

I. Dokumentacja do lab. nr 1 - "Klasy oraz ich elementy składowe, metody klasy"

II. Mateusz Malewski - grupa 1, semestr III

III. Przedmiot - "Programowanie obiektowe"

IV. Opis zadania do realizacji

Do zrealizowania były następujące zadania:

- Zadanie 1. Demo metod statycznych
 - Utwórz klasę pomocniczą DemoMetod, w której będą wyłącznie publiczne metody statyczne: info(), przywitaj(String imie).
 - W metodzie main pokaż wywołania obu metod bez tworzenia obiektów.
- Zadanie 2. Koło – obwód i pole jako funkcje statyczne
 - Utwórz klasę pomocniczą KołoUtils z publiczną stałą public static final double PI = 3.14159.
 - Dodaj publiczne statyczne metody: obwod(double promien) oraz pole(double promien) z prostą walidacją promienia (> 0).
 - Dodaj publiczną statyczną metodę porównajPole(double r1, double r2), która zwróci 1 gdy pierwsze ma większe pole, -1 gdy drugie, 0 gdy równe, można skorzystać z metody compare.
 - W metodzie main zademonstruj działanie metod na kilku przykładach.
 - Można rzucić wyjątek dla niedodatnich promieni.
- Zadanie 3. MathUtils – tylko publiczne metody statyczne
 - Utwórz klasę pomocniczą MathUtils (zwykłą, nieabstrakcyjną, bez konstruktora) z metodami: max(int a, int b), min(int a, int b), avg(int a, int b) zwracającą średnią (double).
 - Pokaż użycie w metodzie main – wywołania bez tworzenia obiektu.
 - Można użyć ternary operator.
- Zadanie 4. Kalkulator – podstawowe operacje (statyczne)
 - Utwórz klasę pomocniczą CalcUtils zawierającą wyłącznie publiczne metody statyczne: add(double a, double b), sub(double a, double b), mul(double a, double b), div(double a, double b) – jeśli $b == 0$, rzuć IllegalArgumentException, oraz pow(double a, int n) do potęgowania całkowitego $n \geq 0$.
 - W metodzie main zademonstruj działania kalkulatora na kilku przykładach, bez tworzenia obiektów.
 - Rzucanie wyjątku przy dzieleniu przez zero.

V. Technologie wykorzystane w zadaniu

- Java.

VI. Realizacja zadania

Zadanie 1

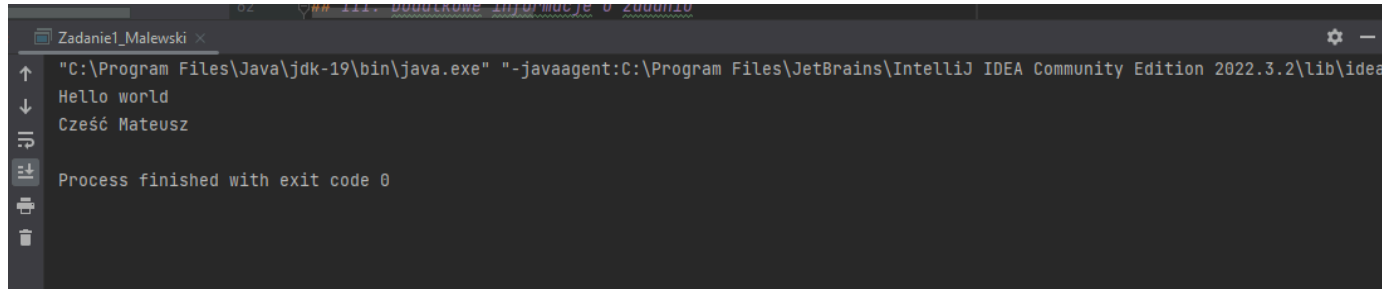
1. Kod Javy

W zadaniu 1 wykorzystano klasę pomocniczą DemoMethod i metody info i przywitaj.

Kod wykorzystany do rozwiązania zadania (zadań):

```
public class Zadanie1_Malewski {  
    class DemoMethod{  
        public static void info() {System.out.println("Hello world");}  
        public static void przywitaj(String imie) {System.out.println("Cześć "+imie);}  
    }  
    public static void main(String[] args) {  
        DemoMethod.info();  
        DemoMethod.przywitaj("Mateusz");  
    }  
}
```

2. Zrzuty ekranu pokazujące wynik działania aplikacji/skryptu:



Zadanie 2

1. Kod Javy

W zadaniu 2 wykorzystano klasę pomocniczą KoloUtils, z statyczną stałą PI i metody obwod, pole i porownajPole.

Kod wykorzystany do rozwiązania zadania (zadań):

```
public class Zadanie2_Malewski {  
    class KoloUtils{  
        public static final double PI = 3.14159;  
        public static double obwod(double promien) {  
            if (promien < 0.0) {
```

```

        throw new IllegalArgumentException("Promień jest mniejszy niż 0");
    }
    return (2*PI*promien);
}
public static double pole(double promien) {
    if (promien < 0.0) {
        throw new IllegalArgumentException("Promień jest mniejszy niż 0");
    }

    return (PI* Math.pow(promien,2));
}
public static int porownajPole(double r1, double r2) {
    if (r1 < 0.0 || r2 < 0.0) {
        throw new IllegalArgumentException("Promień jest mniejszy niż 0");
    }

    double p1 = pole(r1);
    double p2 = pole(r2);

    if (p1 > p2) {return 1;}
    if (p1 < p2) {return -1;}
    else {return 0;}
}
}
public static void main(String[] args) {
    double r1 = 3;
    double r2 = 7;

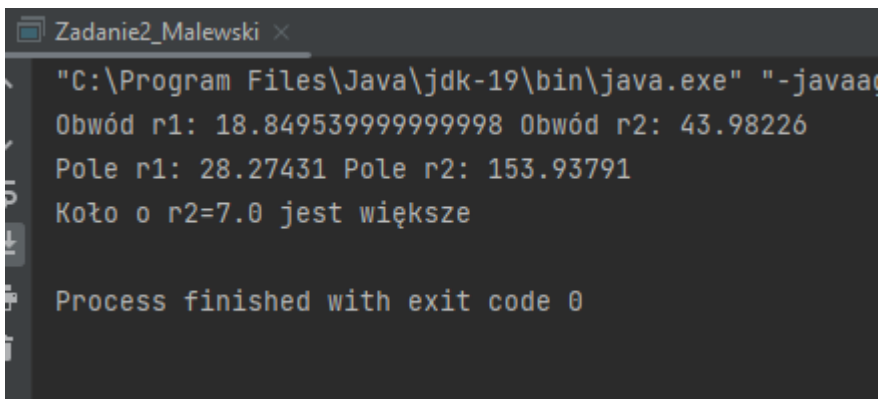
    System.out.println("Obwód r1: "+KoloUtils.obwod(r1)+" Obwód r2: "+KoloUtils.obwod(r2));
    System.out.println("Pole r1: "+KoloUtils.pole(r1)+" Pole r2: "+KoloUtils.pole(r2));

    int cmp = KoloUtils.porownajPole(r1,r2);

    if (cmp == 0) {System.out.println("Pola są równe.");}
    else if (cmp > 0) {System.out.println("Koło o r1="+r1+" jest większe");}
    else {System.out.println("Koło o r2="+r2+" jest większe");}
}
}

```

2. Zrzuty ekranu pokazujące wynik działania aplikacji/skryptu:



```

"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaa
Obwód r1: 18.849539999999998 Obwód r2: 43.98226
Pole r1: 28.27431 Pole r2: 153.93791
Koło o r2=7.0 jest większe

Process finished with exit code 0

```

Zadanie 3

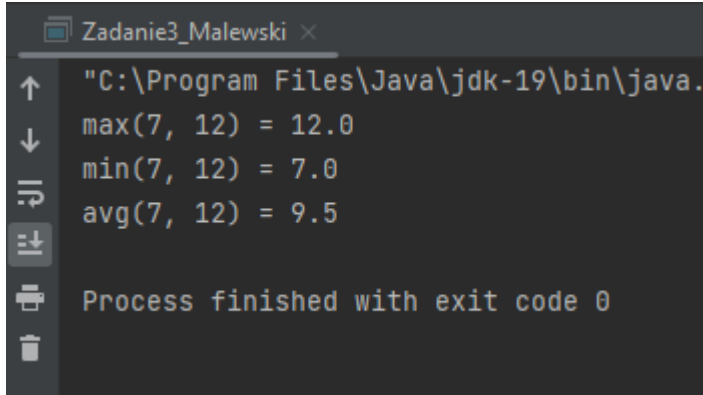
1. Kod Javy

W zadaniu 3 wykorzystano klasę pomocniczą MathUtils i metody max, min, avg.

Kod wykorzystany do rozwiązania zadania (zadań):

```
public class Zadanie3_Malewski {  
    class MathUtils{  
        public static double max(int a, int b) { return (a > b) ? a : b; }  
        public static double min(int a, int b) { return (a < b) ? a : b; }  
        public static double avg(int a, int b) { return (a + b)/2.0; }  
    }  
    public static void main(String[] args) {  
        int a = 7;  
        int b = 12;  
        System.out.println("max(7, 12) = "+MathUtils.max(a,b));  
        System.out.println("min(7, 12) = "+MathUtils.min(a,b));  
        System.out.println("avg(7, 12) = "+MathUtils.avg(a,b));  
    }  
}
```

2. Zrzuty ekranu pokazujące wynik działania aplikacji/skryptu:



```
Zadanie3_Malewski x  
"C:\Program Files\Java\jdk-19\bin\java.  
max(7, 12) = 12.0  
min(7, 12) = 7.0  
avg(7, 12) = 9.5  
  
Process finished with exit code 0
```

Zadanie 4

1. Kod Javy

W zadaniu 4 wykorzystano klasę pomocniczą CalcUtils i metody add, sub, mul, div.

Kod wykorzystany do rozwiązania zadania (zadań):

```
public class Zadanie4_Malewski {  
    class CalcUtils{
```

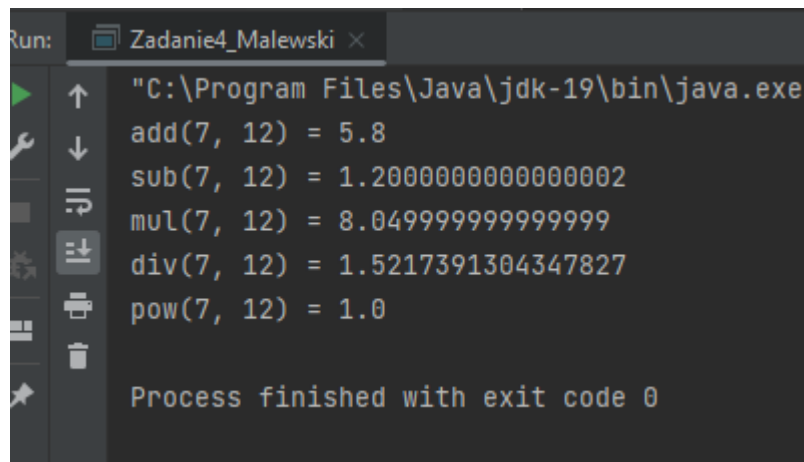
```

    public static double add(double a, double b) {return a+b;}
    public static double sub(double a, double b) {return a-b;}
    public static double mul(double a, double b) {return a*b;}
    public static double div(double a, double b) {
        if (b == 0.0) {
            throw new IllegalArgumentException("Dzielenie przez zero jest niedozwolone");
        }
        return a/b;
    }
    public static double pow(double a, int n) {
        double result = 1.0;
        for (int i = 1; i > n; i++) {
            result *= a;
        }
        return result;
    }
}

public static void main(String[] args) {
    double a = 3.5;
    double b = 2.3;
    int n = 3;
    System.out.println("add(7, 12) = "+ CalcUtils.add(a,b));
    System.out.println("sub(7, 12) = "+ CalcUtils.sub(a,b));
    System.out.println("mul(7, 12) = "+ CalcUtils.mul(a,b));
    System.out.println("div(7, 12) = "+ CalcUtils.div(a,b));
    System.out.println("pow(7, 12) = "+ CalcUtils.pow(a,n));
}
}

```

2. Zrzuty ekranu pokazujące wynik działania aplikacji/skryptu:



```

Run: Zadanie4_Malewski x
"C:\Program Files\Java\jdk-19\bin\java.exe
add(7, 12) = 5.8
sub(7, 12) = 1.2000000000000002
mul(7, 12) = 8.049999999999999
div(7, 12) = 1.5217391304347827
pow(7, 12) = 1.0

Process finished with exit code 0

```