

12. Klasy i pakiety

1 Klasy

W rozwiązaniu poniższych zadań można posłużyć się materiałami z wykładów lub oficjalnym tutorialiem języka Javy: <http://docs.oracle.com/javase/tutorial/java/java00/index.html>

Zadanie 01 Stwórz klasę Student. W funkcji main (należącej do tej samej klasy) utwórz nowy obiekt.

Zadanie 02 Do klasy Student dodaj prywatne atrybuty firstName i lastName typu String oraz publiczną metodę zwracającą pełne nazwisko studenta getLastname().

Zadanie 03 Do klasy Student dodaj konstruktor dwuargumentowy Student(String firstName, String lastName) ustawiający odpowiednie wartości atrybutów. Wykorzystaj słowo kluczowe this.

Czy możliwe jest teraz utworzenie obiektu klasy Student jak w zadaniu 1? Zaktualizuj kod w głównej funkcji.

2 Tablice

Zadanie 04 Do klasy Student dodaj prywatny atrybut int[] marks, w którym będą przechowywane wszystkie otrzymane przez studenta oceny. Dodaj również atrybut przechowujący aktualny rozmiar tablicy. Jej maksymalny rozmiar ustal dowolnie.

Dodaj metodę addMark(int) pozwalającą dopisać ocenę. Utworzonemu obiektowi studenta dodaj kilka ocen.

Zadanie 05 Do klasy dopisz konstruktor trójargumentowy, który jako trzeci argument będzie przyjmował tablicę ocen. Do inicjalizacji dwóch pierwszych atrybutów wykorzystaj konstruktor dwuargumentowy poprzez słowo kluczowe this, np.:

```
1 public Foo(int bar) {  
2     this.bar = bar;  
3     // ...pewien kod niezbedny do inicjalizacji obiektu  
4 }  
5  
6 public Foo(int bar, String baz) {  
7     this(bar); // wywołanie konstruktora jednoargumentowego  
8     this.baz = baz;  
9 }
```

Zadanie 06 Napisz funkcję `int[] getMarks()` zwracającą wszystkie otrzymane przez studenta oceny. Pamiętaj, że należy zwrócić kopię tablicy, np. wykorzystując metodę `copyOf(int[], int)` z klasy `Arrays`.

Czy w przypadku tablicy przekazywanej jako trzeci argument w konstruktorze trójargumentowym wystarczy jej zwykłe przypisanie czy należało również ją skopiować?

3 Składowe statyczne

Zadanie 07 Do klasy `Student` dodaj prywatny statyczny atrybut `id`, w którym przechowywana będzie liczba wszystkich utworzonych obiektów klasy `Student`. W odpowiednim konstruktorze dopisz kod zwiększający ten atrybut (wystarczy w jednym — którym?).

Zadanie 08 Do klasy `Student` dopisz statyczną metodę zwracającą liczbę wszystkich utworzonych obiektów klasy `Student`. Na końcu funkcji `main` napisz kod wypisujący tę wartość.

Zadanie 09 Przenieś inicjalizację statycznego atrybutu `id` do bloku `static`.

4 Metody `toString()` oraz `equals()`

Zadanie 10 W funkcji `main` utwórz tablicę obiektów klasy `Student`. Korzystając z rozszerzonej pętli `for` (nie jest to wymagane do poprawnej realizacji zadania) wypisz obiekt każdego studenta na standardowe wyjście. Co zostało wypisane?

Rozszerzona pętla `for`:

```
1 String[] array = { "foo", "bar", "baz" };
2
3 for (String elem: array) {
4     System.out.println(elem); // elem będzie przyjmował kolejno wartości
5                               // 'foo', 'bar' i 'baz'
6 }
```

Zadanie 11 W klasie `Student` przeciąż metodę `String toString()`, która zwróci informacje o studencie zawierające jego imię, nazwisko oraz oceny, np.:

```
1 Jan Kowalski: 5, 3, 2,
```

W jakiej postaci obiekty klasy `Student` są teraz wypisywane?

Zadanie 12 Utwórz studenta z identycznym imieniem, nazwiskiem oraz ocenami jak jeden z wcześniej utworzonych obiektów. Porównaj ich wykorzystując operator porównania `==` oraz metodę `equals()`. np.:

```
1 System.out.println(s1 == s2);  
2 System.out.println(s1.equals(s2));
```

Zadanie 13 Cemu służy metoda `equals` i czym się różni od operatora `==`? Zdefiniuj metodę `equals`, aby zwracała prawdę, jeżeli studenci mają te same imię i nazwisko. Przetestuj odpowiednim fragmentem kodu.

5 Pakiety

Zadanie 14 Jaką funkcję pełni słowo kluczowe `public` przed deklaracją klasy? Kiedy należy je stosować?

Uwaga: poniższe zadania sugeruję wykonać poza IDE.

Zadanie 15* Wydziel klasę `Student` do pakietu `University` (instrukcja `package`). Stwórz oddzielną klasę aplikacji w pakiecie `app`, w której umieścisz wcześniejszą funkcję `main`. Do drugiego z tych plików będziesz musiał zaimportować klasę `Student` z pakietu `university` poprzez słowo kluczowe `import`. Pamiętaj o odpowiedniej hierarchii katalogów, np.:

```
1 .  
2 - university  
3   - app  
4     - UniversityApp.java  
5   - university  
6     - Student.java
```

Zadanie 16* Załóżmy, że mamy klasę `Group` reprezentującą grupę zajęciową. Klasa ta będzie używana razem z klasą `Student` również poza pakietem. W jakim miejscu i/lub pliku należy ją umieścić oraz z jakimi modyfikatorami dostępu?

Zadanie 17* Załóżmy, że mamy klasę `Locker`, która reprezentuje prywatną skrzynkę studenta. Klasa ta nigdy nie będzie używana poza pakietem `university`. W jakich miejscach i/lub plikach możemy ją umieścić? Z jakimi modyfikatorami dostępu?