

ANS Elbląg
Instytut Informatyki Stosowanej im. Krzysztofa
Brzeskiego
Programowanie obiektowe I – laboratorium

Studium Stacjonarne, sem. 3, 2022/2023

Sprawozdanie nr : 10,
nr grupy: 1,
dzień: wtorek,
godz. 12:00.

Data wykonania ćwiczenia: 10.01

Data oddania sprawozdania: 30.01

Nazwisko i imię: Kuczawski Kacper

Nr albumu: 20195

Nazwa pliku : lab10_kuczawski_kacper20195

1. W klasie publicznej **Lab10z1** utworzyć interfejs **Obliczenia** zawierający metody obliczania pola trójkąta, kwadratu i trapezu. Metody te to, odpowiednio, **poleTrojkata**, **poleKwadratu**, **poleProstokata** i **poleTapezu**. Następnie stworzyć klasę adapterową **Adapt** implementującą te metody jako puste oraz klasy pochodne **Trojkat**, **Kwadrat**, **Prostokat** i **Trapez** zawierające implementacje poszczególnych metod. W metodzie **main()** klasy **Lab10z1** utworzyć obiekty poszczególnych klas i wywołać dla nich odpowiednie metody. Wydrukować wyniki.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.text.*;
public class temp{
    public static void main(String args[]){

        Trojkat t = new Trojkat();
        System.out.println("Pole Trójkąta: "+t.Ptk(2,4));
        Kwadrat k = new Kwadrat();
        System.out.println("Pole Kwadratu: "+k.Pkw(5));
        Prostokat p = new Prostokat();
        System.out.println("Pole Prostokąta: "+p.Ppk(3,7));
        Trapez tr = new Trapez();
        System.out.println("Pole Trapezu: "+tr.Ptp(3, 5, 7));
    }
}

interface Obliczenia{

    double Ptk(double a, double h);
    double Pkw(double a);
    double Ppk(double a, double b);
    double Ptp(double a, double b, double h);
}

class Adapt implements Obliczenia{

    public double Ptk(double a, double h){return 0;};
    public double Pkw(double a){return 0;};
    public double Ppk(double a, double b){return 0;};
    public double Ptp(double a, double b, double h){return 0;};
}

class Trojkat extends Adapt{

    public double Ptk(double a, double h){
        return (a*h)/2;
    }
}
```

```

class Kwadrat extends Adapt{

    public double Pkw(double a){
        return a*a;
    }
}

class Prostokat extends Adapt{

    public double Ppk(double a, double b){
        return a*b;
    }
}

class Trapez extends Adapt{

    public double Ptp(double a, double b, double h){
        return ((a+b)*h)/2;
    }
}

```

Wyniki:

```

Pole Trójkąta: 4.0
Pole Kwadratu: 25.0
Pole Prostokąta: 21.0
Pole Trapezu: 28.0

Process finished with exit code 0

```

2. Napisać klasę abstrakcyjną **Bankier** zawierającą następujące elementy: pole **imie**, konstruktor wypełniający to pole, metodę abstrakcyjną **licz** o dwóch argumentach typu **double** zwracającą wartość typu **double**. Następnie utworzyć klasę pochodną **UczciwyBankier**: dziedziczącą po klasie **Bankier**, zawierającą implementację metody **licz**, która zwraca sumę swoich argumentów oraz konstruktor wywołujący konstruktor klasy bazowej (nadrzędnej). W kolejnym kroku utworzyć klasę pochodną **PodstępnyBankier**: dziedziczącą po klasie **Bankier**, zawierającą implementację metody **licz**, która zwraca sumę swoich argumentów, pomniejszoną o 20% oraz konstruktor wywołujący konstruktor klasy bazowej. W klasie publicznej **Lab10z2** utworzyć obiekt klasy **Bankier** oraz obiekt klasy **PodstępnyBankier**, i dać im do wykonania te same obliczenia oraz sprawdzić, jakie otrzymali wyniki (odpowiedni komunikat z imieniem bankiera). Wydrukować wyniki.

Treść:

Wyniki:

3. Załóżmy, że w bibliotece oprócz książek, można wypożyczać także czasopisma i że każdy czytelnik może mieć wypożyczone 4 woluminy (czasopisma lub książki). Napisz abstrakcyjną klasę **Wolumin** z chronionym polem **id**, konstruktorem (automatycznie przypisującym kolejny numer do pola **id**) oraz dziedziczącą po klasie **Wolumin** klasy **Ksiazka** (z polami **tytul** i **autor**) oraz **Czasopismo** (z polami **tytul** i **numerWydania**). W metodzie **main()** klasy **Lab10z3** utwórz obiekt klasy **Ksiazka** i obiekt klasy **Czasopismo**, następnie utwórz tablicę obiektów **tabObj** i umieść w niej stworzone obiekty.

Treść:

Wyniki:

4. Utwórz klasę **Pudelko** zawierającą chronione pola **długosc**, **szerokosc**, **wysokosc**, publiczny konstruktor i publiczne metody **objetosc**, **toString** (wypisującą wszystkie wymiary i objętość). Utwórz klasę **Szescian** pochodną klasy **Pudelko** zawierającą konstruktor jednoargumentowy, który ustawia wszystkie pola (**długosc**, **szerokosc**, **wysokosc**) na wartość podaną w argumencie. W metodzie **main()** klasy **Lab10z4** utwórz obiekt **szescian** i oblicz jego objętość.

Treść:

Wyniki:

5. Utwórz klasę **Uczen** zawierającą chronione pola **nazwisko**, **ocena1**, **ocena2**, publiczny konstruktor i publiczne metody **srednia**, **toString** (zwraca napis składający się z nazwiska i średniej), następnie klasę **Student** będącą pochodną klasy **Uczen** i zawierającą dodatkowe pole prywatne **waga1**, publiczny konstruktor wywołujący konstruktor klasy bazowej i ustawiający pole **waga1**, metodę **srednia**, która oblicza średnią ważoną. W metodzie **main()** klasy **Lab10z5** utwórz jeden obiekt klasy **Uczen** i jeden obiekt klasy **Student** z takimi samymi ocenami i wywołaj dla nich metody obliczające średnie. Wydrukować wyniki.

Treść:

Wyniki: