

### Prática 3 - Threads em Java

#### 1. Interface Runnable

- Classe Programa.java

```
/** A interface Runnable oferece um contrato que torna
 * a classe que a implementa executável.
 * Há a opção de estender a classe Thread. No entanto, herdamos
 * um monte de métodos e só implementamos o run.
 * Runnable estabelece que a classe deve implementar somente o run
 * tornando-a uma opção mais compacta e compatível com polimorfismo */

public class Programa implements Runnable {

    private int id;

    public void setId(int i){
        this.id = i;
    }

    public int getId(){
        return this.id;
    }

    public void run () {

        for (int i = 0; i < 1000; i++) {

            System.out.println("Programa " + id + " valor: " + i);

            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }

    }

}
```

- Classe OutroPrograma.java

```
/** A interface Runnable oferece um contrato que torna
 * a classe que a implementa executável.
 * Há a opção de estender a classe Thread. No entanto, herdamos
 * um monte de métodos e só implementamos o run.
 * Runnable estabelece que a classe deve implementar somente o run
 * tornando-a uma opção mais compacta e compatível com polimorfismo */
```

```

public class OutroPrograma implements Runnable {

    private int id;

    public void setId(int i){
        this.id = i;
    }

    public int getId(){
        return this.id;
    }

    public void run () {

        for (int i = 1000; i < 2000; i++) {

            System.out.println("Programa " + id + " valor: " + i);

            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }

    }

}

```

- Classe Teste.java

```

package sd.threads;

public class Teste {

    public static void main(String[] args) {

        Programa p1 = new Programa();

        p1.setId(1);

        Thread t1 = new Thread(p1);

        t1.start();

        Programa p2 = new Programa();

        p2.setId(2);

        Thread t2 = new Thread(p2);

        t2.start();

        OutroPrograma p3 = new OutroPrograma();

        p3.setId(3);
    }

}

```

```
        Thread t3 = new Thread(p3);  
        t3.start();  
        OutroPrograma p4 = new OutroPrograma();  
        p4.setId(4);  
        Thread t4 = new Thread(p4);  
        t4.start();  
    }  
}
```

### Exercícios:

1. Crie um projeto no Eclipse, codifique, execute e observe o funcionamento dessa aplicação com *threads*.
2. Desenvolva uma aplicação usando *threads* que realize duas tarefas distintas:
  1. Receber mensagem do usuário e armazenar esse conteúdo em um arquivo.
  2. Calcular e mostrar a soma dos números pares entre 1 e 1000.