

Implementační dokumentace k 2. úloze do IPP 2021/2022

Jméno a příjmení: David Chocholatý

Login: xchoch09

1 Úvod

Následující dokumentace popisuje implementaci řešení 2. úlohy do předmětu IPP pro akademický rok 2021/2022, která implementuje interpret XML reprezentace kódu jazyka IPPcode22. V sekci 2.2 *Bonusové rozšíření NVI* je také uveden podrobný popis řešení rozšíření pro objektově orientovaný návrh. Úloha byla implementována skripty v jazyce Python 3 a spouštěna pomocí verze Python 3.8.

2 Implementace

2.1 Postup řešení

Hlavní skript se nachází v souboru `interpret.py`. Tento skript implementuje vysoko-úrovňové funkce pro obsluhu uživatelských argumentů, vstupních souborů a zpracování vstupních dat. Obsluha pro uživatelské argumenty je implementována v souborech `argument_handler.py` a `argument_parser.py`. Řízení práce se vstupními soubory zajišťuje implementace obsažená v souboru `file_handler.py`. Celková struktura tříd zajišťující obsluhu uživatelských argumentů a vstupních souborů je zaobalena dle návrhového vzoru *Tovární metoda* (*bonusové rozšíření NVI*), jenž se nachází v souboru `handler.py`. Samotné zpracování vstupních XML dat je implementováno v souboru `xml_parser.py`. Dále funkce pro přípravu a samotné spuštění interpretu implementuje zdrojový kód obsažený v souboru `interpret_util.py`. Instrukci obsahující operační kód, případně s jejími dalšími argumenty, reprezentuje třída *Instruction* implementována ve stejnojmenném souboru `instruction.py`. Pomocné metody pro práci s instrukcí jsou uloženy v souboru `instruction_util.py`. Jedná se především o práci s proměnnými a literály obsaženými v instrukci v podobě atributů. Samotný atribut (*operand*) instrukce je reprezentován třídou *Operand*, která se nachází v souboru `operand.py`. Ta zajišťuje vytvoření operandu a nastavení jeho vlastností. Kromě hlavní struktury programu jsou vytvořeny i další skripty pro dodatečné struktury a metody potřebné pro interpret. Mezi tyto struktury patří vlastní výjimky uložené v souboru `custom_exception.py`, třída reprezentující hodnotu *nil* implementovaná v souboru `nil.py` a instrukční sada (soubor `instruction_set.py`). Jako poslední doposud nezmíněná implementace jsou chybové kódy interpretu nacházející se v souboru `exit_code.py`.

2.2 Bonusové rozšíření NVI

Při implementaci pomocí objektově orientovaného návrhu byl využit návrhový vzor *Tovární metoda*. Základní struktura návrhového vzoru je implementována v souboru `handler.py` a poté jsou konkrétní implementace uloženy v souborech `argument_handler.py` a `file_handler.py`. Hlavní třídou návrhu je třída *Creator* deklarující tovární metodu pro vytvoření objektu konkrétní třídy a metodu pro získání obsluhy. Dále v základní struktuře návrhového vzoru je implementována třída *Handler* deklarující metodu pro získání obsluhy dle konkrétní třídy. Obsluhu argumentů implementuje třída *ArgumentHandler* nacházející se ve zmíněném souboru `argument_handler.py`. Tato třída deklaruje inicializační funkci pro vytvoření argumentů a obsahuje metodu pro získání obsluhy vstupních uživatelských argumentů. Instance zmíněné třídy *ArgumentHandler* je vytvořena pomocí tovární metody deklarované ve třídě *ArgumentCreator* odvozené od třídy *Creator*. Jako druhá konkrétní implementace dle návrhového vzoru *Tovární metoda* je implementována obsluha souborů. Implementace se nachází ve zmíněném souboru `file_handler.py`. Podobně jako obsluha argumentů obsahuje inicializační metodu pro vytvoření obsluhy souborů a metodu pro její získání. Obdobně je instance třídy *FileHandler* vytvářena pomocí tovární metody třídy *FileCreator*. Použití návrhového vzoru *Tovární metoda* v konkrétním případě je výhodné z důvodu, že obě konkrétní obsluhy, ať již argumentů nebo souborů, z hlediska logiky zajišťují stejnou funkcionalitu a zároveň je nutné implementovat totožné metody. Vše tedy zajišťuje továrna vytvářející dva druhy produktu z pohledu návrhového vzoru *Tovární metoda*.