

PRÁTICA LABORATORIAL JUNIT5

Objetivos:

- Conceitos de Software Quality Assurance
- Testes em Java
- Gradle
- Junit5

EXERCÍCIOS

Parte 1

1. Tenha em consideração o seguinte excerto de código que representa um sistema de gestão de funcionários em Java. Contém funções de calcular o salário. Deve ser criada uma classe de testes para cada uma destas funções.

```
public class Employee {
    private String name;
    private String position;
    private int hoursWorked;
    private double hourlyRate;

    public Employee(String name, String position, double hourlyRate) {
        this.name = name;
        this.position = position;
        this.hourlyRate = hourlyRate;
        this.hoursWorked = 0;
    }

    public void setHoursWorked(int hoursWorked) {
        this.hoursWorked = hoursWorked;
    }

    public double calculateSalary() {
        double salary = 0.0;

        if (position.equals("Manager")) {
            salary = hourlyRate * hoursWorked;
        } else if (position.equals("Developer")) {
            salary = hourlyRate * hoursWorked * 1.2;
        } else if (position.equals("Intern")) {
            salary = hourlyRate * hoursWorked * 0.8;
        }

        return salary;
    }
}
```

2. Seguidamente, tenha em conta a seguinte classe Enterprise que terá uma ArrayList com os funcionários todos, assim como um método que permite calcular os salários totais. Crie a classe de teste necessária, e, caso algo não esteja correto, implemente uma solução.

```
import java.util.ArrayList;
import java.util.List;

public class Enterprise {
    private ArrayList<Employee> employees;

    public Enterprise() {
        employees = new ArrayList<>();
    }

    public void addEmployee(Employee employee) {
        employees.add(employee);
    }

    public void removeEmployee(Employee employee) {
        employees.remove(employee);
    }

    public double calculateTotalSalaries() {
        double totalSalaries = 0.0;
        for (Employee employee : employees) {
            totalSalaries = employee.calculateSalary();
        }
        return totalSalaries;
    }
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Enterprise enterprise = new Enterprise();  
  
        Employee manager = new Employee("Diamantino Pereira da Silva", "Manager", 20.0);  
        Employee developer = new Employee("Roberto Sem Esperança", "Developer", 15.0);  
        Employee intern = new Employee("Jacinto Penedo", "Intern", 10.0);  
  
        enterprise.addEmployee(manager);  
        enterprise.addEmployee(developer);  
        enterprise.addEmployee(intern);  
  
        double totalSalaries = enterprise.calculateTotalSalaries();  
        System.out.println("Total salaries: " + totalSalaries);  
    }  
}  
  
public void removeEmployee(Employee employee) {  
    employees.remove(employee);  
}  
  
public double calculateTotalSalaries() {  
    double totalSalaries = 0.0;  
    for (Employee employee : employees) {  
        totalSalaries = employee.calculateSalary();  
    }  
    return totalSalaries;  
}  
}
```

3. Reveja um programa que tem uma classe chamada ano, que tem um atributo inteiro ano e um array de classes meses, podendo ter 12 no máximo. Seguidamente, a classe meses tem o atributo de nome, e tem um array de tamanho fixo 31 que tenha a classe dias. A classe dias tem um atributo inteiro chamado dia que pode armazenar apenas de 1 a 31, assim como um arraylist de Strings chamado tarefas.

```
import java.util.ArrayList;

class Dia {
    private int dia;
    private ArrayList<String> tarefas;

    public Dia(int dia) {
        this.dia = dia;
        this.tarefas = new ArrayList<>();
    }

    public int getDia() {
        return dia;
    }

    public ArrayList<String> getTarefas() {
        return tarefas;
    }

    public void adicionarTarefa(String tarefa) {
        tarefas.add(tarefa);
    }
}

class Mes {
    private String nome;
    private Dia[] dias;

    public Mes(String nome) {
        this.nome = nome;
        this.dias = new Dia[31];
        for (int i = 0; i < 31; i++) {
            dias[i] = new Dia(i + 1);
        }
    }

    public String getNome() {
        return nome;
    }

    public Dia[] getDias() {
        return dias;
    }
}
```

```
class Ano {
    private int ano;
    private Mes[] meses;

    public Ano(int ano) {
        this.ano = ano;
        this.meses = new Mes[12];
        for (int i = 0; i < 12; i++) {
            meses[i] = new Mes(obterNomeMes(i + 1));
        }
    }

    public int getAno() {
        return ano;
    }

    public Mes[] getMeses() {
        return meses;
    }

    private String obterNomeMes(int mes) {
        switch (mes) {
            case 1:
                return "Janeiro";
            case 2:
                return "Fevereiro";
            case 3:
                return "Março";
            case 4:
                return "Abril";
            case 5:
                return "Maio";
            case 6:
                return "Junho";
            case 7:
                return "Julho";
            case 8:
                return "Agosto";
            case 9:
                return "Setembro";
            case 10:
                return "Outubro";
            case 11:
                return "Novembro";
            case 12:
                return "Dezembro";
            default:
                return "";
        }
    }
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Ano ano = new Ano(2023);  
        Mes[] meses = ano.getMeses();  
        Mes primeiroMes = meses[0];  
        Dia[] dias = primeiroMes.getDias();  
        Dia primeiroDia = dias[0];  
        primeiroDia.adicionarTarefa("Consulta Ortopedista");  
        primeiroDia.adicionarTarefa("Entrega Trabalho Prático de QA");  
  
        System.out.println("Ano: " + ano.getAno());  
        for (Mes mes : meses) {  
            System.out.println("Mês: " + mes.getNome());  
            for (Dia dia : mes.getDias()) {  
                System.out.println("Dia: " + dia.getDia());  
                System.out.println("Tarefas:");  
                for (String tarefa : dia.getTarefas()) {  
                    System.out.println("- " + tarefa);  
                }  
            }  
        }  
    }  
}
```

Analise se as classes têm todas as funções bem implementadas através de testes unitários.

Bom trabalho! 😊