UNIVERSIDADE FEDERAL TECNOLÓGICA DO PARANÁ

PÓS-GRADUAÇÃO LATO SENSU TECNOLOGIA JAVA

DISCENTE: MARLENE VASCONCELOS MORAES DE OLIVEIRA

1. Como o uso de várias *threads* Java permite escrever programas mais eficientes?

O uso de multithreading permite que o Sistema Operacional consiga dividir tarefas entre todos os processadores disponíveis, o que deixa o programa mais eficiente e aproveitamento do tempo ocioso para utilização através das threads.

2. Na criação de um objeto executável, por que pode ser melhor estender Thread em vez de implementar *Runnable*?

Apesar de ser vantajoso implementar a interface Runnable para estender outra classe, a classe Thread tem a vantagem de sobrepor somente um único método (run()).

3. Mostre como podemos usar *join*( ) para esperar um objeto de *thread* chamado MyThrd terminar.

```java
package thread;

class MyThrd extends Thread {
    public void run() {
        for (int i = 0; i < 5; i++) {
            try {
                Thread.sleep(1000);
                System.out.println("Thread em execução");
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class Main {
    public static void main(String[] args) {
        MyThrd t1 = new MyThrd();
        MyThrd t2 = new MyThrd();

        t1.start();
        t2.start();

        try {
            t1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("Programa finalizado.");
    }
}
```

4. Mostre como configurar uma *thread* chamada MyThrd com três níveis acima da prioridade normal.

```java
package thread;

class MyThrd extends Thread {
    public void run() {
        for (int i = 0; i < 5; i++) {
            try {
                Thread.sleep(1000);
                System.out.println("Thread em execução");
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class Main {
    public static void main(String[] args) {
        MyThrd t1 = new MyThrd();
        MyThrd t2 = new MyThrd();

        t1.setPriority(3);

        t1.start();
        t2.start();

        try {
            t1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("Programa finalizado.");
    }
}
```

5. Qual é o efeito da inclusão da palavra-chave *synchronized* em um método?

Synchronized permite apenas que uma thread por vez possa acessar o método, pois as threads, quando utilizam o mesmo método ao mesmo tempo, podem compartilhar variáveis, retornando resultados imprecisos.