

Laboratorium z przedmiotu Programowanie Obiektowe — lista 8 Programowanie w języku Java

Piotr Lechowicz

1 Cel zajęć

Dziedziczenie w języku Java.

2 Zadania

1. Stworzenie abstrakcyjnej klasy `Kształt` posiadającej dwa pola z dostępem chronionym:

```
int x;  
int y;
```

Pola określają położenie obiektu.
2. Dodanie getterów i setterów do pól określających położenie obiektu
3. Stworzenie konstruktora dwuargumentowego — argumentami powinno być położenie obiektu.

1 punkt

4. Stworzenie klas `Koło` i `Prostokąt` dziedziczące po klasie `Kształt`
5. Dodanie do klasy `Koło` zmiennej określającej średnicę koła.
6. Stworzenie konstruktora w klasie `Koło` przyjmującego 3 argumenty — położenie i średnica koła

2 punkty

7. Dodanie do klasy `Prostokąt` zmiennych określających długości boków.
8. Stworzenie konstruktora w klasie `Prostokąt` przyjmującego 4 argumenty — położenie i długości boków.

3 punkty

9. Stworzenie abstrakcyjnej metody `void rysuj()` w klasie `Kształt` służącej do rysowania obiektu.
10. Przeciążenie metody `rysuj()` w klasach pochodnych. (Metoda może wypisywać w konsoli tekst informujący o tym jaka to figura i jakie jest jej położenie)
11. Stworzenie w metodzie `main` parę obiektów (zmienne rzutowane na klasę bazową) i przetestowanie działania metody.

4 punkty

12. Stworzenie abstrakcyjnej metody w klasie bazowej `double getPolePowierzchni()` zwracającej pole powierzchni i przeciążenie jej w klasach pochodnych. Przetestowanie metody w klasie `main`.

W przypadku korzystania z kodu tworzącego GUI, ten podpunkt może zostać pominięty.

5 punktów

3 Zadania dodatkowe

W celu sprawdzenia działania programu można skorzystać z gotowego kodu tworzącego okno aplikacji. W tym celu należy pobrać pliki z moodle'a i rozpakować na dysku. Następnie należy zaimportować projekt w programie Eclipse.

- Importowanie istniejącego projektu w Eclipse – *File / New / Java Project* → *Project name:* — należy podać nazwę projektu → należy odznaczyć *"use default location"* i podać ścieżkę do folderu → *Next / Finish*
(Jeżeli folder do zaimportowania znajduje się w folderze *workspace* należy nie odznaczać *"use default location"*)

Aplikację należy uruchomić z klasy `laboratory.gui.FXApplication.java`. W celu kompatybilności z dostarczonym kodem należy:

- Klasę `Kształt` nazwać `MyShape`
- Metoda `draw()` w klasie `MyShape` musi mieć następującą sygnaturę:

```
1 import javafx.scene.canvas.GraphicsContext;  
2 ...  
3 void draw(GraphicsContext gc) {  
4 }
```

Kontekst graficzny (`GraphicsContext`) jest klasą umożliwiającą wywołanie funkcji rysujących na płótnie (`Canvas`).

Kolor ustawia się za pomocą:

```
1 import javafx.scene.paint.Color;
2
3 // metoda setFill klasy GraphicsContext
4 public GraphicsContext::setFill(Color)
5
6 // konstruktor koloru
7 public Color(double red, double green, double blue, double
    opacity)
```

Kształty rysuje się za pomocą:

```
1 public GraphicsContext::fillOval(double x, double y, double
    width, double height)
2
3 public GraphicsContext::fillRect(double x, double y, double
    width, double height)
```

- Odkomentować w klasie `ShapeCanvas` linię
`shape.draw(canvas.getGraphicsContext2D());`
- Uzupełnić w klasie `ShapeFactory` metodę
`MyShape generateShape()`,
która losowo generuje kształty do narysowania.