

Глава 14

ПОТОКИ ВЫПОЛНЕНИЯ

Класс Thread и интерфейс Runnable

К большинству современных распределенных приложений (Rich Client) и Web-приложений (Thin Client) выдвигаются требования одновременной поддержки многих пользователей, каждому из которых выделяется отдельный поток, а также разделения и параллельной обработки информационных ресурсов. Потоки – средство, которое помогает организовать одновременное выполнение нескольких задач, каждую в независимом потоке. Потоки представляют собой классы, каждый из которых запускается и функционирует самостоятельно, автономно (или относительно автономно) от главного потока выполнения программы. Существуют два способа создания и запуска потока: расширение класса **Thread** или реализация интерфейса **Runnable**.

// пример # 1 : расширение класса Thread: Talk.java

```
package chapt14;
```

```
public class Talk extends Thread {  
    public void run() {  
        for (int i = 0; i < 8; i++) {  
            System.out.println("Talking");  
            try {  
                // остановка на 400 миллисекунд  
                Thread.sleep(400);  
            } catch (InterruptedException e) {  
                System.err.print(e);  
            }  
        }  
    }  
}
```

При реализации интерфейса **Runnable** необходимо определить его единственный абстрактный метод **run()**. Например:

/ пример # 2 : реализация интерфейса Runnable: Walk.java: WalkTalk.java */*

```
package chapt14;
```

```
public class Walk implements Runnable {  
    public void run() {  
        for (int i = 0; i < 8; i++) {  
            System.out.println("Walking");  
            try {  
                Thread.sleep(400);  
            }  
        }  
    }  
}
```

```

    } catch (InterruptedException e) {
        System.err.println(e);
    }
}
}
package chapt14;

public class WalkTalk {
    public static void main(String[] args) {
        // новые объекты потоков
        Talk talk = new Talk();
        Thread walk = new Thread(new Walk());
        // запуск потоков
        talk.start();
        walk.start();

        // Walk w = new Walk(); // просто объект, не поток
        // w.run(); // выполнится метод, но поток не запустится!
    }
}

```

Использование двух потоков для объектов классов **Talk** и **Walk** приводит к выводу строк: Talking Walking. Порядок вывода, как правило, различен при нескольких запусках приложения.

Жизненный цикл потока

При выполнении программы объект класса **Thread** может быть в одном из четырех основных состояний: “новый”, “работоспособный”, “неработоспособный” и “пассивный”. При создании потока он получает состояние “новый” (**NEW**) и не выполняется. Для перевода потока из состояния “новый” в состояние “работоспособный” (**RUNNABLE**) следует выполнить метод **start()**, который вызывает метод **run()** — основной метод потока.

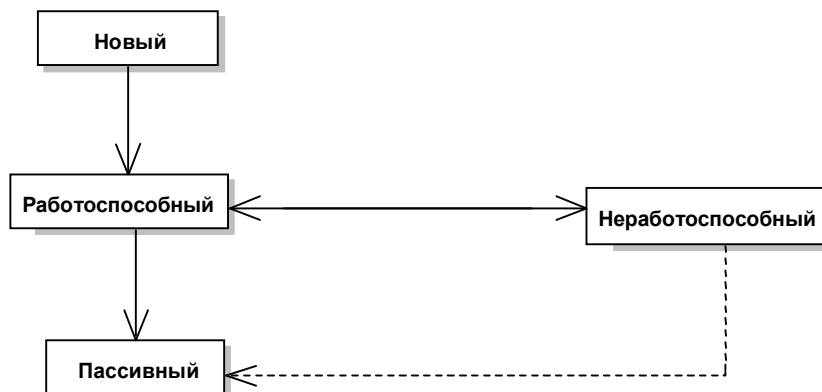


Рис. 14.1. Состояния потока