

Treads

في معظم البرامج الواقعية نحتاج نعمل اكتر من عملية في نفس الوقت مش ورا بعض زي مثلا الجيمز الي بتحتاج تبقا مشغلة ميوزك وتأثيرات صوتية وبترسم الشخصيات وتحركهم قدامك، وهنا نحتاج الثريدث لانها بتمكنا نعمل اكتر من عملية في نفس الوقت في نفس البرنامج

معلومة عل الماشي: فعليا المعالج مش بيقدر يعمل غير عملية واحدة في نفس الوقت، ولكن الثريدث بتخليه ينفذ جزء من عملية وبعد كدة ينفذ جزء من عملية تاني وهكذا علشان يوهمك انهم بيتعملو مع بعض

والجافا فيها الكلاس Thread هو اللي بيمثل الثريد شيء منطقي 😂'

معلومة عل الماشي: أي برنامج في الجافا لازم يقون فيه علي الأقل Thread واحد وهو ال Main thred واللي بيتعمل تلقائيا ودالة main بتشتغل فيه

اكشلي في طيرقتين أساسيتين ممكن تعمل بيهم ثريد :

1- اني اورث الكلاس Thread واعمل Override لدالة run أي كود انا محتاجة في الكلاس اكنة كلاس عادي خالص وبكتتب في دالة run اللي انا عملتها Override الكود الي انا عايزة يتنفذ لما الثريد يشتغل، وعلشان اشغل الثريد بعمل كائن من الكلاس اللي انا عملته وبعد كدة بعمل call لدالة start اللي ببقا وارثها من كلاس Thread

```
// Create my thread class
class MyThread extends Thread {
    public void run() {
        System.out.println("MyThread is running");
    }
}

public class Threads_1 {
    public static void main(String[] args) {
        // Create my thread object
        MyThread mt = new MyThread();
        // Start my thread
        mt.start();
    }
}
```

- بس الطريقة دي مش مستحبة في معظم الأوقات لان في الجافا ماينفعلش تورث غير كلاس واحد ولما تورث الكلاس Thread كدة انت خلاص اخدت المرة اللي ليا وغالبا بنبقا محتاجين نورث كلاس ثاني.

2- وهنا يجي دور الطريقة الثانية وهيا الطريقة الأكثر استخداما، وهيا اني اعمل implement لل Runnable interface واللي فيها دالة واحدة محتاج اعملها Override وهيا دالة run والطريقة دي بتسمحلي اورث أي كلاس ثاني عادي وكمان في الجافا ممكن اعمل implements لاكثر من interface عادي فاكدة مش بخسر أي ميزة، وبعد ما اعمل الكلاس الجميل بتاعي واكتب الكود اللي انا عايزة يشتغل لما التريد يشتغل في دالة run، بعمل كائن من الكلاس اللي انا عملته وبعمل كائن من نوع Thread وبمرر الكائن اللي انا عملته من الكلاس بتاعي ك parameter لل Contractor بتاع كلاس Thread ، وكدة بقا معايا Thread فيه كل الكود بتاعي وفيه كل دوال كلاس Thread واللي منهم دالة start اللي بستعملها علشان اشغل التريد.

```
class MyThread implements Runnable {
    @Override
    public void run() {
        System.out.println("MyThread is running");
    }
}

public class Threads_2 {
    public static void main(String[] args) {
        // Create my thread object
        MyThread mt = new MyThread();
        // Start my thread
        Thread t = new Thread(mt);
        t.start();
    }
}
```

Important Thread class methods

هنا هشرح شوية دوال في الكلاس Thread واللي الدكتور ركز عليها:

METHOD	DISTRIBUTION
<code>public final void setName(String name)</code>	بنستعملها علشان نغير اسم الثريد
<code>public final String getName()</code>	بنستعملها علشان نعرف اسم الثريد
<code>public final void setPriority(int newPriority)</code>	بنستعملها علشان نحدد أولوية الثريد، والاولوية بتكون بين 1 و 10 والاولوية الافتراضية بتكون 5 لو ماغيرناهاش بالدالة دي، مع العلم انها مش بتأثر علي ترتيب تنفيذ الثريدث
<code>public final int getPriority()</code>	بنستعملها علشان نعرف أولوية الثريد واللي بتكون 5 لو ما غيرناهاش بدالة <code>setPriority</code>
<code>public static void sleep(long millis)</code> <code>throws InterruptedException</code>	بنستعملها علشان نوقف الثريد بشكل مؤقت وبتأخذ المدة اللي عايزين نوقف الثريد فيها وبتكن بالملي ثانية، وللازم نحطها في <code>try</code> و <code>catch</code> لانها بترمي اكسبشن من نوع <code>InterruptedException</code> لو ما عرفتتش تعمل <code>sleep</code> للثريد
<code>public void start()</code>	بنستعملها علشان نشغل الثريد بتاعنا

Thread class constants

كلاس Thread فيه 3 ثوابت بتمثلي ال priority وهيا:

public static final int MAX_PRIORITY	بيمثل اعلي قيمة لل priority واللي هيا 10
public static final int NORM_PRIORITY	بيمثل القيمة العادية لل priority واللي هيا 5 زي ما قولنا فوق، وممكن نقول انو بيمثل القيمة المتوسطة لل priority
public static final int MIN_PRIORITY	بيمثل اقل قيمة لل priority واللي هيا 1 زي ما قولنا في شرح دالة setPriority

Good luck <3

By [Anas Elgarhy](#)