

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

Studium Stacjonarne, sem. 3, 2022/2023

Praca domowa 1
nr grupy: 1 ,

Data wykonania ćwiczenia: 01.10

Data oddania sprawozdania: 13.11

Nazwisko i imię: Kuczawski Kacper

Nr albumu: 20195

Nazwa pliku :prdom1_kuczawski_kacper20195

Zadanie domowe nr 1 .

Opisać i użyć w programie funkcje matematyczne nie występujące w języku C :
abs, addExact, copySign, decrementExact, expm1, floorDiv, floorMod, getExponent, hypot, IEEERemainder, incrementExact, max, min, multiplyExact, negateExact, nextDown, nextUp, nextAfter, min, random, rint, scalb, signum, subtractExact, toDegrees, toIntExact, toRadians, ulp. Opis każdej funkcji umieścić w programie, tak aby był widoczny na wydruku. Wydrukować argument i wartość danej funkcji dla tego argumentu . Funkcje występujące w języku C należy tylko wymienić.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
import static java.lang.Math.*;
public class temp{
    public static void main(String args[]){
        System.out.println("\nFunkcje występujące w C:acos, asin, atan, cbrt,
ceil, floor, pow");
        double x;
        x=2.74;
        System.out.println("\nx=" + x + " abs(x)=" + abs(x) + "\nMetoda abs
oblicza wartość bezwzględną zmiennej");
        int sil;
        sil=3*12*11*10*9*8*7*6*5*4*3*2*1;
        int sil2;
        sil2=3*12*11*10*9*8*7*6*5*4*3*2*1;
        System.out.println("\nx=" + sil + "y=" + sil2 + "\naddExact(x,y)=" +
addExact(sil,sil2) + "\nMetoda addExact oblicza sumę zmiennych,");
        System.out.println("jednak gdy ta przekroczy ich zakres drukuje
błąd");
        double y;
        y=-9;
        System.out.println("\nx="+ x + " y=" + y + " copySign(x,y)=" +
copySign(x,y) + "\nMetoda copySign nadaje");
        System.out.println("pierwszej zmiennej znak drugiej zmiennej");
        sil=-2147483647;
        System.out.println("\nx=" + sil + "decrementExact(x)=" +
decrementExact(sil) + "\nMetoda decrementExact odejmuje 1 od zmiennej,");
        System.out.println("jednak gdy ta przekroczy jej zakres drukuje
błąd");
        x=3;
        System.out.println("\nx=" + x + " expm1(x)=" + expm1(x) + " exp(x)=" +
exp(x));
        System.out.println("Metoda expm1 oblicza wartość e podniesionej");
        System.out.println("do potęgi zmiennej i zmniejsza wynik o 1");
        int z;
        z=1279;
```

```

        System.out.println("\nx=" + z + " floorDiv(x,100)=" + floorDiv(z,100)
+ " floorDiv(x,1000)=" + floorDiv(z,1000));
        System.out.println("Metoda floorDiv zaokrągla zmienną do danej
wartości");
        System.out.println("i wyświetla ilość wielokrotności tej wartości");
        System.out.println("\nx=" + z + " floormod(x,100)=" + floorMod(z,100)
+ " floorMod(x,1000)=" + floorMod(z,1000));
        System.out.println("Metoda floorMod dzieli zmienną przez daną
wartość");
        System.out.println("i wyświetla resztę z dzielenia");
        x=256.74;
        System.out.println("\nx=" + x + " getExponent(x)=" + getExponent(x));
        System.out.println("Metoda getExponent oblicza największy wykładnik
potęgi 2,");
        System.out.println("którego wynik jest mniejszy i najbliższy
zmiennej");
        x=4;
        System.out.println("\nx="+ x + " y=" + y + " hypot(x,y)=" +
hypot(x,y));
        System.out.println("\nMetoda hypot oblicza pierwiastek z wyrażenia
(x^2+y^2)");
        System.out.println("\nx="+ x + " y=" + y + " IEEEremainder(x,y)=" +
IEEEremainder(x,y));
        System.out.println("incrementExact zwraca argument zwiększony o jeden,
zgłaszając wyjątek,");
        System.out.println("jesli wynik przepelnia okreslony typ danych long
lub int, w zaleznosci od tego, który typ danych zostal uzyty w argumentcie
metody.");
        System.out.println("Przed "+z+" Po: "+Math.incrementExact(z));
        System.out.println("max daje maksimum z dwoch argumentow. Argumentem
moze byc int, float, long,double");
        System.out.println("Przed "+x+" i "+y+" Po: "+Math.max(x,y));
        System.out.println("min jest wbudowana funkcja Java, która zwraca co
najmniej dwie liczby.");
        System.out.println("Przed "+x+" i "+y+" Po: "+Math.min(x,y));
        System.out.println("multiplyExact zwraca iloczyn argumentow.");
        int n;
        n=10;
        System.out.println("Przed "+z+" i "+n+" Po:
"+Math.multiplyExact(z,n));
        System.out.println("negateExact zwraca negacje argumentu.");
        System.out.println("Przed "+z+" Po: "+Math.negateExact(z));
        System.out.println("nextDown");
        System.out.println("Przed "+z+" Po: "+Math.nextDown(z));
        System.out.println("nextUp zwraca wartosc zmiennoprzecinkowa
sasiadujaca z podanym parametrem w kierunku dodatniej nieskonczonosci.");
        System.out.println("Przed "+z+" Po: "+Math.nextUp(z));
        System.out.println("nextAfter zwraca liczbe zmiennoprzecinkowa
sasiadujaca z pierwszym argumentem w kierunku drugiego argumentu.");

```

```

        System.out.println("Przed "+x+" i "+y+" Po: "+Math.nextAfter(x,y));
        x=256.74;
        System.out.println("rint zwraca wartosc najblizsza podanej wartosci i
jest rowna matematycznej liczbie calkowitej.");
        System.out.println("Przed "+x+" Po: "+Math rint(x));
        x=4;
        System.out.println("scalb sluzzy do uzyskania wartosci a x 2^ scale.");
        System.out.println("Przed "+z+" i "+n+" Po: "+Math.scalb(z,n));
        System.out.println("signum zwraca funkcje Sign wartosci przekazanej
jako argument.");
        System.out.println("Przed "+y+" Po: "+Math.signum(y));
        System.out.println("subtractExact-zwraca roznice argumentow. Zglosi
wyjatek, jesli wynik przekroczy wartosc int lub long.");
        System.out.println("Przed "+z+" i "+n+" Po:
"+Math.subtractExact(z,n));
        System.out.println("toDegrees sluzzy do konwersji kąta mierzonego w
radianach na w przyblizeniu rownowazny kat mierzony w stopniach.");
        System.out.println("Przed "+x+" Po: "+Math.toDegrees(x));
        System.out.println("toIntExact zwraca wartosc dlugiego argumentu.");
        System.out.println("Przed "+z+" Po: "+Math.toIntExact(z));
        System.out.println("toRadians sluzzy do konwersji kata mierzonego w
stopniach na w przyblizeniu rownowazny kat mierzony w radianach. ");
        x=90;
        System.out.println("Przed "+x+" Po: "+Math.toRadians(x));
        System.out.println("ulp zwraca rozmiar ulp argumentu. Ulp oznacza
jednostke o najmniejszej precyzji.");
        x=256.74;
        System.out.println("Przed "+x+" Po: "+Math.ulp(x));

    }
}

```

Wyniki:

```

Funkcje występujące w C:acos, asin, atan, cbrt, ceil, floor, pow

x=2.74 abs(x)=2.74
Metoda abs oblicza wartość bezwzględną zmiennej
Exception in thread "main" java.lang.ArithmeticException: integer overflow
    at java.base/java.lang.Math.addExact(Math.java:911)
    at temp.main(temp.java:15)

```

(gdy wartości przekraczają zakres)

```
x=2.74 abs(x)=2.74
```

Metoda abs oblicza wartość bezwzględną zmiennej

```
x=1437004800y=10
```

```
addExact(x,y)=1437004810
```

Metoda addExact oblicza sumę zmiennych,
jednak gdy ta przekroczy ich zakres drukuje błąd

```
x=2.74 y=-9.0 copySign(x,y)=-2.74
```

Metoda copySign nadaje
pierwszej zmiennej znak drugiej zmiennej

```
x=-2147483647decrementExact(x)=-2147483648
```

Metoda decrementExact odejmuje 1 od zmiennej,
jednak gdy ta przekroczy jej zakres drukuje błąd

```
x=3.0 expm1(x)=19.085536923187668 exp(x)=20.085536923187668
```

Metoda expm1 oblicza wartość e podniesionej
do potęgi zmiennej i zmniejsza wynik o 1

```
x=1279 floorDiv(x,100)=12 floorDiv(x,1000)=1
```

Metoda floorDiv zaokrągla zmienną do danej wartości
i wyświetla ilość wielokrotności tej wartości

```
x=1279 floorMod(x,100)=79 floorMod(x,1000)=279
Metoda floorMod dzieli zmienną przez daną wartość
i wyświetla resztę z dzielenia

x=256.74 getExponent(x)=8
Metoda getExponent oblicza największy wykładnik potęgi 2,
którego wynik jest mniejszy i najbliższy zmiennej

x=4.0 y=-9.0 hypot(x,y)=9.848857801796104

Metoda hypot oblicza pierwiastek z wyrażenia (x^2+y^2)

x=4.0 y=-9.0 IEEEremainder(x,y)=4.0
incrementExact zwraca argument zwiększony o jeden, zgłaszając wyjątek,
jeśli wynik przepelnia określony typ danych long lub int, w zależności od tego, który typ danych został użyty w argumencie metody.
Przed 1279 Po: 1280
max daje maksimum z dwóch argumentów. Argumentem może być int, float, long, double
Przed 4.0 i -9.0 Po: 4.0
min jest wbudowana funkcja Java, która zwraca co najmniej dwie liczby.
Przed 4.0 i -9.0 Po: -9.0
multiplyExact zwraca iloczyn argumentów.
Przed 1279 i 10 Po: 12790
negateExact zwraca negację argumentu.
Przed 1279 Po: -1279
nextDown
Przed 1279 Po: 1278.9999
nextUp zwraca wartość zmiennoprzecinkową sąsiadującą z podanym parametrem w kierunku dodatniej nieskończoności.
Przed 1279 Po: 1279.0001

nextAfter zwraca liczbę zmiennoprzecinkową sąsiadującą z pierwszym argumentem w kierunku drugiego argumentu.
Przed 4.0 i -9.0 Po: 3.9999999999999996
rint zwraca wartość najbliższą podanej wartości i jest równa matematycznej liczbie całkowitej.
Przed 256.74 Po: 257.0
scalb służy do uzyskania wartości a x 2^ scale.
Przed 1279 i 10 Po: 1309696.0
signum zwraca funkcję Sign wartości przekazanej jako argument.
Przed -9.0 Po: -1.0
subtractExact-zwraca różnicę argumentów. Zgłosi wyjątek, jeśli wynik przekroczy wartość int lub long.
Przed 1279 i 10 Po: 1269
toDegrees służy do konwersji kąta mierzonego w radianach na w przybliżeniu równoważny kąt mierzony w stopniach.
Przed 4.0 Po: 229.1831180523293
toIntExact zwraca wartość długiego argumentu.
Przed 1279 Po: 1279
toRadians służy do konwersji kąta mierzonego w stopniach na w przybliżeniu równoważny kąt mierzony w radianach.
Przed 90.0 Po: 1.5707963267948966
ulp zwraca rozmiar ulp argumentu. Ulp oznacza jednostkę o najmniejszej precyzji.
Przed 256.74 Po: 5.684341886080802E-14

Process finished with exit code 0
```