

MAS – wybrane konstrukcje obiektowych języków programowania 1

1. Interfejsy

Napisz program porównujący paczki pod względem ich zapelnienia, tj. stosunku wagi do objętości:

- a) Każda paczka jest pudełkiem o danej szerokości, długości i wysokości w centymetrach.
- b) Każda paczka ma daną wagę w gramach. Z tych atrybutów będzie można wyliczyć zapelnienie (zwróć uwagę na typy przy dzieleniu, wykorzystaj rzutowanie).
- c) Stwórz klasę *Paczka* implementującą interfejs *java.lang.Comparable* z możliwością porównania zapelnienia paczek (przeczytaj dokumentację na temat tego interfejsu).
- d) Stwórz demo działania programu:
 - stwórz przykładowe paczki
 - porównaj dwie paczki za pomocą metody *compareTo*
 - wrzuc parę paczek do listy (*ArrayList*), posortuj ją za pomocą gotowej metody *sort* z klasy *java.util.Collections*, pokaż wynik sortowania, znajdź najmniej zapelnioną paczkę za pomocą odpowiedniej gotowej metody również z klasy *java.util.Collections*
- e) Stwórz własny interfejs o nazwie *Stackable* z metodą *stack* przyjmującą jako parametr obiekt klasy paczka i zwracającą typ *boolean*.
- f) Zaimplementuj stworzony interfejs *Stackable* w klasie paczka, w ten sposób, że metoda *stack* zwraca:
 - *true* jeśli paczka z parametru jest mniejsza niż dana paczka (tzn. nie jest szersza i jednocześnie nie jest dłuższa)
 - *false* w przeciwnym wypadku
- g) Stwórz klasę *Worek*, który posiada wagę, średnicę i wysokość (jest w kształcie walca). Zaimplementuj oba interfejsy również dla niego.

2. Wyjątki

Napisz program obsługujący konto bankowe:

- a) Konto bankowe posiada numer i saldo oraz metodę pozwalającą na wyciągnięcie środków z konta.
- b) Metoda ta rzuca wyjątek jeśli wyciągnięcie środków spowodowałoby debet na koncie.
- c) Stwórz odpowiedni wyjątek rozszerzający klasę *java.lang.Exception* oraz informujący ile zabrakło środków za pomocą specjalnej metody (wyliczoną brakującą kwotę wyjątek dostaje przez konstruktor).
- d) Przetestuj działanie programu: łapanie wyjątku i wyświetlanie brakującej kwoty gdy wyciągane środki przekraczają saldo.
- e) Dodaj obsługę nadawania numeru konta: numer jest przekazywany jako String w formacie: AAAAAAAC

- C to suma kontrolna (1 cyfra),
- AAAAAAA to cyfry numeru konta,

Możliwe wyjątki:

- zła liczba cyfr (można użyć *java.lang.NumberFormatException*)
- niedopuszczalne znaki (można użyć *java.lang.NumberFormatException*)
- nieprawidłowa suma kontrolna (specjalny wyjątek)
- w bloku *finally* ustaw saldo na 0 (zostanie ustawione nawet gdy był wyjątek)

Aby wyliczyć sumę kontrolną należy:

1. Przemnożyć kolejne cyfry przez odpowiednie wagi: [7 1 3 9 7 11 3]
2. Zsumować wyniki mnożenia.
3. Podzielić przez modulo 10 (= reszta z dzielenia, operator %).
4. Otrzymany wynik jest sumą kontrolną.

Przykład:

Konto: 10103421

Wagi: 7 1 3 9 7 11 3

Konto: 1 0 1 0 3 4 2

Wynik: $7*1 + 1*0 + 3*1 + 9*0 + 7*3 + 11*4 + 3*2 = 81$

$81 \bmod 10 = 1$ (cyfra kontrolna 1)