

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

Studium Stacjonarne, sem. 3, 2022/2023

Sprawozdanie nr : 5,
nr grupy: 1,
dzień: wtorek,
godz. 12:00.

Data wykonania ćwiczenia: 24.11

Data oddania sprawozdania: 01.12

Nazwisko i imię: Kuczawski Kacper

Nr albumu: 20195

Nazwa pliku : lab5_kuczawski_kacper20195

1. Utworzyć klasę publiczną **lab5Zad1**. W metodzie **main()** tej klasy utworzyć dwa obiekty typu **BigDecimal** oraz obliczyć i wydrukować ich sumę, różnicę, iloczyn i iloraz.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
public class temp{
    public static void main(String args[]){
        BigDecimal x=new BigDecimal("3891383912.4592");
        BigDecimal y=new BigDecimal("1259821344.42");
        System.out.print("\nx= " + x);
        System.out.print("\ny= " + y);
        System.out.print("\nSuma: " + (x.add(y)));
        System.out.print("\nRóżnica: " + (x.subtract(y)));
        System.out.print("\nIloczyn: " + (x.multiply(y)));
        System.out.print("\nIloraz: " + (x.divide(y, ROUNDINGMODE.CEILING)));
    }
}
```

Wyniki:

```
x= 3891383912.4592
y= 1259821344.42
Suma: 5151205256.8792
Różnica: 2631562568.0392
Iloczyn: 4902448512248708932.397664
```

Utworzyć klasę publiczną **lab5Zad2**. W metodzie **main()** tej klasy zamieścić obliczenia wartości poniższych wyrażeń z operatorami bitowymi, porównać wyniki z podanymi wynikami dla podanych argumentów, w sprawozdaniu przedstawić odpowiednie obliczenia na poziomie bitów dla jednej wybranej pary argumentów.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
public class temp {
    public static void main(String args[]) {
        int k = 1;
        int bit[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        int bit1[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        int bit2[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        short lanbit1 = 12;
        short lanbit2 = -35;
        for (int i = 0; i < 16; i++) {
            if ((lanbit1 & k) != 0) bit1[i] = 1;
            else bit1[i] = 0;
            k = k * 2;
        }
        for (int i = 15; i >= 0; i--) System.out.print(" " + bit1[i] + " ");
    }
}
```

```

        System.out.print("\n");
        k = 1;
        for (int i = 0; i < 16; i++) {
            if ((lanbit2 & k) != 0) bit2[i] = 1;
            else bit1[i] = 0;
            k = k * 2;
        }
        for (int i = 15; i >= 0; i--) System.out.print(" " + bit2[i] + " ");
        System.out.print("\n");
    }
}

```

Wyniki:

0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	

3. Utworzyć klasę publiczną **lab5Zad3**. Napisać metodę statyczną **toBinary** realizującą konwersję liczby całkowitej nieujemnej na postać binarną. W klasie **main()** wywołać tę metodę dla przykładowych wczytywanych liczb. Wydrukować wyniki.

Treść:

```

import java.io.*;
import java.util.*;
import java.text.*;
public class temp {
    public static void main(String args[]) {
        Scanner sc=new Scanner(System.in);
        int x;
        int p=1;
        do{
            System.out.print("\nPodaj liczbę dziesiętną:");
            x=sc.nextInt();
            System.out.print("\n");
            temp.toBinary(x);
            System.out.print("\nCzy chcesz podać kolejną liczbę? 0=n, 1=t");
            p=sc.nextInt();
        }
        while(p>0);
    }
    public static void toBinary(int n){
        int bit[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        int k=1;
        for(int i=0;i<16;i++) {
            if ((n&k)!=0) bit[i]=1; else bit[i]=0;
            k=k*2;
        }
        for (int i=15;i>=0;i--) System.out.print(" " + bit[i] + " ");
        System.out.print("\n");
        k=1;
    }
}

```

Wyniki:

```
Podaj liczbę dziesiętną:1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Czy chcesz podać kolejną liczbę? 0=n, 1=t255

Podaj liczbę dziesiętną:255

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
|
Czy chcesz podać kolejną liczbę? 0=n, 1=t0
```

4. Utworzyć klasę publiczną **lab5Zad4**. W metodzie **main()** tej klasy zrealizować następujące obliczenia (obliczenia należy rozpocząć od maski **mask0=0;**)

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
public class temp{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int x;
        int l;
        int b1, b2;
        System.out.print("Podaj liczbę:");
        l=sc.nextInt();
        temp.toBinary(l);
        System.out.print("Podaj numer bitu do zmiany:");
        x=sc.nextInt();
        int maska0=0;
        temp.toBinary(maska0);
        int maska1=~maska0;
        temp.toBinary(maska1);
        int maska2a=maska1<<1;
        temp.toBinary(maska2a);
        int maska2=~maska2a;
        temp.toBinary(maska2);
        int maska3=maska2<<(8-(x));
        temp.toBinary(maska3);
        b1=l|maska3;
        int maska4=~maska3;
        temp.toBinary(maska4);
        b2=l&maska4;
        System.out.print("\nLiczba z określonym 1:");
        temp.toBinary(b1);
        System.out.print("\nLiczba z określonym 0:");
```

```

temp.toBinary(b2);
int maskaM0=maska1<<4;
temp.toBinary(maskaM0);
int maskaM1=~(maska1<<4);
temp.toBinary(maskaM1);
int maskaS0=~(maska1<<4);
temp.toBinary(maskaS0);
int maskaS1=maska1<<4;
temp.toBinary(maskaS1);
}
static void toBinary(int n){
    int bit[] = {0, 0, 0, 0, 0, 0, 0, 0};
    int k=1;
    for(int i=0;i<8;i++) {
        if ((n&k)!=0) bit[i]=1; else bit[i]=0;
        k=k*2;
    }
    for (int i=7;i>=0;i--) System.out.print(" " + bit[i] + " ");
    System.out.print("\n");
    k=1;
}
}

```

Wyniki:

```

Podaj liczbę:254
 1 1 1 1 1 1 1 0
Podaj numer bitu do zmiany:2
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
1 0 1 1 1 1 1 1

Liczba z określonym 1: 1 1 1 1 1 1 1 0

Liczba z określonym 0: 1 0 1 1 1 1 1 0
1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1
1 1 1 1 0 0 0 0

```

5. Utworzyć klasę publiczną **lab5Zad5**. W klasie tej zamieścić metodę statyczną **valueOfGroup** (z wykładu) wyznaczającą wartość dziesiętną grupy **n**-bitów począwszy od pozycji **p**. W metodzie **main()** wczytać argumenty **n** i **p** z zabezpieczeniem formatu i wartości, następnie wywołać dla nich metodę **valueOfGroup**.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class temp {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n, p;
        short x;
        System.out.print("Podaj liczbę:");
        x = sc.nextShort();
        temp.toBinary(x);
        System.out.print("Podaj ilość bitów do odczytania:");
        n = sc.nextInt();
        System.out.print("Podaj pozycję początku odczytania:");
        p = sc.nextInt();
        System.out.println("Liczba dziesiętnie:"+temp.valueOfGroup(x,n,p));
    }
    static void toBinary(short n){
        int bit[] = {0, 0, 0, 0, 0, 0, 0, 0};
        int k=1;
        for(int i=0;i<8;i++) {
            if ((n&k)!=0) bit[i]=1; else bit[i]=0;
            k=k*2;
        }
        for (int i=7;i>=0;i--) System.out.print(" " + bit[i] + " ");
        System.out.print("\n");
        k=1;
    }
    static int valueOfGroup(short x, int n, int p){
        return x>>(p+1-n)&~(~0<<n);
    }
}
```

Wyniki:

Podaj liczbę:254	HEX	FE
1 1 1 1 1 1 1 0	DEC	254
Podaj ilość bitów do odczytania:9	OCT	376
Podaj pozycję początku odczytania:8	BIN	1111 1110
Liczba dziesiętnie:254		

6. Utworzyć klasę publiczną **lab5Zad6**. W metodzie **main()** wczytać liczbę ujemną typu **int**, następnie zastosować operatory **>>** i **>>>** przesuwające o 2 bity i porównać wyniki. W sprawozdaniu przedstawić obliczenia i ich interpretację.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class temp {
    public static void main(String[] args) {
        int u = -7;
        int u1 = u>>u;
        int u2 = u>>>u;
        int u3 = u1>>>u1;
        int u4 = u2>>u2;
        System.out.println("u:"+u);
        System.out.println("u1 = >>u:"+u1);
        System.out.println("u2 = >>>u:"+u2);
        System.out.println("u3 = >>>u1:"+u3);
        System.out.println("u4 = >>u2:"+u4);
    }
}
```

Wyniki:

```
u: -7
u1 = >>u: -1
u2 = >>>u: 127
u3 = >>>u1: 1
u4 = >>u2: 0
```

7. Opisać w sprawozdaniu cel stosowania i działanie mechanizmu rozszerzenia znakowego (ang. sign extension).

Mechanizm ten pozwala zmieniać typ danych, poprzez rozszerzenie bitowe. Np. double -> int