

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

Studium Stacjonarne, sem. 3, 2022/2023

Sprawozdanie nr : 6,
nr grupy: 1,
dzień: wtorek,
godz. 12:00.

Data wykonania ćwiczenia: 20.12

Data oddania sprawozdania: 09.01

Nazwisko i imię: Kuczawski Kacper

Nr albumu: 20195

Nazwa pliku: lab6_kuczawski_kacper20195

Zad 1. Utworzyć klasę publiczną **Lab6Zad1**. W metodzie **main()** tej klasy utworzyć i wydrukować obiekty typu **String** utworzone w następujący sposób

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class temp{
    public static void main(String args[]){
        String s1="ANS";
        System.out.print("\ns1: " + s1);
        String s2=new String("ANS");
        System.out.print("\ns2: " + s2);
        String s3=s1+s2;
        System.out.print("\ns3: " + s3);
        char[] tabZ={'A','N','S'};
        String s4=new String(tabZ);
        System.out.print("\ns4: " + s4);
        Date data=new Date();
        String s5="Data: "+data;
        System.out.print("\n " + s5);
    }
}
```

Wyniki:

```
s1: ANS
s2: ANS
s3: ANSANS
s4: ANS
Data: Mon Jan 09 22:15:01 CET 2023
Process finished with exit code 0
```

Zad. 2. W metodzie **main()** klasy **Lab6Zad1**, dla obiektu **String liczbaStr="123"**; obliczyć wartość numeryczną po konwersji na typ **int** przy zastosowaniu każdej z metod

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class temp{
    public static void main(String args[]){
        String liczbaS1="123";
        int x, y;
        x=Integer.parseInt(liczbaS1);
        y=Integer.valueOf(liczbaS1);
        System.out.print("\nx: " + x);
        System.out.print("\ny: " + y);
    }
}
```

Wyniki:

```
x: 123
y: 123
```

Zad. 3. Utworzyć klasę publiczną **Lab6Zad3**. W metodzie **main()** tej klasy wykonać następujące działania

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class temp{
    public static void main(String args[]){
        String Lan="0x";
        int lHex=0, lsoHex=0, lsd1Hex=0, lsd2Hex=0, WSHex=0, DSHex=0;
        Scanner sc=new Scanner(System.in);
        do{
            System.out.println("\nPodaj x (między 10000 a 10FFFF): ");
            String Lan2=Lan+sc.next();
            lHex=Integer.decode(Lan2);
        }
        while(lHex<65536&&lHex>111411);
        String so="0x10000";
        lsoHex=Integer.decode(so);
        String sd1="0xD800";
        lsd1Hex=Integer.decode(sd1);
        String sd2="0xDC00";
        lsd2Hex=Integer.decode(sd2);
        WSHex=lsd1Hex+(lHex-lsoHex);
        DSHex=lsd2Hex+(lHex-lsoHex);
        System.out.println("\nWyższy surogat int: " + WSHex);
        System.out.printf("\nWyższy surogat hex: 0x%x", WSHex);
        System.out.println("\nDolny surogat int: " + DSHex);
        System.out.printf("\nDolny surogat hex: 0x%x", DSHex);
    }
}
```

Wyniki:

```
Podaj x (między 10000 a 10FFFF):
10AC92

Wyższy surogat int: 1082514

Wyższy surogat hex: 0x108492
Dolny surogat int: 1083538

Dolny surogat hex: 0x108892
```

Zad 4. Dla obiektu `String val1 = "\u5B66\uD8F0\uDE30";`

a) wyznaczyć długość łańcucha (metoda `length()`)

b) wyznaczyć liczbę jednostek kodowych w sposób następujący

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class temp{
    public static void main(String args[]){
        String val1 = "\u5B66\uD8F0\uDE30";
        int d = val1.length();
        int l = val1.codePointCount(0, d);
        System.out.println("Długość: "+d+"\nLiczba jednostek: "+l);
    }
}
```

Wyniki:

```
Długość: 3
Liczba jednostek: 2
```

Zad 5. a) Dla obiektu `String pozdrowienie="Hello";` wydrukować wartość zwracaną przez wywołanie metody `charAt(0)` oraz metody `codePointAt(0)`.

b) Dla obiektu `val1` sprawdzić rezultat wykrywania jednostek kodowych, podać i interpretację.

Treść:

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

        String pzdr = "Greetings";
        char lit = pzdr.charAt(0);
        int cp = pzdr.codePointAt(0);

        int cp1=0, cp2=0;

        String val1 = "\u0041\uD8F0\uDE30";
        cp1 = val1.codePointAt(0);
        cp2 = val1.codePointAt(1);

        System.out.println("Greetings");
        System.out.println("\nLitera 1: "+lit+"\nJednostka kodowa dla 0: "+cp);
        System.out.println("\nVal1");
    }
}
```

```

        System.out.println("\nJednostka kodowa dla 0: "+cp1+"Jednostka kodowa
dla 1: "+cp2);

        System.out.println("Jednostka kodowa jest znakiem surogatowym :"+
+Character.isSupplementaryCodePoint(cp1));
        System.out.println("Jednostka kodowa jest znakiem surogatowym :"+
+Character.isSupplementaryCodePoint(cp2));
    }
}

```

Wyniki:

```

Greetings

Litera 1: G
Jednostka kodowa dla 0: 71

Vall

Jednostka kodowa dla 0: 65Jednostka kodowa dla 1: 311856
Jednostka kodowa jest znakiem surogatowym :false
Jednostka kodowa jest znakiem surogatowym :true

```

Zad. 6. Zdefiniować odpowiednie przykładowe obiekty i wykonać następujące działania:

a) Porównać dwa obiekty typu **String** stosując metody **equals**, **equalsIgnoreCase**, **compareTo** i **compareToIgnoreCase**.

Treść:

```

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

        String s1 = "Greetings";
        String s2 = "Cyberpunk";
        String s3 = "greetings";
        String s4 = "punk";
        String s5 = "Cyberpunk";

        System.out.println("Łańcuchy znaków: "+s1+", "+s2+", "+s3+", "+s4+",
+s5);
        System.out.println("Porównanie s1 i s2: "+s1.equals(s2));
        System.out.println("Porównanie s1 i s3: "+s1.equals(s3));
        System.out.println("Porównanie s2 i s5: "+s2.equals(s5));
    }
}

```

Wyniki:

```
Łańcuchy znaków: Greetings, Cyberpunk, greetings, punk, Cyberpunk
Porównanie s1 i s2: false
Porównanie s1 i s3: false
Porównanie s2 i s5: true
```

Zad. 7. Dany jest łańcuch `String str = " zielony,niebieski,czerwony"`, zastosować metodę `split` do wydobycia poszczególnych podciągów w postaci `String results[]=str.split(",")`.

Treść:

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

        String str = "zielony,niebieski,czerwony";
        String results[] = str.split(",");

        for(int i=0; i<results.length; i++){
            System.out.println("Łańcuch "+(i+1)+" po rozdzieleniu:
"+results[i]);
        }
    }
}
```

Wyniki:

```
Łańcuch 1 po rozdzieleniu: zielony
Łańcuch 2 po rozdzieleniu: niebieski
Łańcuch 3 po rozdzieleniu: czerwony
```

Zad. 8. Użyć metody `substring` do wydobycia określonego podciągu z łańcucha (`substring (poz1,poz2)`, `poz1` oznacza numer pozycji, od której należy pobierać podciąg, a `poz2` pozycję, której już nie chce się pobierać znaków)

Treść:

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

        String s1 = "Cyberpunk";
        System.out.println(s1.substring(5, 9));

    }
}
```

Wyniki:

```
C:\Users\User\Documents\openjdk-8-jre\bin>java Lab6z9
punk
Process finished with exit code 0
```

Zad. 9. W metodzie `main()` klasy publicznej `Lab6z9` utworzyć obiekt `s1` klasy `String` zawierający napis `"programowanie"`. Następnie utworzyć obiekt `sb1` klasy `StringBuilder` w oparciu o obiekt `s1`. Zmodyfikować obiekt `sb1` tak, aby zawierał napis `"programowanie obiektowe I"` (zastosować metodę `append()`), w kolejnym kroku zmienić znak `'p'` na `'P'` przy zastosowaniu metody `setCharAt()`, Przeprowadzić konwersję obiektu `sb1` na obiekt klasy `String` i wydrukować.

Treść:

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

        String s1 = "programowanie";
        System.out.println(s1);
        StringBuilder sb1 = new StringBuilder(s1);
        sb1.append(" obiektowe I");
        sb1.setCharAt(0, 'P');
        sb1.toString();
        System.out.println(sb1);
    }
}
```

Wyniki:

```
programowanie
Programowanie obiektowe I
Process finished with exit code 0
```