Programação II – Prática 11

|  |  |
| --- | --- |
| **Ficha Teórica** | 9,10 e 11 |

# Exercício 1 Tratamento de Exceções ArithmeticException

# Faça um programa Java que solicite dois números ao utilizador e, em seguida, imprime o resultado da divisão do primeiro pelo segundo. Trate a exceção *ArithmeticException*. Teste o método *getMessage* e *printStackTrace* de *Exception* e observe o tipo de mensagem que retorna.

# Exercício 2: Excepções mais comuns

Segue abaixo algumas excepções comuns, para cada uma delas crie um exemplo próprio onde faz o seu devido uso.

**ClassCastException**

**IllegalArgumentException**

**NullPointerException**

**NumberFormatException**

**NegativeArraySizeException**

# Exercício 3: Tratamento de Exceções

Crie uma classe **Funcionário** que tem 2 subclasses **AlunoTrabalhador** e **Professor**. O professor tem um salario fixo positivo, enquanto o **alunoTrabalhador** recebe pelas horas que ele trabalha, o aluno trabalhador nunca poderá trabalhar horas negativas e também não poderá exceder 160horas.

# Exercício 4: Exceções Próprias

Refaça o exercício anterior mais não usando as exceções disponíveis na biblioteca do JAVA, crie suas propiás exceções que tem mensagens mais claras.

# Exercício 5: Exceções Próprias

Seja o trecho de programa ao lado. Suponha que o método saca da classe Conta vai ser rescrito de forma a lançar uma excepção criada por você cuja classe é ContaExcepcao. A exceção é lançada sempre que o saldo da conta for negativo. Implemente a classe ContaExcepcao. Implemente o método saca que lança a exceção. E rescreva o código abaixo com o devido tratamento de exceção.

Conta minhaConta = new Conta();

minhaConta.deposita(100);

minhaConta.setLimite(100);

minhaConta.saca(1000);

# Exercício 6: Exceções

O que é o bloco finally do comando try catch. Dê um exemplo em que este bloco deve ser utilizado.

**--- FIM DO DOCUMENTO ---**