

Programowanie Java
Anna Gogolińska
Zestaw 4

Zad 1. Stwórz klasę abstrakcyjną *Pojazd* z polami *miejsca*, *kola*, *cena*, *waga* zaimplementowanymi metodami *setter* i *getter* dla tych pól oraz abstrakcyjnymi metodami *czyToWieloslada()*, *czyMaDach()*, *czyUtonie()*, *wypisz()*. Stwórz klasy dziedziczące z klasy *Pojazd*: *Samochod*, *Rower*, *Lodz* implementujące wszystkie abstrakcyjne metody. W klasie z metodą *main()* stwórz parę obiektów i przetestuj ich działanie.

Zad 2. Utwórz interfejs *Srednia* z metodą *obliczSrednia(-)* przyjmującą jako argument tablicę liczb, a zwracającą jedną liczbę. Następnie stwórz klasy implementujące interfejs *Srednia*:

- *SredniaArytmetyczna()*,
- *SredniaGeometryczna()*,
- *SredniaHarmoniczna()*.

Zad 3. Zmodyfikuj klasę *Pojazd* poprzez implementację interfejsu *Comparable*. Stwórz tablicę zawierającą elementy typu *Samochod*, *Rower*, *Lodz* - przynajmniej kilka i posortuj je względem ilości miejsc za pomocą metody *Arrays.sort(-)*.

Zadania samodzielne

Zad 4.

- Stwórz interfejs *Figura* z dwoma metodami, zwracającymi pole i obwód (wynik typu *double*).
- Stwórz trzy klasy implementujące interfejs *Figura*: *Kwadrat*, *Prostokat* i *Trojkat*.
 - Stwórz pola odpowiednie dla figur w każdej z klas.
 - Dla *Trojkat* stwórz pola *a*, *b*, *c* oraz *h* (gdzie *h* może być *null*, zakładamy że opada na bok *a*).
 - Dodaj konstruktory, metody *setter* i *getter* dla pól.
 - Zaimplementuj metody *pole()* i *obwod()* (w zależności od rodzaju figury).
 - Dla *Trojkat* sprawdzaj czy podana jest wysokość. Jeśli nie użyj wzoru Herona.
- Stwórz klasę *Obszar*, która będzie posiadać:
 - listę obiektów *Figura* (utworzyć pole: *ArrayList<Figura> list = new ArrayList<Figura>()*)
 - metodę *dodajFigura(Figura f)* – metoda ta ma dodawać figury do listy
 - metodę *poleObszaru()* zwracającą łączne pole wszystkich figur na liście.
- W *main()* utworzyć obiekt klasy *Obszar* i dodać do niego losowe figury – losować należy typ figur (np. jeśli wylosowano 0 to dodajemy *Kwadrat*, jeśli 1 to *Trojkat* itd.) jak również wartości boków (przynajmniej dla *Kwadrat* i

Prostokat, dla *Trojkat* można „na sztywno” wpisać parę możliwych wartości z nich losować). Wypisać jego pole.

Zad 5

Utworzyć klasę abstrakcyjną *PrzegladanieGrafu* realizującą algorytm przeglądania grafu. Do przechowywania kolejnych do odwiedzenia wierzchołków wykorzystać klasę *ArrayList*. Jako metody abstrakcyjne utworzyć metody *wstaw(-)* oraz *pobierz()* numer wierzchołka. Prosty graf zakodować w postaci macierzy sąsiedztwa (np. podawanej w konstruktorze klasy), a następnie utworzyć klasy DFS oraz BFS realizujące algorytmy przeglądania grafu w głąb oraz wszerz.

Pobierz dane ze standardowego wejścia:

- linia 1: lista N wierzchołków grafu oddzielona spacjami (każdy wierzchołek to litera
- linie 2-N+1: listy sąsiedztwa oddzielone spacjami (wierzchołek lista sąsiadów)
- linia N+2: wierzchołek startowy
- linia N+3: algorytm (DFS lub BFS)

Wyświetl wierzchołki w kolejności odwiedzin.

Przykład

Wejście:

A B C D E

A B

B A C D

C B E

D B

E C

A

BFS

Wyjście:

A B C D E