

## 16. Klasy generyczne, kontenery

### 1 Klasy generyczne

Klasy generyczne w języku Java udostępniają programiście możliwość programowania uogólnionego, podobnie jak szablony w C++:

```
1 class Container<T> {  
2     private T elem;  
3  
4     public void set(T e) { this.elem = e; }  
5     public T get() { return this.elem; }  
6 }
```

Typy dla klasy szablonowej określa się podczas tworzenia obiektu:

```
1 Container<Integer> iCont;  
2 iCont.set(123);  
3 int x = iCont.get();
```

**Zadanie 01** Zmodyfikuj definicję klasy `Box` w taki sposób, aby mogła przechowywać w składowej `elements` dane dowolnego typu.

Wskazówka: inicjalizacja tablicy przechowującej typ generyczny może wyglądać następująco:  
`elements = (T[])new Object[size];`

**Zadanie 02** W funkcji `main` napisz fragment kodu demonstrujący użycie klasy `Box` przechowującej elementy co najmniej trzech typów: liczbowego, łańcuchowego oraz własnej przykładowej klasy.

### 2 Kontenery

Jednym z najczęściej wykorzystywanych kontenerów w języku Java są kontenery implementujące interfejs `List<E>`, np. klasa `ArrayList<E>`:

```
1 import java.util.List;  
2 import java.util.ArrayList;  
3  
4 List<int> list = new ArrayList<int>();  
5  
6 list.add(5);  
7 list.add(10);  
8  
9 list.size();  
10 list.get(0)
```

Tablice asocjacyjne w języku Java reprezentowane są przez klasy `HashMap` oraz `TreeMap` implementujące interfejs `Map`.

**Zadanie 03** Wykorzystaj listę `List`, która zastąpi tablicę przechowującą elementy pudełka `Box`. Jakie metody udostępnia interfejs `List`?

Wymagana będzie zmiana kilku metod i być może implementacja metody odpowiedzialnej za konwersję listy do łańcucha znaków.

**Zadanie 04** Do klasy `Box` dodaj metodę `countElements`, która zwróci tablicę asocjacyjną informującą o tym, jakie elementy i ile razy są zawarte w pudełku. Kluczami tablicy asocjacyjnej powinny być obiekty klasy odpowiadającej klasie przechowywanym elementom, a wartościami dodanie liczby całkowite.

Dodaj fragment kodu demonstrujący użycie powyższej metody z wypisaniem tablicy asocjacyjnej na ekran.

### 3 Bloki inicjalizacyjne

Bloki inicjalizacyjne (instancyjne i statyczne) umożliwiają wczesne wykonanie bloku kodu, w szczególności inicjalizacji zmiennych.

```
1 static int foo;
2 static {
3     foo = 0;
4 }
5
6 int bar
7 {
8     bar = 0;
9 }
```

**Zadanie 05** Przenieś kod inicjalizujący zmienną `name` oraz zmienną statyczną `boxes` do bloków inicjalizacyjnych. Kiedy dokładnie względem konstruktorów oraz klas bazowych wywoływane są bloki inicjalizacyjne?