

FACAMP – Linguagens de Programação

Aulas 9 e 10 – Exercícios

1. O que são threads? Quais são as vantagens do uso de threads?
2. Por que a execução de múltiplas threads pode gerar concorrência?
3. O que é uma seção crítica de um código?
4. O que são mutexes e como podem ser utilizados para proteger a concorrência?
5. O que são semáforos?
6. O que são monitores e como são implementados em Java?
7. O que são as Checked e as Unchecked Exceptions em Java?
8. Descreva como as exceções em Java podem ser: anunciadas, lançadas, obtidas, tratadas e repassadas.
9. Dada a implementação da classe abaixo:

```
public class Erros {
    public static void main(String[] args) {
        String s = null;
        int x[] = new int[3];
        try {
            for(int i = 0; i <=3; i++) {
                x[i] = i;
                System.out.println("Mensagem " + i);
            }
            System.out.println("Mensagem 4 " + s.substring(0,1));
        } catch (NullPointerException e) {
            System.out.println("Mensagem 5");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Mensagem 6");
        } finally {
            System.out.println("Mensagem 7");
        }
    }
}
```

- a. Que exceção em tempo de execução será inicialmente lançada pela JVM, quando a classe for executada? Por qual motivo?
 - b. E o que será impresso no console?
 - c. Se for corrigido esse erro de código, que outro erro será lançado pela JVM, em tempo de execução? Por qual motivo?
 - d. E o que será impresso no console desta vez?
10. No método abaixo, foram realizadas chamadas ao construtor e aos métodos write, flush e close, de FileWriter, que preveem uma checked exception do tipo IOException e, por isso, não pode ser compilado.

```
public class Questao9 {
    public void gravarArquivo(String arquivo, char[] dados){
        FileWriter writer = new FileWriter(arquivo);
        writer.write(dados);
        writer.flush();
        writer.close();
    }
    public static void main(String args[]){
        String nomeArquivo = "teste.txt";
    }
}
```

- a. Forneça duas soluções distintas para o problema:
 - i. Altere o método e insira o código para tratamento da exceção.
 - ii. Altere o método e insira o código para repassar a exceção.
 - b. Considere agora, para a primeira solução (em i), que o arquivo deva ser fechado nas duas situações: na ocorrência e na ausência da exceção. Modifique o método para implementar isso.
 - c. Agora, considerando a segunda solução (em ii), realize a chamada do método gravarArquivo no método main, passando o nomeArquivo fornecido.
11. Descreva, em linhas gerais, o modelo de eventos (Listeners) em Java.