

UPskill – Java+.NET

Programação Orientada a Objetos – Exceções

EXERCÍCIO FUNCIONÁRIO

1. Crie um projeto JAVA com o nome **prj_Exc1** e um package designado **pkg_Exc1**, ao qual deverá adicionar todas as classes existentes no fim deste enunciado. Em seguida, resolva cada uma das seguintes questões:
 - a. Compile e teste o código disponibilizado. Qual a razão de só ser mostrada a informação dos dois primeiros funcionários criados no método *main()*?
 - b. Considerando que também só são aceites vencimentos cujo valor seja um valor positivo, faça as alterações ao código com vista a validar os vencimentos dos funcionários, seguindo a mesma abordagem da alínea anterior.
2. Adicione um package com o nome **pkg_Exc1a** ao projeto da pergunta anterior. Teste cada um dos exemplos que se encontram no Moodle. Retire conclusões sobre a utilidade do tratamento das exceções em Java.

Código da classe **ExcecaoAlfabetoEspaco**:

```
public class ExcecaoAlfabetoEspaco extends Exception {
    public ExcecaoAlfabetoEspaco() {
        super();
    }
    public ExcecaoAlfabetoEspaco(String msg) {
        super(msg);
    }
}
```

Código da classe **Funcionario1**:

```
public class Funcionario1 {
    private String nome;
    private float vencimento;

    private static final String NOME_POR_OMISSAO = "sem nome";
    private static final float VENC_POR_OMISSAO = 500;

    public Funcionario1() throws ExcecaoAlfabetoEspaco {
        setNome(NOME_POR_OMISSAO);
        setVencimento(VENC_POR_OMISSAO);
    }

    public Funcionario1(String nome, float vencimento) throws ExcecaoAlfabetoEspaco {
        setNome(nome);
        setVencimento(vencimento);
    }
}
```

```
public String getNome() {
    return nome;
}
public float getVencimento() {
    return vencimento;
}

public void setNome(String nome) throws ExcecaoAlfabetoEspaco{
    char c;
    for(int i=0;i<nome.length();i++){
        c = nome.charAt(i);
        if (Character.isLetter(c)== false && Character.isSpaceChar(c)== false)
            throw new ExcecaoAlfabetoEspaco("O " + nome + " tem caracteres que não são
letras");
    }
    this.nome = nome;
}

public void setVencimento(float vencimento) {
    this.vencimento = vencimento;
}

public String toString() {
    return nome + " tem o vencimento de " + vencimento;
}

@Override
public boolean equals(Object outroObjeto) {
    if (this == outroObjeto) {
        return true;
    }
    if (outroObjeto == null || getClass() != outroObjeto.getClass()) {
        return false;
    }
    Funcionario1 obj = (Funcionario1) outroObjeto;
    return this.nome.equalsIgnoreCase(obj.nome) &&
        this.vencimento == obj.vencimento;
}
}
```

Código da classe **Teste1**:

```
public class Teste1 {
    public static void main(String[] args) {
        Funcionario1 f1=null;
        Funcionario1 f2=null;
        Funcionario1 f3=null;
        Funcionario1 f4=null;

        // Validar se os nomes têm só letras ou o carácter espaço
        try {
            f1 = new Funcionario1("Pedro Silva", 2000);
            f2 = new Funcionario1("Ana Gomes", 3500);
            f3 = new Funcionario1("Paulo Barros", 900);
            f4 = new Funcionario1("Catarina Machado", 1500);
        } catch (ExcecaoAlfabetoEspaco e){
            System.out.println(e.getMessage());
        }
        System.out.println("Listar funcionarios");
        if (f1!= null)
            System.out.println("f1 -> "+f1.toString());
        if (f2!= null)
            System.out.println("f2 -> "+f2.toString());
        if (f3!= null)
            System.out.println("f3 -> "+f3.toString());
        if (f4!= null)
            System.out.println("f4 -> "+f4.toString());
    }
}
```