

**Pracownia Programowania
Obiektowego**

Lista nr 7 - 15.04.2018

Język programowania: Java

OS: Windows 10 Enterprise 64-bit

IDE i kompilator: IntelliJ IDEA

Opis implementacji zadań z listy 7

Mateusz Zając, 298654

15.04.2018

1 Podstawy implementacji klas

Przed przystąpieniem do realizacji zadań z listy musimy utworzyć odpowiednią hierarchię klas. Naszą klasą abstrakcyjną, po której dziedziczą obie podklasy jest **Figura**. Wspólną częścią podklas **Okrag** oraz **Trojkat** są pola:

- **String name** - nazwa figury
- **String colour** - kolor figury

Pozostałe pola:

- **double radius** (pole klasy **Okrag**) - promień okręgu
- **double base** (pole klasy **Trojkat**) - podstawa trójkąta
- **double height** (pole klasy **Trojkat**) - wysokość trójkąta

Z tak utworzoną hierarchią możemy przystąpić do właściwej części listy zadań.

2 Pierwsza część

Pierwsza część polega na implementacji interfejsu do edycji obiektów. Dzięki bibliotece **Swing** mamy możliwość wyświetlenia na ekranie okna dialogowego z odpowiednimi polami tekstowymi i przyciskami, które są niezbędne do edycji pól obiektu. Wymagany z zadaniu interfejs jest realizowany przy pomocy metody **guiFig** odpowiednio klas **Okrag** i **Trojkat**. Na początku metoda wywołuje **loadObject**, opisaną w drugiej części zadania. Wczytujemy z pliku obiekt do edycji. Kolejne linijki metody mają na celu zdefiniowanie pól edycji, ich etykiet oraz przycisku potwierdzającego zapis wpisanych danych do pól obiektu. Do obiektu **but1** (przycisku „zapisz”) dopisujemy „słuchacza”. Dzięki niemu kiedy tylko użytkownik naciśnie przycisk, zostanie wywołane parsowanie zawartości pól i ich zapis (za pomocą metody **saveObject**). Argument przekazywany do metody **guiFig** jest używany do określenia jaki plik ma czytać program i do jakiego pliku po edycji ma zapisywać nowe wartości pól.

3 Druga część

Druga część zadania to implementacja interfejsu ***Serializable*** razem z metodami służącymi do zapisu oraz odczytu obiektów z pliku:

- ***loadObject(String s)*** - Czyta plik o nazwie podanej z argumentu. Jeżeli podany plik nie istnieje, inicjowany jest nowy obiekt z domyślnymi wartościami pól.
- ***saveObject(String s)*** - Zapisuje obiekt do pliku o nazwie podanej w argumentu.

4 Uruchomienie programu

W celu uruchomienia programu musimy poza wywołaniem pliku wykonalnego programu z linii poleceń podać także nazwę szukanego pliku z zapisanym obiektem (bez rozszerzenia) oraz nazwę klasy obiektu, który jest zawarty w pliku.