

CC-02AN-DOO-Atividade-13-Exceptions-[Try-Catch-Finally]

Seu e-mail (pedro.costa4@uscsonline.com.br) foi registrado quando você enviou este formulário.

Questão 1 - Reproduzir o código abaixo, o que será exibido na console ao executar o código nas três condições:

- 1) com dois números inteiros.
- 2) com um frase "abc" e um número inteiro
- 3) com um número inteiro e uma frase "abc"

Cole o código e as mensagens. Explique com suas palavras o que ocorreu no caso de erro.

*

```
package material.excecao;

import java.util.Scanner;

/**
 * Classe que demonstra o uso do try / catch.
 */
public class ExemploTryCatch {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try {
            System.out.print("Digite um valor inteiro...");
            int numero1 = s.nextInt();
            System.out.print("Digite um valor inteiro...");
            int numero2 = s.nextInt();

            System.out.println(numero1 + " + " + numero2 + " = " + (numero1+numero2));
        } catch (Exception ex) {
            System.out.println("ERRO - Valor digitado nao e um numero inteiro!");
        }
    }
}
```

Exemplo do bloco try-catch

Digite um valor inteiro...:5	Digite um valor inteiro...:10
Digite um valor inteiro...:3	Digite um valor inteiro...:abc
5 + 3 = 8	ERRO - Valor digitado nao e um numero inteiro!

Código:

```
package uscs;
import java.util.Scanner;
public class ExemploTryCatch {
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        try{
            System.out.println("Digite um valor inteiro...");
            int numero1 = s.nextInt();
            System.out.println("Digite um valor inteiro...");
            int numero2 = s.nextInt();
```

```
        System.out.println(numero1+" + "+numero2+" = "+(numero1+numero2));
    } catch (Exception ex){
        System.out.println("ERRO - Valor digitado nao e um numero inteiro");
    }
}
}
```

1) Mensagem para execução com dois números inteiros:

Digite um valor inteiro...:

10

Digite um valor inteiro...:

4

10 + 4 = 14

2) Mensagem para execução com um frase "abc" e um número inteiro:

Digite um valor inteiro...:

abc

ERRO - Valor digitado nao e um numero inteiro

3) Mensagem para execução com um número inteiro e uma frase "abc"

Digite um valor inteiro...:

10

Digite um valor inteiro...:

abc

ERRO - Valor digitado nao e um numero inteiro

Após a execução do exercício, é possível concluir que, assim que o bloco "Try-Catch" encontra um erro, ele captura a exceção e exibe a mensagem de erro definida. Se o usuário digitar um valor que não seja um número inteiro e tentar atribuí-lo a uma variável declarada como "int", o bloco "Try" será encerrado e o erro será mostrado ao usuário. Se o primeiro valor digitado for um número inteiro válido, o programa prossegue para a próxima etapa, mas caso contrário, o processo é interrompido e a mensagem de erro é exibida.

Questão 2 - Reproduzir o código abaixo, o que será exibido na console.

1) Entre com uma senha alfabetica qualquer

2) Entre com a senha numerica definidad 123456

Cole o código e as mensagens. Explique com suas palavras o que ocorreu no caso de erro.

*

Exemplo de Lançamento de uma Exceção

throw

```
/**
 * Classe utilizada para demonstrar o uso da palavra chave throw,
 * utilizada quando queremos lançar uma exceção.
 */
public class ExemploThrow {
    public static final String SENHASECRETA = "123456";

    public static void main(String[] args) {
        try {
            Scanner s = new Scanner(System.in);
            System.out.print("Digite a senha: ");
            String senha = s.nextLine();
            if(!senha.equals(SENHASECRETA)) {
                throw new Exception("Senha invalida!!!");
            }
            System.out.println("Senha correta!!!\nBem vindo(a)!!!");
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

Digite a senha: abc
Senha invalida!!!

Digite a senha: 123456
Senha correta!!!
Bem vindo(a)!!!

Código:

```
package uscs;
import java.util.Scanner;
public class ExemploThrow {
    public static final String SENHASECRETA = "123456";

    public static void main(String[] args){
        try{
            Scanner s = new Scanner(System.in);
            System.out.print("Digite a senha: ");
            String senha = s.nextLine();
            if(!senha.equals(SENHASECRETA)){
                throw new Exception("Senha invalida!!!");
            }
            System.out.println("Senha correta!!!\nBem-vindo(a)!!!");
        } catch(Exception ex){
            System.out.println(ex.getMessage());
        }
    }
}
```

1) Mensagem para entrada com uma senha alfabética qualquer:

Digite a senha: abcdefg
Senha invalida!!!

1) Mensagem para entrada com a senha numérica definida "123456":

Digite a senha: 123456
Senha correta!!!
Bem-vindo(a)!!!

Após a execução do código e do exercício de teste, foi possível entender na prática para que serve a funcionalidade "throw new Exception", que personaliza uma nova exceção de erro durante um bloco "Try-Catch". Dessa forma, nesse código em específico, foi realizada uma

verificação "If", que caso a variável senha (que armazenou o valor digitado pelo usuário), fosse diferente do valor definido anteriormente, uma mensagem informando o erro de "senha inválida" seria anunciada e o bloco seria encerrado imediatamente, e caso fossem iguais, o bloco seguiria normalmente. Logo, é possível concluir que essa função permite personalizar diversas mensagens informando os erros específicos ao longo de blocos Try.

Questão 3 - Reproduzir o código abaixo, o que será exibido na console.

- 1) Insira dois números inteiros
- 2) Insira um número inteiro e um zero

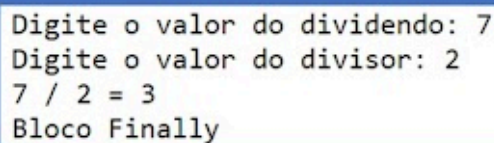
Cole o código e as mensagens. Explique com suas palavras o que ocorreu no caso de erro.

*

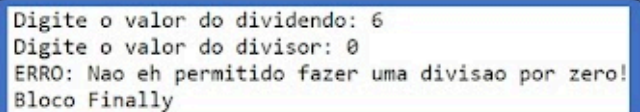
```
public class ExemploFinally {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try {
            int dividendo, divisor;
            System.out.print("Digite o valor do dividendo: ");
            dividendo = s.nextInt();
            System.out.print("Digite o valor do divisor: ");
            divisor = s.nextInt();

            if(divisor == 0) {
                throw new Exception("Nao eh permitido fazer uma divisao por zero!");
            }

            System.out.println(dividendo+" / "+divisor+" = "+(dividendo / divisor));
        } catch (Exception ex) {
            System.out.println("Erro: " + ex.getMessage());
        } finally {
            System.out.println("Bloco Finally.");
        }
    }
}
```



```
Digite o valor do dividendo: 7
Digite o valor do divisor: 2
7 / 2 = 3
Bloco Finally
```



```
Digite o valor do dividendo: 6
Digite o valor do divisor: 0
ERRO: Nao eh permitido fazer uma divisao por zero!
Bloco Finally
```

Código:

```
package uscs;
import java.util.Scanner;
public class ExemploFinally {
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        try{
            int dividendo, divisor;
            System.out.print("Digite o valor do dividendo: ");
            dividendo = s.nextInt();
            System.out.print("Digite o valor do divisor: ");
            divisor = s.nextInt();

            if(divisor == 0){
                throw new Exception("Nao e permitido fazer uma divisao por zero!");
            }

            System.out.println(dividendo+" / "+divisor+" = "+(dividendo/divisor));

        } catch(Exception ex){
            System.out.println("Erro: " + ex.getMessage());
        } finally{
```

```
        System.out.println("Bloco Finally.");  
    }  
}  
}
```

1) Mensagem para entrada com dois números inteiros:

Digite o valor do dividendo: 10

Digite o valor do divisor: 2

10 / 2 = 5

Bloco Finally.

2) Mensagem para entrada com um número inteiro e um zero:

Digite o valor do dividendo: 10

Digite o valor do divisor: 0

Erro: Não é permitido fazer uma divisão por zero!

Bloco Finally.

Após concluir o exercício, foi possível perceber na prática a funcionalidade de "Finally" em um bloco Try, que permite emitir uma mensagem de finalização do bloco, que é sempre emitida ao finalizar os processos do bloco, independente se ocorreu tudo certo ou se houve alguma "Exception". Assim como no exercício anterior, dentro do bloco Try foi desenvolvida uma verificação IF, onde caso o valor de divisor fosse igual a zero, o bloco seria interrompido com uma mensagem específica para esse erro, o que acontece antes da tentativa de dividir com os valores fornecidos pelo usuário.