

C++: Klasy cz. 2

Zajęcia laboratoryjne nr 2 - Metodyka i Techniki Programowania II

Autorzy instrukcji: dr inż. Jarosław Bułat, mgr inż. Artur Kos

Wykład do laboratorium: PI_09.pdf

Zad 1.1 Prosta klasa

Klasa Circle jest zdefiniowana jako:

```
class Circle {  
public:  
    float radius_;  
}
```

- 1.W funkcji `main()` utwórz dwa obiekty tej klasy.
- 2.Zainicjalizuj ich składową `radius_` wybranymi liczbami.

Zad 1.2 Konstruktor

Rozszerz klasę z Zad1.1 dodając konstruktor:

- 1.Zadeklaruj i zdefiniuj konstruktor z 1 parametrem reprezentującym promień okręgu, który powinien być przypisywany do składowej `radius_`.
- 2.W funkcji `main()` utwórz nowy obiekt używając zdefiniowanego konstruktora i wybranej liczby (przekazywanej jako parametr). Dlaczego wystąpił błąd?
- 3.Rozwiąż problem dodając konstruktor bezparametryczny.

Zad 1.3 Destruktor

Rozszerz klasę zdefiniowaną w Zad 1.1 o destruktora (nie bazuj na programie utworzonym w punkcie Zad 1.2):

- 1.Zmień typ elementu `radius_` z `float` na `float *`.
- 2.Dodaj konstruktor, który powinien alokować pamięć i przypisywać ją do `radius_`.
- 3.Dodaj destruktora.

4.W funkcji `main()` zademonstruj poprawność działania konstruktora i destruktora.

Zad 1.4 Metody

Rozszerz klasę zdefiniowaną w Zad 1.1 (nie bazuj na programie utworzonym w poprzednich punktach) metodami:

- 1.Zadeklaruj i zdefiniuj dwie metody bezparametrowe: `circumference()` i `area()` obliczające odpowiednio: obwód i powierzchnię koła.
- 2.Obie metody powinny wykorzystywać do obliczeń element `radius_`.
- 3.Metody powinny obliczać i zwracać wynik i niczego nie wyświetlać.
- 4.Zdefiniuj obie metody poza deklaracją klasy.
- 5.W funkcji `main()` zademonstruj poprawność działania obu metod.

Zad 1.5 Inicjalizacja składowych prywatnych

Klasa `Circle` jest zdefiniowana jako:

```
#include <iostream>
using namespace std;

class Circle{
private:
    float radius_;
    float area_;
    float circumference_;

public:
    Circle(float);
};
```

- 1.Dodaj definicję konstruktora inicjalizującego składowe `radius_` i `area_`.
- 2.W funkcji `main()` utwórz nowy obiekt za pomocą konstruktora.
- 3.Zastąp definicję konstruktora nową definicją z listą inicjalizującą.
- 4.W funkcji `main()` utwórz nowy obiekt za pomocą konstruktora z listą inicjalizującą.

Zad 1.6 setter i getter

- 1.W oparciu o program utworzony w punkcie Zad1.5. rozszerz klasę `Circle` o:

- setter modyfikujący element `radius_`
 - getter zwracający wartość elementu `radius_`
- 2.Zademonstruj działanie setter-a i getter-a w funkcji `main()`.

UWAGA!

****ZACHOWAJ KODY PROGRAMÓW NA NASTĘPNE ZAJĘCIA LABORATORYJNE ****

Literatura:

- 1.Wykłady z sem. 1.
- 2.Dowolna pozycja literaturowa z zakresu.