Java Thread

appSchool.co.il

Thread

thread הוא רצף של פעולות שמתבצעות במקביל לקטעי קוד אחרים.

ריבוי Thread נקרא multi threading

לדוגמה : אני יכול לשמוע מוזיקה ובמקביל לערוך מסמך word ולכל אחד יש thread במקביל.

שני threadים יכולים לעבוד על אותם נתונים

thread

1. לכל תכנית יש את ה-thread הראשי
2. המטרה של thread הוא לתת לנו את ההרגשה שרצים מספר קטעי קוד במקביל על מספר מעבדים.
3. בפועל - קיים רק מעבד 1
4. תור הפקודות - בתור זה עומדות פקודות בהמתנה לביצוע.

cpu - כל thread הוא רצף של פעולות שמתבצע בנפרד.לכל thread מוקצה מעבד וירטואלי(virtual cpu).

data - נתוני ה-thread יכולים להיות משותפים ליותר מ-thread אחד.

code - רצף של פעולות קוד אשר מתבצעות ב-thread

דוגמה 1 ל-thread ב-java

```
public class downloadImg extends
Thread
{
   Bitmap b;
String url;
  public downloadImg(String url)
 {
     this.url=url;
 }
public void run()
{
       b=downloadfromInternet(url);
}
public void downloadfromInternet()
{
     .........
}
```

```
public class MainActivity extends Activity
{
 downloadImg    downloadImgA=new
downloadImg("http://www.g.co.il/img1");
 downloadImg    downloadImgB=new
downloadImg("http://www.g.co.il/img1");
downloadImgA.start();
downloadImgB.start();

}
```

דוגמה 2 - ל-trread

```
public class person extends Thread {

    private String fname;
    private String lname;
    int age;
    public person(String fname, String
lname, int age) {
        this.fname = fname;
        this.lname = lname;
        this.age = age;
    }
    public String getFname() {
        return fname;
    }
    public void setFname(String fname)
{
        this.fname = f
```

```
@Override
public void run() {
for(int i=0;i<10;i++)
  {                    System.out.println
(Thread.currentThread().getName());
try
{
    Thread.sleep(2000);
}
 catch (InterruptedException e) {
e.printStackTrace();
  }                System.out.println(this.
fname);
      }
}
```

המשך דוגמה thread 2

```
public class Program {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
      person p1=new person("asaf","david",5);
      person p2=new person("eli","shalom",6);
      p1.start();
      p2.start();
}
```

סיכום דוגמה 2

4 שלבים ליצירת thread

1. יורשים מ-thread
2. ביצוע override לפונקציה run
3. ניגשים ל-mainActivity ויוצרים אובייקט
4. מפעילים את הפונקציה start

דוגמה 3- Thread

```
public class person implements
Runnable {    private String fname;
        private String lname;
        int age;
        public person(String fname, String
lname, int age) {
                this.fname = fname;
                this.lname = lname;
                this.age = age;
        }
        public String getFname() {
                return fname;
        }
        public void setFname(String fname)
{
                this.fname = fname;
        }
```

```
@Override
public void run() {
for(int i=0;i<10;i++)
  {                    System.out.println
(Thread.currentThread().getName());
try
{
    Thread.sleep(2000);
}
 catch (InterruptedException e) {
e.printStackTrace();
  }                 System.out.println(this.
fname);
        }
}
```

המשך דוגמא thread 3

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
   person p1=new person("asaf","david",5);
   person p2=new person("eli","shalom",6);
   Thread t1=new Thread(p1);
   Thread t2=new Thread(p2);
   t1.start();
   t2.start();
}
```

סיכום דוגמה 3

4 שלבים ליצירת thread

1. מבצעים implement לממשק Runnable
2. ביצוע override לפונקציה run
3. ניגשים ל-mainActivity ויוצרים אובייקט מסוג Thread ומעבירים לו ב-constructor אובייקט שמבצע implement ל -Runnable
4. מפעילים את הפונקציה start לאובייקט מסוג Thread

sleep

Thread.sleep הנה פונקציה סטטית שמשהה את פעולת ה-thread למשך x אלפיות שניה.

stop

הפעולה stop עוצרת את ה-thread

רצוי לא להשתמש בפעולה זו כי היה יכולה להשאיר אותנו במצב לא צפוי.

בדרך כלל נעצור thread באמצעות דגלים,counter וכו. וכך לא נגיע למצבים בלתי צפויים.

תזמון thread

ניתן לקבוע ל-thread סדר קדימויות

1. MAX_PRIORITY
2. NORM_PRIORITY
3. MIN_PRIORITY

THREAD עם דרגת קדימות גבוה יותר יתקבל ל-cpu.

synchronization

1. יהיו מקרים בהם שני theadים יפעלו על אותם נתונים
2. במקרה כזה thread אחד יכול להפסיק ו-thread שני מתחיל ולהפך.
3. דוגמה למחסן אוכל שמושכים ממנו אוכל למספר מטבחים.ומזרימים למחסן אוכל במקביל.מטבח יכול לקבל אורז רק במידה ויש יותר מ-1 קילו במחסן.

פתרון הבעיה

1. synchronized - אם thread נכנס עם synchronize אזי ה thread נועל את האובייקט רק עבורו.
2. כאשר הוא יסיים המנעול יפתח ו-thread אחר יוכל לגשת לאובייקט.

notify() and wait()

1. פעולות אלו שייכות למחלקה object
2. אם במהלך thread מופעלת הפונקציה wait אזי ה-thread עוצר כדי להעיר אותו נאלץ להפעילו על ידי notify אבל הוא לא יוכל להתעורר כי הוא קפוא.ולכן הערתו תעשה על ידי thread אחר.
3. כל עוד לא הופעל על ה-thread notify או notifyall אזי ה-thread ימשיך להיות קפוא.
4. notify או wait יכולים להיות מופעלות רק מתוך בלוק של synchronized

יום ללא תכנות הוא יום מבוזבז!