří měli hotové všechny úlohy i do 20 minut.

Poslední oblasti 4 a 5 byly nasazeny opět v dvouhodinovém bloku. Součástí první hodiny byla práce na úloze 3.6 pro rekursivní výpočet faktoriálu. Tato úloha byla před tvorbou žáků

prakticky ukázána, a tak nenastal problém s její tvorbou ze strany žáků až na drobné syntaktické chyby, které žáci na základě chybových hlášení opravili. V druhé hodině po prezentaci Fibonacciho posloupnosti pracovali na zadání úlohy 3.7. Zadání bylo splněno a ověřeno a poté žáci po diskusi zkoušeli výpočet vyšších členů posloupnosti.

Ověření všech úloh proběhlo v pořádku a cíle v tomto metodickém listu byly naplněny.

Celý metodický list byl ověřen během 5 vyučovacích hodin, která je adekvátní. Naopak pokud by byla možnost vymezit více hodin, tak by byla možnost problematiky složitosti Fibonacciho posloupnosti rekurzí lépe žáky prozkoumat a také navrhnout a realizovat řešení výpočtu jiným způsobem.

6.2.4 Pole

Tento metodický list byl rozložen do závěrečných 4 vyučovacích hodin během výuky téma dle ŠVP Funkce, globální proměnné a pole.

První blok výuky by věnován nasazení oblasti 1 a 2. Ve druhé polovině hodiny byla žákům zadána úloha 4.1 na tvorbu polí. Po úspěšném ověření úlohy byla ihned zadána úloha 4.2 ve které žáci volně pokračovali v úloze 4.1 a doplnili skript o další dílčí body dle zadání. Časová dotace pro tyto oblasti byla adekvátní. Se žáky bylo možné diskutovat na téma asociativní pole, se kterým se doposud v jiných jazycích nesetkali.

Druhý blok byl věnován nasazení oblasti 3 a 4, kde po úvodní diskusi a praktické ukázce cyklu foreach začali žáci pracovat na úloze 4.3. U některých žáků dle očekávání nastal problém s chybným zápisem cyklu, neboť používali syntaxi známou z jazyka C#. Po upozornění na chybu všichni tuto úlohu dokončili. Očekávané problémy tedy nastávali kvůli odlišnosti syntaxe v různých jazycích.

Ve druhé hodině bylo představeno vícerozměrné pole v PHP a po praktických ukázkách začali žáci pracovat na zadání úlohy 4.4. Úloha naopak proběhla bez problémů u obou skupin.

Ověření všech úloh proběhlo v pořádku a cíle v tomto metodickém listu byly naplněny.

Celý metodický list byl ověřen během 4 vyučovacích hodin. Uvedená doba je v tomto případě adekvátní a bylo možné vždy na konci bloku o nových zkušenostech více diskutovat.

Tato možnost by byla jistě ztížená, pokud by byla metodika učena ve formě jedné samostatné hodiny.

6.2.5 Textové soubory, řetězce

Tento metodický list byl rozložen do 6 vyučovacích hodin stejnojmenného tématu dle ŠVP.

Během následujícího bloku byla nasazena oblast 1 a 2. V první hodině došlo k prezentaci a demonstraci práce s funkcí `fopen` a poté byla žákům zadána úloha 5.1, během které načítali obsah textového souboru. U některých žáků došlo k chybě na straně funkce `filesize`, kde místo proměnné s adresou souboru vkládali referenci na funkci `fopen`. V tomto případě byla znovu vysvětlena syntaxe těchto funkcí a demonstrován postup. Poté již žáci úlohu dokončili.

Druhá hodina byla rozšířením předchozí a volně navázala na tuto úlohu. Proběhla prezentace a vysvětlení parametrů pro čtení a zápis, demonstrace zápisu funkcí `fwrite` a poté byla žákům zadána úloha 5.2. Žáci na základě již dokončené úlohy 5.1 splnili tuto úlohu bez problémů kromě drobných chyb v podobě chybějících středníků, které byly ihned opraveny.

Během první vyučovací hodiny byla probrána oblast 1 tohoto metodického listu zaměřená na čtení textových souborů. Během hodiny žáci pracovali na úloze 1 v tomto metodickém listu. Během této úlohy nastalo několik problémů zejména v podobě chybné cesty či názvu textového souboru. Také v případě prázdného souboru funkce `fread` vracela chybu nulové délky.

Oblast 3 a 4 byla nasazena v následujícím bloku, kdy po prezentaci a demonstraci funkce `explode` vypracovali žáci zadání úlohy 5.3. Úloha se zdála žákům snadná a časový limit 15 minut byl dostačující. Následovalo představení jednotlivých funkcí pro práci s řetězci a znaky. Po demonstrativních použitích pracovali žáci na zadání úlohy 5.4. Jednotlivé body byly splněny v časovém limitu a v závěru tohoto bloku zbyl čas pro další diskusi nad problematikou kódování znaků s diakritikou.

Závěrečná oblast 5 byla nasazena do dalšího bloku. Na tuto oblast je doporučován rozsah 55-70 minut. Žáci během tohoto bloku pracovali na jednotlivých bodech zadání úlohy 5.5. Část žáků splnila tuto úlohu před koncem první hodiny tohoto dvouhodinového bloku. Zbylá část pro splnění úloh potřebovala přibližně 60 minut. Funkce byly vytvořeny. Drobné

problémy nastaly se zápisem řetězce na začátek souboru. V tomto případě bylo potřeba znovu vysvětlit kroky, vedoucí k dokončení úlohy.

Ověření všech úloh proběhlo v pořádku a cíle v tomto metodickém listu byly nakonec naplněny, ovšem minimální čas u závěrečné úlohy se zdá být u některých žáků nedostatečný. Mohl by být navýšen na hodnotu 60 či 65 minut. U této úlohy je takřka práce ve dvouhodinovém bloku nutností. Pokud by výuka probíhala po samostatných hodinách, je nutné vyhradit čas na připomenutí úlohy a aby se žáci zorientovali po určitém čase ve svém kódu.

6.2.6 Formuláře

Tento metodický list byl rozložen do 9 vyučovacích hodin stejnojmenného tématu dle ŠVP.

Tento metodický list je nejrozsáhlejším a je očekáváno nejvíce možných vzniků problémů.

Během prvního dvouhodinového bloku byla probraná oblast 1 spolu se zadáním úlohy 6.1. Žáci pracovali samostatně, úloha byla splněna v časovém limitu v začátcích druhé vyučovací hodiny. Poté byla nasazena oblast 2 formou prezentace a demonstrace zpracování dat z formuláře. Na úlohu 6.2 nezbyl v tomto bloku čas, což ale odpovídá časovému harmonogramu.

Další blok tedy po teoretickém úvodu začal zadáním úlohy 6.2. Úloha byla žáky v časovém limitu 20 minut splněna. Během tvorby byly jen drobné komplikace dle očekávání na straně syntaxe. Větší komplikace nenastaly. Poté byla nasazena oblast 3 věnována validaci formulářových dat, která díky právě dokončené úloze 6.2 volně navázala v podobě ošetření dat, ochrany před XSS útoky apod. Žákům byla zadána úloha 6.3, která dle zadání přímo vychází z předchozí úlohy. Úloha byla dokončena v časovém limitu a bez problémů.

Třetí blok byl věnován oblasti 4, obsahující úlohu 6.4. Žáci pracovali s již dříve vytvořenými skripty z předchozích úloh. Úloha ovšem vyžadovala více času na zpracování dat, správné použití předchozích funkcí. Žáci měli převážně problém se orientovat v dřívějších kódech, které bylo nutno implementovat do této úlohy. Během celé tvorby byla i dle metodiky demonstrace jednotlivých dílčích kroků, probíhala praktická ukázka použití předchozích funkcí. Žáci nakonec úlohu dokončili. Až na některé žáky byla u zbytku třídy využita takřka maximální hranice 75 minut pro úspěšné dokončení.

Následující oblast 5 byla rozdělena do 3 vyučovacích hodin. Žáci dostali za úkol vytvořit registrační formulář s ukládáním do CSV souborů dle zadání úlohy 6.5. Tato úloha byla u větší části třídy bez pomoci neúspěšná. Problém nastal až při čtení dat z CSV, kdy bylo potřeba rozdělit řetězec a poté ještě první a poslední prvek upravit pomocí funkce substr. Jelikož tento bod v zadání formulován jako „vyjměte data z uvozovek“, žáci přemýšleli nad samotným vyjímáním částí řetězce.

Část žáků tuto úlohu dokončila v tomto bloku, zbytek třídy tuto úlohu dokončoval až ve třetí hodině, věnované této oblasti. S pomocí a opětovným vysvětlením se povedlo úlohu dokončit, ovšem u některých žáků by bez pomoci byla v tomto časovém limitu neřešitelná.

Za chybu většina žáků označila nejasně formulované zadání, kde dlouho přemýšleli nad možným řešením.

Ověření úloh 6.1–6.4 proběhlo v pořádku, problematickou se stala úloha 6.5, kde cíle dosáhlo samostatně jen 6 žáků, zbylí až po určité pomoci ze strany pedagoga a demonstraci jednoho z možných řešení. Cíle v tomto metodickém listu byly naplněny.

Celý metodický list byl ověřen během 9 vyučovacích hodin. Tato doba je pro běžnou tvorbu adekvátní. V minimálním případě by šlo prvním oblastem věnovat opravdu jen minimální stanovený čas a tím by bylo možné celý list v krajním případě odučit i během stanovených 7 hodin.

6.2.7 Hlavičky

Tento metodický list byl ověřen během 1 vyučovací hodiny během výuky tématu dle ŠVP Hlavičky, session.

Tento metodický list byl nasazen druhou vyučovací hodinu po předchozím ne příliš zdařené úloze 6.5. Po prezentaci a demonstraci vybraných možností funkce header dostali žáci zadanou úlohu 7.1, během které modifikovali již hotový skript návštěvní knihy z úlohy 6.4. Tato úloha byla velice snadná, což dle reakce v závěrečné diskusi žáci po předchozí úloze 6.5 uvítali.

Ověření úlohy proběhlo v pořádku u všech přítomných žáků a cíle v tomto metodickém listu byly naplněny.

Celý metodický list byl ověřen během 1 vyučovací hodiny, což je také uvedená minimální doporučená hodinová dotace. V tomto případě je tato minimální hodinová dotace dostačující.

6.2.8 Session

Tento metodický list byl rozložen do 2 vyučovacích hodin během výuky tématu dle ŠVP Hlavičky, session.

Během závěrečného bloku tohoto tematického celku byla nasazena oblast 1 a 2. První hodina věnovaná oblasti 1 seznámila žáky s problematikou relací a superglobální proměnné \$_SESSION na které plynule navázala oblast 2. V této oblasti byla žákům zadána úloha 8.1. Tato úloha byla žáky vypracována v časovém limitu s komplikacemi na straně chybně napsaných názvů klíčů či při použití dřívějších formulářů nepřejmenováním formulářových polí dle zadání. Tyto chyby byly s drobnou pomocí odhaleny.

Dle závěrečné diskuse je často problém s časem, kdy v případě vzniku chyby žáci ztrácejí příliš mnoho času nad hledáním chyby ve skriptu zejména z důvodu krátkého praktického používání jazyka PHP.

Ověření úlohy proběhlo v pořádku a cíle v tomto metodickém listu byly naplněny.

Celý metodický list byl ověřen během 2 vyučovacích hodin, což je i uvedená minimální doporučená hodinová dotace. V tomto případě je tato minimální hodinová dotace dostačující a zbyl dostatek prostoru pro případnou diskusi se žáky probíranou problematikou.

6.2.9 Závěrečná hodina a dotazníkové šetření

Po dokončení všech oblastí závěrečného metodického listu 5.2.8 *Relace – session* byl v následující hodině udělán kolektivní rozbor všech úloh, opakování získaných znalostí a dovedností v jehož závěru žáci vyplnili jednoduchý anonymní dotazník, vytvořený mnou takéž v jazyce PHP s databází MariaDB, pro získání zpětné vazby na průběh ověření této metodiky v praxi. Tento dotazník a jeho výstup je popsán v kapitole 6.3 *Závěrečný dotazník*.

Na konci čtvrtého ročníku prováděli žáci anonymní hodnocení výuky všech předmětů na základě požadavku vedení Střední průmyslové školy, p.o., které je popsáno v kapitole 6.4 *Hodnocení výuky na SPŠ Ostrov*.

6.3 Závěrečný dotazník

Součástí ověření této metodiky byl již zmíněný elektronický anonymní dotazník.

Otázky byly navrženy tak, aby pokryly různé aspekty výuky, včetně teoretické a praktické části, času na vypracování úloh, a celkového přínosu výuky.

Otázky:

1. Považujete dosavadní výuku PHP za přínosnou?
2. Teoretická část výuky byla jasná a zřetelná?
3. Byl poskytnut dostatečný prostor pro případné dotazy a nejasnosti?
4. Praktické příklady byly jasné, srozumitelné?
5. Postupná zvyšující se náročnost příkladů byla přiměřená?
6. Byl dostatek času na vypracování praktických úloh?
7. Použijete získané informace v praxi?
8. Motivovala Vás tato výuka k případné volbě dlouhodobé maturitní práce z PVA?

Každá otázka nabízela pouze volbu mezi čtyřmi povinnými přepínači:

- ano
- spíše ano
- spíše ne
- ne

Na Tento dotazník odpovědělo celkem 12 žáků z první skupiny o 12 žácích a 11 žáků z druhé skupiny o 13 žácích.

Tabulka výsledných počtů reakcí ano, spíše ano, spíše ne a ne u jednotlivých otázek:

Otázka	Počet odpovědí			
	Ano	Spíše ano	Spíše ne	ne
1	21	2	0	0
2	18	5	0	0
3	23	0	0	0
4	19	3	1	0
5	18	5	0	0
6	13	5	5	0
7	18	2	3	0
8	3	4	5	11

Tabulka 2 - počty odpovědí v závěrečném dotazníku

Vyhodnocení tohoto dotazníku

Celkem bylo zpracováno 23 odpovědí.

1. Považujete dosavadní výuku PHP za přínosnou?

Ano: 21 (91,3%)

Spíše ano: 2 (8,7%)

Spíše ne: 0 (0%)

Ne: 0 (0%)

Vyhodnocení: Výsledky ukazují, že většina žáků (91,3%) považuje výuku PHP za přínosnou. Tento výsledek ukazuje, že obsah a způsob výuky touto metodikou jsou efektivní a odpovídají očekáváním žáků.

2. Teoretická část výuky byla jasná a zřetelná?

Ano: 18 (78,3%)

Spíše ano: 5 (21,7%)

Spíše ne: 0 (0%)

Ne: 0 (0%)

Vyhodnocení: Teoretická část výuky je pro většinu žáků (78,3%) jasná a zřetelná. Nikdo neoznačil teoretickou část jako nejasnou, což naznačuje, teoretický obsah v této metodice je srozumitelný a dobře strukturovaný.

3. Byl poskytnut dostatečný prostor pro případné dotazy a nejasnosti?

Ano: 23 (100%)

Spíše ano: 0 (0%)

Spíše ne: 0 (0%)

Ne: 0 (0%)

Vyhodnocení: Všichni žáci se shodují, že byl poskytnut dostatečný prostor pro dotazy a nejasnosti.

4. Praktické příklady byly jasné, srozumitelné?

Ano: 19 (82,6%)

Spíše ano: 3 (13,05%)

Spíše ne: 1 (4,35%)

Ne: 0 (0%)

Vyhodnocení: Většina žáků (82,6%) považuje praktické příklady za jasné a srozumitelné. Jedna odpověď označila příklady jako spíše nesrozumitelné, což může naznačovat potřebu zlepšení v prezentaci a demonstraci praktických úkolů.

5. Postupná zvyšující se náročnost příkladů byla přiměřená?

Ano: 18 (78,3%)

Spíše ano: 5 (21,7%)

Spíše ne: 0 (0%)

Ne: 0 (0%)

Vyhodnocení: Náročnost příkladů byla pro většinu žáků přiměřená, což naznačuje, že žáci byli schopni zpracovávat gradující obtížnost úloh bez větších problémů.

6. Byl dostatek času na vypracování praktických úloh?

Ano: 13 (56,5%)

Spíše ano: 5 (21,7%)

Spíše ne: 5 (21,7%)

Ne: 0 (0%)

Vyhodnocení: Zatímco přibližně polovina žáků (56,5%) měla jasný dostatek času na vypracování úloh, necelá čtvrtina žáků (21,7%) již pociťuje, že času na vypracování by mohlo být o něco více a poslední necelá čtvrtina (21,7%) žáků se domnívá, že stanovený čas na vypracování úloh je krátký. Tento výsledek naznačuje, že by mohlo být užitečné přehodnotit časový rámec pro praktické úlohy v této metodice.

7. Použijete získané informace v praxi?

Ano: 18 (78,3%)

Spíše ano: 2 (8,7%)

Spíše ne: 3 (13,0%)

Ne: 0 (0%)

Vyhodnocení: Většina žáků (78,3%) plánuje použít získané informace v praxi. To ukazuje na praktickou hodnotu výuky a její důležitost pro budoucí využití v praxi.

8. Motivovala Vás tato výuka k případné volbě dlouhodobé maturitní práce z PVA?

Ano: 3 (13%)

Spíše ano: 4 (17,4%)

Spíše ne: 5 (21,7%)

Ne: 11 (47,8%)

Vyhodnocení: Výsledek ukazuje, že jen malá část studentů byla motivována k volbě dlouhodobé maturitní práce z PVA. Téměř polovina studentů (47,8%) nebyla motivována.

Vysvětlení možných negativních reakcí u motivace žáků k volbě dlouhodobé maturitní práce z PVA je, že dlouhodobá maturitní práce je rozsáhlý projekt, kterému musí být věnováno opravdu velké úsilí, a kromě samotné aplikace je potřeba vytvořit také dokumentaci a celou práci prezentovat. Žáci často volí pro ně snadnější jednodenní praktickou maturitní zkoušku. Dlouhodobou maturitní práci si často vybírají jen jednotlivci ze tříd, řádově dva až pět žáků z celé třídy, a volí si témata z předmětu PVA i PGV, tedy tvorby desktopové či webové aplikace.

Naopak 3 respondenti odpovídající ano a 4 odpovídající spíše ano jsou v tomto směru z mého hlediska úspěchem.

Dnes vím, že tuto dlouhodobou maturitní práci z předmětu PVA si z této třídy vybrali původně čtyři žáci, ovšem dva z ní ještě odstoupili pro účast na jiném projektu (Technowizz). Dva zbylí žáci úspěšně složili zkoušku, jeden s takřka maximálním ziskem všech možných bodů.

6.4 Hodnocení výuky na SPŠ Ostrov

Na Střední průmyslové škole Ostrov, p.o. probíhá každé dva roky anonymní hodnocení výuky ze strany žáků na všechny předměty, které absolvovali. Ve výsledku může každý pedagog vidět svůj průměr ze všech předmětů ve srovnání se školním průměrem, či v rámci předmětu vůči ostatním pedagogům, pokud jej také vyučují. Výsledky jsou dostupné nejen učitelům, ale slouží pro vedení školy, na základě nich poté probíhají i další pohovory.

Toto hodnocení probíhalo v následujícím školním roce a bylo vztaženo na výuku tohoto předmětu ve třetím a čtvrtém ročníku.

Hodnocení probíhá na adrese <https://home.spsostrov.cz/hodnoceni-vyuky/>

Po přihlášení pedagoga je možno vidět aktuální a archivní výsledky.

Otázky v tomto dotazníku jsou:

Vedení a organizace vyučovací hodiny

1. Učitel látku vysvětluje takovým způsobem, že ji dokáže pochopit

2. Hodiny jsou záživné a zajímavé, není to ztráta času
3. Při výuce jsou daná jasná pravidla a jsou dodržována

Přístup k hodnocení žáků

4. Hodnocení při zkoušení je objektivní a spravedlivé
5. Vím přesně, co se mám naučit, abych uspěl
6. Zkoušení odpovídá probíranému učivu

Odbornost učitele

7. Učitel chodí do hodin připraven
8. Učitel rozumí tomu, co vyučuje

Chování k žákům a autorita učitele

9. Vážíme si učitele jako osobnosti
10. Dokáže si udržet ve třídě klid
11. Chová se k nám vstřícně a slušně
12. Dokáže žáka pochválit

Legenda:

- 1 = ano
2 = spíše ano
3 = někdy ano, někdy ne
4 = spíše ne
5 = ne

Tisk do PDF z tohoto hodnocení je přílohou této práce¹²¹.

Statistika hodnocení předmětu v porovnání se všemi předměty:

¹²¹ Soubor hodnoceni-sps.pdf v příloze této práce

Statistika - 2023/2024														
Třída	Hodnotící žáci	Předmět	1	2	3	4	5	6	7	8	9	10	11	12
I4	22	PVA												
			1.36 (100%)	1.27 (100%)	1 (100%)	1 (100%)	1 (100%)	1 (100%)	1 (100%)	1 (100%)	1 (100%)	1.14 (100%)	1 (100%)	1.09 (100%)

Obrázek 1 - celkové hodnocení předmětu PVA třídou I4

Detaily - 2023/2024														
Žák	1	2	3	4	5	6	7	8	9	10	11	12	Vzkaz	
1. žák	1	1	1	1	1	1	1	1	1	1	1	1		
2. žák	1	1	1	1	1	1	1	1	1	1	1	1		
3. žák	3	2	1	1	1	1	1	1	1	1	1	1		
4. žák	2	2	1	1	1	1	1	1	1	3	1	1		
5. žák	2	2	1	1	1	1	1	1	1	1	1	1		
6. žák	1	1	1	1	1	1	1	1	1	1	1	1		
7. žák	1	1	1	1	1	1	1	1	1	1	1	1	Více zábavné a naučné hodiny jsem nikdy v životě neměl. Budou mi tyto hodiny s Vámi velmi chybět.	
8. žák	1	1	1	1	1	1	1	1	1	1	1	1		
9. žák	2	2	1	1	1	1	1	1	1	1	1	1	PHP master	
10. žák	2	1	1	1	1	1	1	1	1	2	1	1		
11. žák	2	1	1	1	1	1	1	1	1	1	1	2		
12. žák	1	1	1	1	1	1	1	1	1	1	1	1		
13. žák	1	2	1	1	1	1	1	1	1	1	1	2		
14. žák	1	1	1	1	1	1	1	1	1	1	1	1		
15. žák	1	1	1	1	1	1	1	1	1	1	1	1	Velmi zkušený učitel s přátelskou a sympatickou osobností. Schopen vyložit tematiku velmi dobře.	
16. žák	1	1	1	1	1	1	1	1	1	1	1	1		
17. žák	1	1	1	1	1	1	1	1	1	1	1	1		
18. žák	1	1	1	1	1	1	1	1	1	1	1	1		
19. žák	1	1	1	1	1	1	1	1	1	1	1	1		
20. žák	1	1	1	1	1	1	1	1	1	1	1	1	Weby umíme	
21. žák	1	1	1	1	1	1	1	1	1	1	1	1	Děkuji za 3 roky zábavné a prakticky využitelné výuky. Váš individuální přístup k žákům je skvělý, díky němu se na hodinách nikdo nenudí a prakticky nikdo není pozadu. Pokud někoho výuka nebaví a chce se soustředit na jiné předměty, netrápíte ho, což je také dost nevidané a příjemné. Mimo to také děkuji za vedení mé DMP, A. W.	
22. žák	2	2	1	1	1	1	1	1	1	1	1	1	Stejně jako AVT :D	

Obrázek 2 - Detail hodnocení předmětu PVA třídou I4

6.5 Vyhodnocení ověření

Podle hodnocení výuky, výsledků vzdělávání a závěrečného dotazníku lze říci, že metodika je použitelná v praxi s drobnými změnami zejména v její minimální časové dotaci u některých metodických listů či oblastí.

Její obsah a příklady mohou vést k získání dovedností žáků v rámci tvorby skriptovacího jazyka PHP. Takto navržené metodické listy na sebe obsahově navazují a postupně prohlubují znalosti až k tvorbě jednoduché funkční aplikace.

U této metodiky nastává problém s poměrně rychlým vývojem jazyka PHP. V době ověřování této metodiky například neexistovala funkce pro automatickou tvorbu velkých a malých počátečních písmen ve variantě s možností definovat kódování znaků. Při finalizaci této práce jsou tyto funkce dostupné jako `mb_ucfirst` a `mb_lcfirst`, přesněji od verze PHP 8.4.0, která byla v prvotní testovací verzi vydána v červnu 2024. (The PHP Group, 2024)

Tento problém je jistě i problémem většiny metodik a publikací, které se zabývají programovacími jazyky.

Nutno podotknout, že třída, se kterou byla metodika ověřována, patřila a doposud i statisticky patří k jedné z nejlepších tříd na škole, neboť více jak polovina třídy pobírala díky prospěchu do průměru 1,5 i školní prospěchová stipendia během celé docházky.

Tato třída během své školní docházky pracovala na několika projektech v rámci SOČ (Středoškolská odborná činnost), kde někteří žáci postoupili i do celorepublikového kola¹²²
¹²³.

Další úspěch spojený s tímto předmětem je vítězství v soutěži Technowizz ve školním roce 2022/2023 skupinou z této třídy, kde byla část vědomostí, získaných při nasazení metodiky, užita k řešení problematiky. (Střední průmyslová škola Ostrov, p.o., 2023)

Veliký úspěch těchto žáků byl bez pochyby v roce 2023/2024 těsně před maturitní zkouškou, kdy pod vedením kolegy, vyučujícího paralelně předmět Programové vybavení, pracovali žáci ve třech týmech na zadání jednotlivých úloh soutěže Technowizz, kde využívali i

¹²² https://www.soc.cz/dokumenty/vysledkove_listiny/2023-24.pdf

¹²³ https://www.soc.cz/dokumenty/vysledkove_listiny/2022-23.pdf

znalosti ohledně databází a práce s daty z této metodiky. Všechny tři týmy vyhrály první místo v jednotlivých kategoriích této soutěže. (Ústecká komunitní nadace, 2024)

Vzhledem k těmto skutečnostem může být celkové vnímání snadné obtížnosti u některých úloh značně ovlivněno.

Závěr

Analýzou kurikulárních dokumentů bylo zjištěno, že rámcové vzdělávací programy nabízejí prostor pro výuku skriptovacích jazyků. Školy mohou této výuce věnovat i v rámci disponibilních hodin větší hodinovou dotaci, než určuje zmíněný rámec.

Na Střední průmyslové škole Ostrov, p.o., je věnována výuce skriptovacího jazyka PHP dle platného ŠVP dostatečná doba pro nasazení této metodiky a žáci mají dostatek prostoru pro postupné získání dovedností z této problematiky.

V praktické části byla navržena kompletní metodika výuky tematického celku *III. Programování v PHP* spolu se sadou gradujících navazujících úloh, která byla následně nasazena v praktické výuce a ověřena na výše zmíněné škole.

Nasazením metodiky do praxe včetně navrhované série úloh bylo zjištěno, že úlohy jsou, až na úlohu 6.5, proveditelné bez větších komplikací. Většina očekávaných problémů v praxi nenastala. Z dotazníkového šetření vyplývá, že takřka čtvrtina třídy by uvítala více času pro jejich vypracování, ovšem tři čtvrtiny třídy považují čas za dostačující.

Úloha 6.5 v metodickém listu 5.2.6 *Formuláře* byla z hlediska proveditelnosti nejsložitější a na základě následné diskuse a hodnocení žáků byla její formulace příliš obecná, a tím někteří žáci nebyli schopni úlohu bez pomoci dokončit. Pro praktické použití této úlohy by bylo vhodné jasně formulovat jednotlivé body zadání, které povedou žáky k dosažení cíle.

Z praktického nasazení metodiky dále vyplynulo, že výuka formou dvouhodinových bloků se jeví efektivnější. Díky gradaci náročnosti a návaznosti jednotlivých oblastí a úloh je v případě vícehodinového bloku navazující kapitola žáky lépe přijímána. V případě, že došlo k rozdělení jedné oblasti do hodin, které na sebe v rozvrhu nenavazují, bylo nutné věnovat více času úvodu hodiny a připomenout žákům problematiku z předchozí hodiny.

Následnou analýzou bylo zjištěno, že stanovené doporučené časy v jednotlivých metodických listech jsou dostačující.

Výuka formou lokálních serverů byla úspěšná. Žákům vyhovovalo přímé ukládání souborů do adresáře na lokálním pevném disku. Jako další použitelná pomůcka se ukázal online Tryit Editor, zmíněný v kapitole 3.4.4 *Online prostředí pro interpretaci skriptů – W3Schools.com*.

V rámci zpětné vazby ze závěrečného dotazníku plyne, že obsah a teoretická část odpovídá očekáváním žáků a její přijímání je ze strany žáků kladné.

Ověřilo se, že výuka skriptovacího jazyka PHP na střední škole je dle této metodiky pochopitelná a zvládnutelná.

Seznam použitých zkratk

3NF – Třetí normální forma

ACID – Atomicity, Consistency, Isolation, Durability

ASC – Ascending order

ASP – Active Server Pages

API – Application programming interface

C# – C Sharp

CSS – Cascading Style Sheets

DDL – Data definition language

DESC – Descending order

DML – Data manipulation language

DOM – Document Object Model

FTP – File Transfer Protocol

HTML – Hypertext Markup Language

ICT – Information and Communication, Informační a komunikační technologie

IDE – Integrated Development Environment, vývojové prostředí

IoT – Internet of Things

IT – Informační technologie

IVP – Individuální vzdělávací plán

JSP – JavaServer Pages

LDAP – Lightweight Directory Access Protocol

MŠMT – Ministerstvo školství, mládeže a tělovýchovy

NPI – Národní pedagogický institut

PDF – Portable Document Format

PDO – PHP Document Object

PGV – Programové vybavení

PL/SQL – Procedural Language/Structured Query Language

PVA – Programování a vývoj aplikací

PHP – PHP: Hypertext Preprocessor, dříve Personal Home Page

RVP – Rámcový vzdělávací program

RVP SOV – Rámcový vzdělávací program středního odborného vzdělávání

SCP – Secure Copy

SPŠ – Střední průmyslová škola

SQL – Structured Query Language

SSL – Secure Sockets Layer

SVP – Speciální vzdělávací potřeby

ŠVP – Školní vzdělávací program

TLS – Transport Layer Security

UNF – Nenormalizovaná databáze

URL – Uniform Resource Locator

XML – Extensible Markup Language

Seznam použitých informačních zdrojů

Beneš, Miroslav. 2024. *Skriptovací jazyky*. Ostrava : FEI VŠB-TU Ostrava, 2024.

Durdilová, Ivana. 2012. Skriptovací jazyky. [Online] 2012.

<http://uzlabina2.aspone.cz/jazykyskriptovaci.aspx>. [Citace: 20. 11 2024.]

Grud, David. 2024. Nette. [Online] 2024. <https://nette.org/cs/>. [Citace: 20. 11 2024.]

Jan Fiala PSPad. 2023. Textový editor PSPad. [Online] 2023.

<https://www.pspad.com/cz/>. [Citace: 20. 11 2024.]

JetBrains s.r.o. 2024. Free Educational Licenses - Community Support. [Online] 2024.

<https://www.jetbrains.com/community/education/#students>. [Citace: 20. 11 2024.]

JetBrains: Developer Tools for Professionals and Teams. 2023. PHP Programming - The State of Developer Ecosystem in 2023 Infographic. [Online] 2023.

<https://www.jetbrains.com/lp/devecosystem-2023/php/>. [Citace: 20. 11 2024.]

Kašparová, Jana. 2012. *METODIKA TVORBY ŠKOLNÍCH VZDĚLÁVACÍCH PROGRAMŮ SOŠ a SOU*. Praha : Národní ústav pro vzdělávání, školské poradenské zařízení a zařízení pro další vzdělávání pedagogických pracovníků, 2012.

Lutz, Mark. 2013. *Learning Python*. místo neznámé : O'Reilly Media, Inc, USA, 2013.

MariaDB Foundation. 2024. MariaDB in brief - MariaDB.org. [Online] 2024.

<https://mariadb.org/en/>. [Citace: 20. 11 2024.]

Medium. 2022. The Best Databases for PHP. [Online] 13. 10 2022.

<https://techbelle.medium.com/the-best-databases-for-php-1f7054d7042f>. [Citace: 20. 11 2024.]

MŠMT a NPI České republiky. 2024. Aktualizované rámcové vzdělávací programy SOV. [Online] 2024. <https://revize-sov.edu.cz/revize-rvp-sov>. [Citace: 20. 11 2024.]

MŠMT. 2023. *Rámcový vzdělávací program pro obor vzdělání 18 – 20 – M/01 Informační technologie*. Praha : MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY, 2023.

MŠMT. 2020. *Rámcový vzdělávací program pro obor vzdělání 18 – 20 – M/01 Informační technologie.* Praha : MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY, 2020.

MŠMT. 2020. *Strategie vzdělávací politiky České republiky do roku 2030+.* Praha : MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY, 2020.

MŠMT. 2020. Bílá kniha - Národní program rozvoje vzdělání v ČR [Online] 2020.
<https://msmt.gov.cz/vzdelavani/skolstvi-v-cr/bila-kniha-narodni-program-rozvoje-vzdelani-v-cr>. [Citace: 20. 11 2024.]

Národní pedagogický institut České republiky. 2024. *Rámcové vzdělávací programy.* [Online] 2024. <https://www.npi.cz/ramcove-vzdelavaci-programy>. [Citace: 20. 11 2024.]

netcraft. 2024. June 2024 Web Server Survey. [Online] 2024.
<https://www.netcraft.com/blog/june-2024-web-server-survey/>. [Citace: 20. 11 2024.]

Radix. 2024. Best PHP Frameworks to Consider for Web Development in 2024. [Online] 2024. <https://radixweb.com/blog/best-php-frameworks>. [Citace: 20. 11 2024.]

Refsnes Data. 2024. PHP Installation. [Online] 2024.
https://www.w3schools.com/php/php_install.asp. [Citace: 20. 11 2024.]

Stack Overflow. 2024. Developer Survey 2023. *Stack Overflow.* [Online] 2024.
<https://survey.stackoverflow.co/2023/#technology>. [Citace: 20. 11 2024.]

Střední průmyslová škola Ostrov, p.o. 2023. Technowizz 2022. [Online] 2023.
<https://spsostrov.cz/vedomostni-souteze/technowizz-2022>. [Citace: 20. 11 2024.]

Střední průmyslová škola Ostrov, p.o. 2024. Webový server pro studenty a učitele SPŠ Ostrov. [Online] 2024. <https://home.spsostrov.cz/docs/web-howto>. [Citace: 20. 11 2024.]

Střední průmyslová škola Ostrov, příspěvková organizace. 2022. *Školní vzdělávací program INFORMAČNÍ TECHNOLOGIE obor 18-20-M/01.* Ostrov : Střední průmyslová škola Ostrov, příspěvková organizace, 2022.

Střední průmyslová škola Ostrov, p.o. 2018. *Školní vzdělávací program INFORMAČNÍ TECHNOLOGIE obor 18-20-M/01.* Ostrov : Střední průmyslová škola Ostrov, příspěvková organizace, 2018.

The Apache Software Foundation. 2024. APACHE HTTP SERVER PROJECT.

[Online] 2024. <https://httpd.apache.org/>. [Citace: 20. 11 2024.]

The PHP Documentation Group. 2020. History of PHP. [Online] 2020.

<https://www.php.net/manual/en/history.php.php>. [Citace: 20. 11 2024.]

The PHP Group. 2024. News Archive - 2024. [Online] 2024.

<https://www.php.net/archive/2024.php#2024-11-21-4>. [Citace: 20. 11 2024.]

The PHP Group. 2024. PHP 8.4 Preparation Tasks. [Online] 2024.

<https://wiki.php.net/todo/php84>. [Citace: 20. 11 2024.]

Ústecká komunitní nadace. 2024. Výsledky soutěže Technowizz. [Online] 2024.

<https://www.komunitninadace.cz/zadatel/koho-podporujeme/soutez-technowizz/vysledky-souteze-technowizz/>. [Citace: 20. 11 2024.]

W3Techs. 2024. Usage statistics of PHP for websites. [Online] 2024.

<https://w3techs.com/technologies/details/pl-php>. [Citace: 20. 11 2024.]

websupport. 2023. MariaDB a MySQL: Porovnání, výkon, historie. [Online] 2023.

<https://www.websupport.cz/blog/2023/09/mariadb-a-mysql-porovnani-vykon-historie/>. [Citace: 20. 11 2024.]

Wikipedie: Otevřená encyklopedie. 2024. Objektově orientované programování.

[Online] 2024. [Citace: 20. 11 2024.]

https://cs.wikipedia.org/wiki/Objektiv%C4%9B_orientovan%C3%A9_programov%C3%A1n%C3%AD.

XAMPP. 2024. XAMPP Apache + MariaDB + PHP + Perl. [Online] 2024. [Citace: 20. 11 2024.]

<https://www.apachefriends.org/>.

Vyjádření k využití nástrojů umělé inteligence

Prostředky umělé inteligence byly využity k překladu abstraktu do anglického jazyka formou překladače DeepL dostupný na adrese: <https://www.deepl.com/en/translator>

Druhý prostředek Microsoft Copilot byl využit k seřazení seznamu použitých zkratk, který jsem během tvorby práce postupně plnil celkem 40 zkratkami, které jsem poté nechal abecedně seřadit tímto nástrojem na adrese: <https://copilot.microsoft.com/>

Dále byl nástroj Copilot použit pro získání nápovědy a případných odkazů na správnou tvorbu abstraktu a klíčových slov.

Seznam příloh

Příloha 1 – Tiskové do PDF výsledků hodnocení výuky předmětu PVA u třídy I4 na Střední průmyslové škole Ostrov, p.o.

Seznam obrázků

Obrázek 1 - celkové hodnocení předmětu PVA třídou I4	110
Obrázek 2 - Detail hodnocení předmětu PVA třídou I4	110

Seznam tabulek

Tabulka 1 - Tématický plán PVA 3. ročník.....	33
Tabulka 2 - počty odpovědí v závěrečném dotazníku	105