

Zestaw zaliczeniowy

1. Warunki zaliczenia

Poprawne rozwiązanie wszystkich zadań podstawowych gwarantuje ocenę **4.0**. Samo podjęcie próby rozwiązania skutkuje oceną **3.0** lub **3.5**, zależnie od osiągniętego poziomu.

Rozwiązanie zadania projektowego jest opcjonalne. Jego wykonanie, w połączeniu z poprawnym rozwiązaniem zadań podstawowych, skutkuje oceną **4.5** lub **5.0**, zależnie od jakości wykonania.

Alternatywnym sposobem zaliczenia jest stworzenie dowolnej aplikacji w języku Python o wysokim stopniu skomplikowania, przekraczającym zakres przedmiotu. Taka aplikacja zwalnia od rozwiązywania zestawu zadań. Możliwość zaliczenia przedmiotu w ten sposób należy skonsultować z prowadzącym.

Odpowiedzi na zadania należy przesłać do końca semestru w postaci kodów lub repozytorium na adres mailowy pniedziela@wsei.edu.pl.

2. Zadania podstawowe

- 2.1. Napisz dekorator funkcji, który będzie logował informację o nazwie i typie wszystkich parametrów wejściowych funkcji dekorowanej w postaci:
`{<nazwa_parametru>: <typ_danych>, ...}`.
- 2.2. Napisz klasę, która będzie implementować generator kolejnych n potęg liczby a .
Użyj metod magicznych `__iter__()` i `__next__()`.
Liczby n i a powinny być parametrami wejściowymi generatora.
- 2.3. Za pomocą API zwracającego informacje dot. uniwersytetów w danym państwie:
`http://universities.hipolabs.com/search?country=<nazwa_kraju_eng>`
wyświetl nazwy uniwersytetów z 15 wybranych krajów w postaci:
`{<nazwa_kraju>: [<nazwa_uniwersytet1>, <nazwa_uniwersytet2>, ...], ...}`.
W celu przyspieszenia pobierania danych, wykorzystaj moduł `threading` do realizacji wielowątkowego pobierania informacji.
- 2.4. Napisz program, który zwraca pary bliźniaczych liczb pierwszych dla dużego zakresu.
Dwie liczby pierwsze są bliźniacze, jeśli ich różnica jest równa 2. Przykłady: 3 i 5, 5 i 7.
Wykorzystaj moduł `multiprocessing`, aby utworzyć kilka procesów, z których każdy będzie przetwarzał różne fragmenty zakresu.
- 2.5. Korzystając z danych XML:
`https://www.w3schools.com/xml/cd_catalog.xml`
zawierających kolekcję płyt CD, opracuj program, który wyświetli zestaw utworów i wykonawców w postaci listy: `[(<wykonawca1>, <tytuł1>), ...]`.

- 2.6. Napisz klasę *BankAccount*, która implementuje podstawowe operacje na koncie bankowym, takie jak wpłacanie, wypłacanie i sprawdzanie salda. Klasa powinna wywoływać wyjątek przy próbie wpłaty lub wypłaty niepoprawnej kwoty. Następnie napisz testy jednostkowe za pomocą *PyTest*, które sprawdzą poprawność działania metod oraz obsługę błędów.

3. Zadanie projektowe

Opracuj aplikację, która implementuje najbardziej podstawowe działanie platformy e-commerce. Projekt powinien obejmować proste funkcjonalności podzielone na kilka modułów i klas do zarządzania **zamówieniami**, **użytkownikami**, **koszykiem** oraz **produktami**.

Dodatkowo, aplikacja powinna:

- wykorzystywać paradygmat programowania obiektowego,
- być oparta o dowolną bazę danych (np. SQLite, MySQL) przechowującą dane z modułów,
- posiadać możliwość zwracania szczegółów zamówienia w postaci XML,
- zawierać kilka testów jednostkowych i co najmniej jeden test integracyjny pomiędzy modułami.

Kod powinien być zgodny z standardami formatowania (PEP-8) oraz zawierać plik *README*.