

Programowanie obiektowe

Laboratorium 3. Definiowanie klas, metody klasy `Object`

Tematyka

Zajęcia dotyczą projektowania prostej klasy, reprezentującej pewien rzeczywisty obiekt, oraz zapoznania się z metodami klasy `Object`: `toString()`, `equals()` i `hashCode()`.

Zadania

Zadanie 1.

Zaprojektuj i zaimplementuj klasę reprezentującą wybrany obiekt. (np. z listy poniżej zadań).

Klasa ma zawierać co najmniej trzy pola (np. typu `int`, `double`, `String`, `boolean`), charakteryzujące właściwości obiektu, np. dla klasy `Pudełko` mogłyby to być:

```
/** Wymiary pudełka, kolejno: szerokość, wysokość, głębokość. */
int[] wymiary = ...;
/** Materiał, z którego wykonano pudełko. */
String material = ...;
/** Flaga określająca, czy pudełko jest zamykane. */
boolean zamykane = ...;
```

Pola zadeklaruj jako prywatne, a w klasie dodaj metody ustawiające i zwracające ich wartość:

`set<<ZMIENNA>>()`, `get<<ZMIENNA>>()` - dla każdego pola oddzielnie;
metody `set...()` przyjmują jako parametr nową wartość dla pola i nie zwracają nic,
metody `get...()` nie mają żadnych parametrów wywołania i zwracają aktualną wartość pola.

Napisz prosty program (funkcja `main`) tworzący obiekt tej klasy, ustawiający wartości jego pól (za pomocą metod `set...()`), a następnie je odczytujący (za pomocą metod `get...()`) i wypisujący je na ekran.

Zadanie 2 – opcjonalnie (bo nie było na wykładzie), ale zachęcam do wykonania

W klasie z zadania 1 zaimplementuj metody dziedziczone z klasy `Object`: `toString()`, `equals()` i `hashCode()`, zgodnie z ich definicją i przeznaczeniem.

W programie testowym sprawdź działanie powyższych metod: utwórz dwa nowe obiekty, z których jeden będzie miał takie same wartości pól co obiekt z zadania 1, a drugi będzie miał inne; sprawdź wynik zwracany przez metody `eqals()` i `hashCode()`, wykorzystaj metodę `toString()` do wypisania zawartości informacyjnej obiektów na ekran

(wskazówka: `System.out.println(obj)` – metoda `toString()` zostanie automatycznie wywołana dla obiektu `obj`).

Zadanie 3.

Dodaj do klasy z zadania 1 odpowiedni komentarz dokumentujący: dla klasy, pól i metod.

Wykorzystaj odpowiednio metasymbole `@param` i `@return`.

Wygeneruj dokumentację dla klasy.

Lista przykładowych klas do implementacji

radio

komputer

laptop

monitor

tablet

smartfon

telewizor

samochód

... inne mile widziane