

Programowanie Obiektowe

dr inż. Piotr Bobiński (piotr.bobinski@warszawa.merito.pl)

Zadania na pierwsze ćwiczenia.

Zadania mają na celu ocenę stopnia wiedzy i umiejętności w dziedzinie programowania, znajomości języka programowania itp. w celu dostosowania poziomu zajęć.

Zadania z podstaw programowania

1. Pętle i operacje na liczbach

Zadanie: Napisz program, który pobiera liczbę n od użytkownika i wypisuje wszystkie liczby pierwsze mniejsze niż n .

Cel: Sprawdzenie znajomości pętli (for/while), instrukcji warunkowych (if), operacji modulo (%).

2. Funkcje i warunki

Zadanie: Napisz funkcję `is_palindrome(word)`, która sprawdza, czy podane słowo jest palindromem (np. "kajak", "anna"). Funkcja powinna ignorować wielkość liter.

Cel: Sprawdzenie umiejętności tworzenia funkcji, obsługi stringów, użycia warunków (if), zwracania wartości (return).

3. Listy i operacje na nich

Zadanie: Napisz program, który pobiera od użytkownika listę liczb (np. jako ciąg wpisany w jednej linii, rozdzielony spacjami), a następnie:

- Wypisuje sumę i średnią tych liczb.
- Znajduje największą i najmniejszą liczbę.
- Sortuje listę rosnąco i malejąco.

Cel: Sprawdzenie umiejętności pracy z listami, konwersji typów (int), użycia funkcji wbudowanych (`sum()`, `max()`, `min()`, `sorted()`).

4. Słowniki i obsługa danych

Zadanie: Napisz program, który symuluje prosty system ocen studenta. Program powinien:

1. Pozwolić użytkownikowi wprowadzić imię studenta i listę jego ocen.
2. Przechowywać dane w słowniku (dict).
3. Obliczyć i wyświetlić średnią ocen studenta.

Cel: Sprawdzenie znajomości słowników (dict), obsługi list, iteracji po strukturach danych.

5. Zagnieżdżone pętle – prosta grafika ASCII

Zadanie: Napisz program, który rysuje piramidę z gwiazdek o zadanej wysokości n, np. dla n = 5:

```
*  
  
***  
  
*****  
  
*****  
  
*****
```

Cel: Sprawdzenie znajomości pętli zagnieżdżonych, operacji na stringach, formatowania wyjścia.

6. Obsługa plików

Zadanie: Napisz program, który:

- Otwiera plik tekstowy,
- Wypisuje liczbę wierszy, słów i znaków w pliku,
- Zapisuje wynik do nowego pliku.

Cel: Sprawdzenie podstawowej obsługi plików (open(), read(), write()).

Zadanie z programowania strukturalnego (wprowadzające do programowania obiektowego)

Sprawdzenie wiedzy i umiejętności po kursie podstaw programowania oraz wprowadzenie w świat programowania obiektowego (OOP). Zadanie najpierw wymaga podejścia proceduralnego, a następnie jego refaktoryzacji na obiektowe (*to już będziemy robić wspólnie na zajęciach, ale próbować można*).

Cel ćwiczenia

1. **Ocena początkowych umiejętności** – praca z listami, pętlami, funkcjami, operacjami na danych.

2. **Uświadomienie zalet OOP** – po zakończeniu drugiej części porównanie obu wersji kodu. Dyskusja (*to już na kolejnym spotkaniu*):

- Jak zmieniła się czytelność i organizacja kodu?
 - Jakie korzyści przyniosło enkapsulowanie danych w klasach?
 - Jakie są zalety hierarchii obiektów w większych projektach?
 - Jak OOP pozwala łatwo rozszerzać funkcjonalność (np. dziedziczenie, polimorfizm)?
-

Zadanie: System zarządzania książkami w bibliotece

Część 1 – Podejście proceduralne

Napisz program w dowolnym języku, który umożliwia:

1. Dodanie nowej książki do listy (z tytułem, autorem, rokiem wydania).
2. Wyświetlenie listy wszystkich książek.
3. Wyszukiwanie książek po tytule.
4. Usunięcie książki z listy.

Dane książek można przechowywać w tablicach lub listach.

Część 2 – Podejście obiektowe

Przekształć kod w wersję obiektową. Kluczowe elementy:

- Utworzenie klasy Book, która przechowuje informacje o książce.
- Utworzenie klasy Library, która zarządza kolekcją książek i operacjami na nich.
- Metody klasy Library powinny obsługiwać dodawanie, wyświetlanie, wyszukiwanie i usuwanie książek.