Posudek bakalářské práce

předložené na Matematicko-fyzikální fakultě
Univerzity Karlovy v Praze

☐ posudek vedoucího ☑ posudek oponenta

Autor/ka:	**Šimon Rajčan**
Název práce:	**Adaptivní klasifikátor pošty pro IMAP servery**
Studijní program a obor:	**Informatika, Obecná informatika**
Rok odevzdání:	**2009**

Jméno a tituly vedoucího/oponenta: RNDr. David Obdržálek

Pracoviště: KSI MFF UK

	excelentní	**odpovídající**	**slabší**	**nevyhovující**
Náročnost zadaného tématu		x		
Míra splnění zadání			x	
Rozsah práce			x	
Struktura textové části práce		x	x	
Analýza			x	x
Vývojová dokumentace			x	
Uživatelská dokumentace			x	
Jazyková a typografická úroveň			x	
Návrh a design implementace			x	
Kvalita zpracování softwarové části				x
Stabilita aplikace		?	?	

Nejvýznamnější klady:

Implementace běžně nepoužívaného způsobu filtrování pošty výběrem důležitých zpráv na rozdíl od standardní detekce spamu resp. závadných zpráv.

Nejzávažnější nedostatky:

Zdá se, že práce byla řešena „cestou nejmenšího odporu". Vzniklé softwarové dílečko je rozsahem na úrovni zápočtového programu, kvalita zpracování špatná a uživatelský komfort nízký. K tomu autorem deklarovaná nutnost spouštět program z vývojového prostředí MS Visual Studio 2008 – a to by dle mého názoru mohlo stačit pro neuznání díla jako výsledku bakalářské práce Informatického oboru na MFF UK.
Nutnou podmínkou je navíc použití placené knihovny třetí strany resp. 30-denní zkušební verze, u níž autor na str.20 dokonce popisuje jak obejít ochranu a tím porušit podmínky licence!

Další poznámky:

- V kap. 2.4 je konstatováno, že je možné poštu třídit pouze do 5 kategorií, neboť při vyšším počtu by filtr „mal malú efektivitu". Není jasné, co tím autor míní a také není uvedeno, jak k takovému závěru došel. Tomuto limitu pak je podřízena i implementace, kde je limit použit hard-coded způsobem (např. deklarací pěti úložišť jako samostatných proměnných, okopírováním pěti stejných metod pro zpracování událostí od tlačítek formuláře lišících se pouze číselnou konstantou atd).
- Autor zřejmě nikdy neviděl RFC1939 - Post Office Protocol - Version 3, protože jinak by se neodvážil v 1. kapitole tvrdit, že POP3 „nevie stiahnuť vybranú, ale iba všetku poštu" (ale příkaz RETR x vrátí zprávu s číslem x, tj. vybranou) a že „po prenesení správ zo serveru na klienta sa správy zo serveru vymažú" (použití příkazu RETR přece nijak neimplikuje vyvolání příkazu DELE).
- Heslo uživatele je ukládáno v XML souboru s daty zcela volně a je povinné jej uvést. Uživatelská data, která jsou zadána při vytváření uživtele, je pak možné měnit jen ruční editací zmíněného XML souboru (jeho struktura také není dle mého šťastně navržena, neboť související informace jsou „transponovány" a uvedeny postupně v různých kategoriích).
- V textu není nijak vysvětlen způsob ukládání naučených dat.
- Typografie vzorečků v textu působí velmi „humpolácky" (viz např. str. 10).
- Přiložené CD obsahuje sice spustitelný soubor, ale pouze ve verzi Debug.

Po stránce programátorské kód rozhodně neukazuje na to, že autor pochopil a používá základní programátorské návyky vyučované na MFF UK:

- Aplikace evidentně vznikala metodou copy-paste (viz např. funkce readBodyWords třídy MyDictionary):

```
case "pocetSlovKat0":
      {
            //posuniem sa na ten string
            reader.Read();
            readPocetSlovKatX(Convert.ToInt64(reader.Value), ref pocetSlovBodyKatX, 0);
            break;
      }
case "pocetSlovKat1":
      {
            //posuniem sa na ten string
            reader.Read();
            readPocetSlovKatX(Convert.ToInt64(reader.Value), ref pocetSlovBodyKatX, 1);
            break;
      }
```

atd. až do `pocetSlovKat4`, jednotlivé bloky se liší pouze posledním parametrem.

- Reakce na události peti tlačítek (5 viz výše) je vytvořena jako pět samostatných funkcí, které se liší pouze číselm odpovídajícím pořadí tlačítka. Způsob jejich zobrazování je také poněkud netradiční:

```
private void showSortingButtons()
 {
     hideSortingButtons();
     if (selectedUser_.categories.Count > 0)
     {
         buttonKat0.Visible = true;
         buttonKat0.Text = selectedUser_.categories[0];
     }
     if (selectedUser_.categories.Count > 1)
     {
         buttonKat1.Visible = true;
         buttonKat1.Text = selectedUser_.categories[1];
     }
     if (selectedUser_.categories.Count > 2)
     {
         buttonKat2.Visible = true;
         buttonKat2.Text = selectedUser_.categories[2];
     }
     if (selectedUser_.categories.Count > 3)
     {
         buttonKat3.Visible = true;
         buttonKat3.Text = selectedUser_.categories[3];
     }
     if (selectedUser_.categories.Count > 4)
     {
         buttonKat4.Visible = true;
         buttonKat4.Text = selectedUser_.categories[4];
     }
 }
```

- Některé části vypadají na první pohled podivně, např:

```
reader.MoveToFirstAttribute();
hodnoty.Add(Convert.ToInt32(reader.Value));
//prvu som tam uz pridal zetaz ich musim pridat uz len n-1
for (int i = 1; i < numCategories; i++)
{
    reader.MoveToNextAttribute();
    hodnoty.Add(Convert.ToInt32(reader.Value));
}
```

- Je používán přiřazovací příkaz += 1 namísto operátoru ++ a parametry předávány do metody pomocí proměnné třídy obsahující tuto metodu.
- Menu v uživatelské aplikaci je vytvořeno přímo programem (místo standardního vytvoření v editoru vývojového prostředí)

Další poznámky:

Stabilitu a kvalitu aplikace nebylo možné otestovat, neboť by to vyžadovalo delší provozování (aby se mohlo projevit naučené filtrování). Není také možné odhadnout, jak se bude aplikace chovat po delší době provozu, tj. s tisíci až miliony záznamů v uživatelských slovnících.

	výborně	**velmi dobře**	**dobře**	**neprospěl/a**
Návrh známky				**x**

Datum: 18.6.2009
Podpis: