

Bazel – zadania kwalifikacyjne

4 maja 2022

1 Scrabble

Celem tego zadania jest weryfikacja podstawowych umiejętności programowania w języku Python. Stanowi również podstawę do zadania kolejnego.

1.1 Punktacja (1 pkt)

W języku Python zaimplementuj funkcję `punkty`, która wylicza, ile dane na wejściu słowo jest warte punktów w grze Scrabble.

Istotne są wyłącznie punkty za litery (a nie za położenie na planszy). Potrzebna będzie obsługa co najmniej dwóch różnych języków – w tym angielskiego (jako domyślnego). Sygnaturę funkcji opracuj według własnego uznania. Niezbędne liczby określające punktację znajdziesz w ogólnodostępnych źródłach danych.

1.2 Test (1 pkt)

W osobnym pliku napisz test do opracowanej funkcji `punkty`.

Nie musisz korzystać z żadnego frameworka do testowania, choć jest to również możliwe (np. `unittest`). Przez test rozumiany jest każdy program, który w przypadku wykrycia nieprawidłowej implementacji zakończy pracę z kodem wyjścia różnym od zera. Przykładowym, choć krótkim testem jest:

```
from dodawator import dodaj
if dodaj(3, 4) != 7:
    exit(1)
```

Według uznania dobierz liczbę testowanych wywołań. W wywołaniach funkcji `punkty` użyj koniecznie nazwanych argumentów. Nie zapomnij sprawdzić działania wartości domyślnej języka.

2 Beatnik

Celem tego zadania jest weryfikacja ogólnego zrozumienia procesów budowania oprogramowania oraz przygotowanie narzędzi niezbędnych do obsługi niestandardowego języka. Na warsztacie posłużą one jako przykład do tworzenia nowych reguł budowania.

2.1 Interpreter (1 pkt)

Zaimplementuj interpreter języka *Beatnik*.

Załóż, że nazwa pliku do wykonania zostanie podana jako argument wywołania (`sys.argv[1]`). Pomocny może być opis języka z [Wikipedii](#). Wykorzystaj (zaimportuj) kod opracowany w zadaniu 1.

2.2 Importy (1 pkt)

Rozszerz interpreter o mechanizm importów.

Interpreter po napotkaniu instrukcji zaimportowania innego pliku powinien przejść do jego wykonania, a następnie kontynuować dalszą pracę z aktualnym plikiem. Sposób realizacji tego rozszerzenia możesz dobrać według własnego uznania (np. wykorzystanie nieużywanej wartości punktowej czy słowo kluczowe).

W razie błędu – np. nieistniejący plik do zaimportowania, wypisz komunikat o błędzie i zakończ program z kodem wyjścia różnym od zera.

2.3 Kompilator (1 pkt)

Zaimplementuj kompilator języka *Beatnik* do dowolnego innego (np. Python, C++).

Przez kompilator rozumiemy program, który jako argument wywołania otrzyma nazwę pliku źródłowego (`sys.argv[1]`) oraz nazwę pliku do zapisu skompilowanego kodu (`sys.argv[2]`). W tym drugim ma znaleźć się kod w wybranym języku, który po wykonaniu da równoważny rezultat. Taki plik nie powinien już wczytywać w żaden sposób pliku źródłowego. Może korzystać jednak z innych zależności – przykładowo można wykorzystać ponownie fragmenty kodu interpretera. Nie zapomnij o wsparciu dla importów wprowadzonych w poprzednim podzadaniu.

W razie błędu – np. nieistniejący plik do zaimportowania, wypisz komunikat o błędzie i zakończ program z kodem wyjścia różnym od zera.

3 Obsługa

Celem tych zadań jest weryfikacja umiejętności i konfiguracji sprzętu, które przydadzą się na samym warsztacie.

3.1 Git (1 pkt)

Sklonuj następujące repozytorium z GitHuba: [aszady/bazel-www](https://github.com/aszady/bazel-www).

Wybierz jeden dowolny plik ze swojego rozwiązania (np. `Test`) i zamieść go w dowolnym podkatalogu. Wyślij go jako prośbę o pociągnięcie (ang. pull request). Odpowiedzią do tego zadania jest link do PR na GitHubie.

3.2 Bazel (1 pkt)

Zweryfikuj, czy masz poprawnie zainstalowane: Bazel, g++, Python.

W powyższym repozytorium uruchom polecenia:

```
bazel run //test1:a
```

```
bazel run //test1:b
```

Proste programy powinny się skompilować i uruchomić. Odpowiedzią do tego zadania jest wydruk z rezultatem wykonania tych poleceń.

4 Pytanie otwarte (0 pkt)

Opisz w wybranej przez siebie liczbie słów swoje dotychczasowe doświadczenia z budowania kodu w przeszłości – omów używane narzędzia, ich zalety i wady czy napotkane problemy. Jak powinien wyglądać idealny proces budowania?