

S dynamickými programovacími jazyky je možné psát kód bez typové informace a typy proměnných se mohou měnit za běhu. Přestože se dynamické typování snadněji používá a je vhodné pro rychlé prototypování, dynamicky typovaný kód může být více náchylný k programátorským chybám a představuje nelehký úkol pro překladače nebo interpretry. Vývojáři často používají dokumentační komentáře pro explicitní uvedení typové informace, nicméně dostupné nástroje většinou nekontrolují shodu mezi komentáři a vlastním kódem. V této práci se zaměřujeme na jeden z nejpoužívanějších dynamických programovacích jazyků: PHP. V rámci této práce jsme vyvinuli framework pro statickou analýzu PHP kódu jako část projektu Phalanger -- překladače PHP do .NET. I když, tento framework podporuje jakýkoliv druh statické analýzy, implementovali jsme především typovou analýzu za účelem odhalení typových chyb a nekonzistence kódu s dokumentačními komentáři. S pomocí této analýzy jsme odhalili několik reálných chyb a nekonzistencí s dokumentací v kódu několika reálných PHP projektů s dobrým poměrem falešně pozitivních chyb.