

PAU 2024

Lab. 1

1. Cel: Prosty kalkulator pól i obwodów figur dwu- i trójwymiarowych;
2. Przygotuj interfejs *Printable* z metodą `void print()`;
3. Przygotuj klasę abstrakcyjną *Figure* zawierającą, abstrakcyjne metody:
 - `double calculateArea()`;
 - `double calculatePerimeter()`;
4. Zaimplementuj klasy *Triangle*, *Square*, *Circle* dziedziczące po klasie *Figure* i implementujące interfejs *Printable*:
 - a. Do klas można dopisać dowolne metody i pola pomocnicze;
 - b. Każda klasa powinna mieć konstruktor z parametrami typowymi dla danej figury;
 - c. Metoda *print* z interfejsu powinna wypisywać na ekran informacje o obiekcie (dane charakterystyczne dla figury);
 - d. W przypadku podania niepoprawnych danych podczas konstrukcji obiektu - wyjątek powinien zostać rzucony;
5. Zaimplementuj klasę *Prism* (oznaczającą graniastosłup) o dowolnej podstawie typu ***Figure*** (*użytkownik wybiera trójkąt, prostokąt lub koło*). Oblicz jego pole powierzchni oraz objętość (dodaj odpowiednie pola i metody). Wykorzystaj w tym celu **abstrakcję**;
6. Stwórz prosty konsolowy interfejs użytkownika (oparty o `while` i `switch`) umożliwiający:
 - a. wybór figury;
 - b. podanie danych;
 - c. wyświetlenie;
 - d. wyjście z programu;
7. Architektura programu powinna w maksymalny sposób odgraniczyć część interfejsu użytkownika (wprowadzania/wypisywanie danych) od logiki programu, tj. dane wprowadzaj w oddzielnej klasie a implementację logiki pozostaw odseparowaną od wprowadzania danych.

Wskazówki:

1. ALT+INSERT pozwala włączyć menu generowania kodu (IntelliJ IDEA)
2. Klasa abstrakcyjna z abstrakcyjną metodą:

```
public abstract class MyAbstractClass {  
    abstract void myMethod();  
}
```

3. Dziedziczenie i implementacja interfejsu:

```
public class MyExtendedClass extends MyBaseClass  
    implements MyInterface {  
    // implementacja  
}
```

4. Wypisywanie na ekran:

```
System.out.println("Hello World"); // zakończone znakiem nowej linii  
System.out.print("Hello World");
```

```
System.out.println(String.format(Locale.US, "%s %d %.2f", myString,  
myInt, myDouble));
```

5. Pobieranie danych z klawiatury:

```
Scanner scanner = new Scanner(System.in);  
int myInt = scanner.nextInt();  
double myDouble = scanner.nextDouble();  
String myString = scanner.nextLine();
```

Teoria:

1. JDK, JRE, JVM – co oznaczają te skróty?
2. Czym się różni program napisany w C++ od programu w JAVA pod względem kompilacji (kompilacja do kodu bajtowego vs kompilacja do kodu maszynowego, zalety i wady);
3. Obiektość w języku JAVA – pakiety, budowa klasy, definicja konstruktora, metody klas, dostępność składowych, metoda *main*, przeładowanie nazw metod, domyślne argumenty metod;
4. Typy prymitywów w języku JAVA i ich ogólna charakterystyka;
5. Interfejs (jako konstrukcja programistyczna);
6. Klasa abstrakcyjna;
7. Konwencja nazewnictwa klas, zmiennych i metod. Na zajęciach posługujemy się konwencją zgodną z GJS: <https://google.github.io/styleguide/javaguide.html>

Konwencja OBOWIĄZUJE we wszystkich programach (również na kolejnych zajęciach).

Po uzyskaniu zaliczenia na zajęciach, prześlij źródła w archiwum **zgodnie z konwencją nazewnictwa**.