

Jegyzőkönyv

Szilágyi-Czumbil Ede Balázs Neptun kód: CMY9W3 Git repó:<https://github.com/edeszilagy>

0. hét - „Helló, Berners-Lee!”

Java, C++ összehasonlítás

Könyvek:C++: Benedek Zoltán, Levendovszky Tihamér Szoftverfejlesztés C++ nyelven Java: Nyékyné Dr. Gaizler Judit et al. Java 2 útikalauz programozóknak 5.0 I-II

A Java nyelv teljesen objektum orientált nyelv. Ezzel szemben a C++ lehetőséget ad a generikus programozásra is. A két programnyelv változó kezelése eltérő. Java esetén minden objektum referencia. Ez azt jelenti, hogy az értéket közvetlen a referencián keresztül érjük el. Mindkét nyelv támogatja a publikus, privát és statikus objektum kezelést. Fordító szempontjából amíg a C++ kódot elég natívan fordítani, addig a Java-hoz szükség van egy virtuális fordítóra, ami futtatja a kódot. Emiatt nagyobb az erőforrás igénye is. Java-ban nem nagyon kell foglalkozni a memória szeméttel, mivel van automatikus garbage collector, ami üríti azt, míg C++-nál fel kell szabadítani a memóriát

Python

Könyv: Forstner Bertalan, Ekler Péter, Kelényi Imre: Bevezetés a mobilprogramozásba. Gyors prototípus-fejlesztés Python és Java nyelven

A Python tulajdonképpen egy szkriptnyelv, de nagyon sok csomagot is és beépített eljárást is tartalmaz, ezért komolyabb alkalmazások megírására és komolyabb problémák megoldására is használható. Más modulokkal is együtt tud működni egy Python komponens. A Python egy nagyon magas szintű programozási nyelv. Python esetén nincs szükség fordítás-ra. A Python interpreter elérhető számos platformon. A Pythont könnyű használni, megbízható és jelentős támogatást biztosít hibák javítására. A Pythonban minden adat objektumként szerepel. A rajtuk végzendő műveleteket az objektum típusa határozza meg, amit a rendszer futási időben határoz meg, így nekünk nem kell megadni. A következő típusok lehetnek: szám, string, tuple, list, dictionary. A számok lehetnek egészek, decimálisak, oktálisak vagy akár hexadecimálisak is. Szöveg típus esetén a szöveget két aposztróf közé írva kell megadni.

1. hét - „Helló, Arroway!”

OO szemlélet

A módosított polártranszformációs normális generátor beprogramozása Java nyelven. Mutassunk rá, hogy a mi természetes saját megoldásunk (az algoritmus egyszerre két normálist állít elő, kell egy példánytag, " amely a nem visszaadottat tárolja és egy logikai tag, hogy van-e tárolt vagy futtatni kell az algoritmust.) és az OpenJDK, Oracle JDK-ban a Sun által adott OO szervezés ugyanaz! Segédlink: https://arato.inf.unideb.hu/batfai.norbert/UDPROG/deprecated/Prog1_5.pdf (16-22 fólia)

A feladat lényege, hogy készítsünk egy olyan Java programot, amely két normális eloszlású számot állít elő és hasonlítsuk össze a JDK fejlesztő által készített polártranszformációs algoritmussal.

A kód:

```
public class PolarGenerator {

    boolean nincsTarolt = true;
    double tarolt;

    public PolarGenerator() {
        nincsTarolt = true;
    }

    public double kovetkezo() {
        if(nincsTarolt) {
            double u1, u2, v1, v2, w;
            do {
                u1 = Math.random();
                u2 = Math.random();

                v1 = 2 * u1 - 1;
                v2 = 2 * u2 - 1;

                w = v1 * v1 + v2 * v2;
            } while(w > 1);
            double r = Math.sqrt(-2 * Math.log(w) / w);

            tarolt = r * v2;

            nincsTarolt = !nincsTarolt;
            return r * v1;
        } else {
```

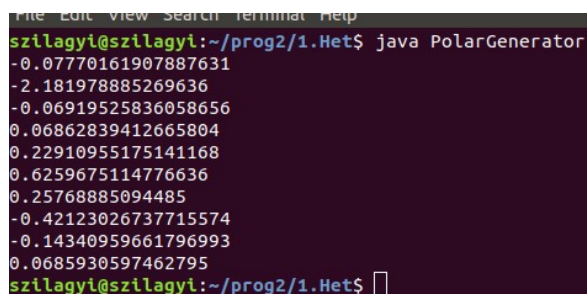
```

        nincsTarolt = !nincsTarolt;
        return tarolt;
    }
}

public static void main(String[] args) {
    PolarGenerator pg = new PolarGenerator();

    for(int i = 0; i < 10; i++) {
        System.out.println(pg.kovetkezo());
    }
}
}

```



```

File Edit View Search Terminal Help
szilagyi@szilagyi:~/prog2/1.Het$ java PolarGenerator
-0.07770161907887631
-2.181978885269636
-0.06919525836058656
0.06862839412665804
0.22910955175141168
0.6259675114776636
0.25768885094485
-0.42123026737715574
-0.14340959661796993
0.0685930597462795
szilagyi@szilagyi:~/prog2/1.Het$ 

```

Miután futtatuk:

„Gagyí”:

Az ismert formális2 „while (x <= t && x >= t && t != x);” tesztkérdéstípusra adj a szokásosnál (miszerint x, t az egyik esetben az objektum által hordozott érték, a másikban meg az objektum referenciája) „mélyebb” választ, írd Java példaprogramot mely egyszer végtelen ciklus, más x, t értékekkel meg nem! A példát építsd a JDK Integer.java forrására3 , hogy a 128-nál inkluzív objektum példányokat poolozza!

a program ami végtelen ciklust ad:

```

public class Gagyí {

    public static void main(String[] args) {

```

```

Integer i = 130;
Integer j = 130;

System.out.println("i = " + i + " j = " + j);

while(i <= j && i >= j && i != j) {

}

}
}

```

```

szilagyi@szilagyi: ~/prog2/1.Het
File Edit View Search Terminal Help
szilagyi@szilagyi:~/prog2/1.Het$ javac Gagy1.java
szilagyi@szilagyi:~/prog2/1.Het$ java Gagy1
i = 130 j = 130

```

lefuttatva:

a program ami nem végtelen ciklust ad:

```

public class Gagy1 {

    public static void main(String[] args) {

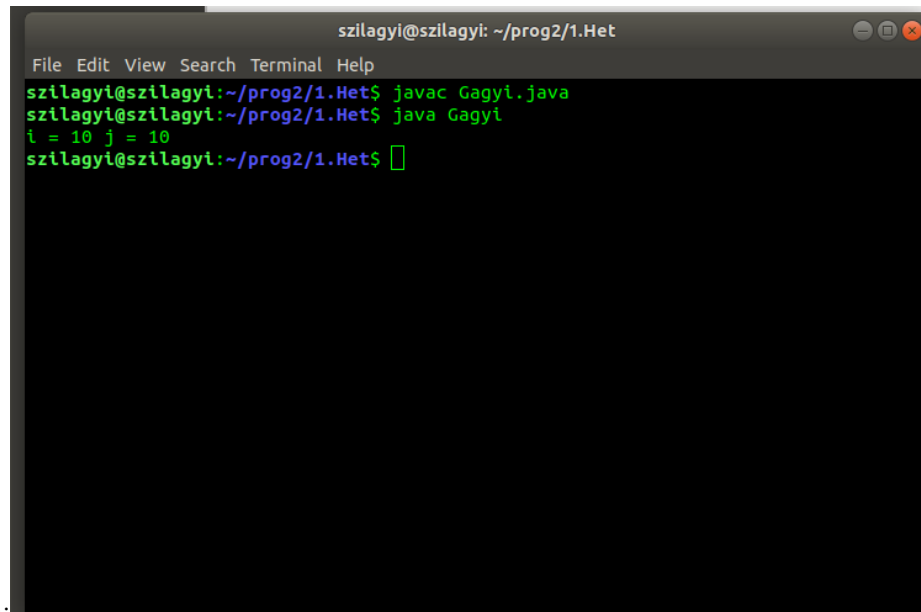
        Integer i = 10;
        Integer j = 10;

        System.out.println("i = " + i + " j = " + j);

        while(i <= j && i >= j && i != j) {

```

```
}  
}  
}
```

A terminal window titled 'szilagyi@szilagyi: ~/prog2/1.Het' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
szilagyi@szilagyi:~/prog2/1.Het$ javac Gagyi.java  
szilagyi@szilagyi:~/prog2/1.Het$ java Gagyi  
i = 10 j = 10  
szilagyi@szilagyi:~/prog2/1.Het$
```

lefuttatva: