



Universidade São Tomás de Moçambique

Nome:

António Manuel Siteo

Amade Netelagegue

Kelvin Mabasso

Turma: 4P7LDS1

Curso: LDS

Crie uma classe de nome Funcionário, com os seguintes atributos: código, nome, contacto, departamento, género, estadoCivil, dias trabalhados e o salário diário. Para além dos métodos de acesso e de modificação, a classe deve implementar o método toString() e um método que devolve valor para calcular o salário do funcionário. O método toString deve implementar o método que calcula o salário

Projecto 1-CRUD com ArrayList e ficheiro de texto;

ViewFuncionario.java

```
import java.io.IOException;
import java.util.Scanner;
public class ViewFuncionario {
    public static void main(String[] args) throws IOException {
        Scanner k = new Scanner(System.in);
        int opcao;
        do {
            System.out.println("\n=== MENU FUNCIONÁRIO ===");
            System.out.println("1 - Adicionar Funcionário");
            System.out.println("2 - Listar Funcionários");
            System.out.println("3 - Actualizar Funcionário");
            System.out.println("4 - Remover Funcionário");
            System.out.println("5 - Ordenar Funcionários");
            System.out.println("0 - Sair");
            System.out.print("Opção: ");
            opcao = k.nextInt();

            switch (opcao) {
                case 1:
                    ControllerFuncionario.adicionarFuncionario();
                    break;
                case 2:
                    ControllerFuncionario.listarFuncionarios();
                    break;
                case 3:
                    System.out.print("Digite o código do funcionário: ");
                    int codAtualizar = k.nextInt();
                    ControllerFuncionario.actualizarFuncionario(codAtualizar);
                    break;
                case 4:
                    System.out.print("Digite o código do funcionário: ");
                    int codRemover = k.nextInt();
                    ControllerFuncionario.removerFuncionario(codRemover);
                    break;
                case 5:
                    ControllerFuncionario.ordenarFuncionarios();
                    break;
                case 0:
                    System.out.println("Encerrando...");
                    break;
                default:
                    System.out.println("Opção inválida!");
            }
        } while (opcao != 0);
    }
}
```

Funcionario.java

```
public class Funcionario {
    private int codigo;
    private String nome;
    private String contacto;
    private String departamento;
    private String genero;
    private String estadoCivil;
    private int diasTrabalhados;
    private double salarioDiario;

    public Funcionario() {
    }

    public Funcionario(int codigo, String nome, String contacto, String departamento, String
genero,
        String estadoCivil, int diasTrabalhados, double salarioDiario) {
        this.codigo = codigo;
        this.nome = nome;
        this.contacto = contacto;
        this.departamento = departamento;
        this.genero = genero;
        this.estadoCivil = estadoCivil;
        this.diasTrabalhados = diasTrabalhados;
        this.salarioDiario = salarioDiario;
    }

    // Getters e Setters
    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getContacto() {
        return contacto;
    }

    public void setContacto(String contacto) {
        this.contacto = contacto;
    }

    public String getDepartamento() {
        return departamento;
    }

    public void setDepartamento(String departamento) {
```

```

        this.departamento = departamento;
    }

    public String getGenero() {
        return genero;
    }

    public void setGenero(String genero) {
        this.genero = genero;
    }

    public String getEstadoCivil() {
        return estadoCivil;
    }

    public void setEstadoCivil(String estadoCivil) {
        this.estadoCivil = estadoCivil;
    }

    public int getDiasTrabalhados() {
        return diasTrabalhados;
    }

    public void setDiasTrabalhados(int diasTrabalhados) {
        this.diasTrabalhados = diasTrabalhados;
    }

    public double getSalarioDiario() {
        return salarioDiario;
    }

    public void setSalarioDiario(double salarioDiario) {
        this.salarioDiario = salarioDiario;
    }

    // Método que calcula o salário total
    public double calcularSalario() {
        return diasTrabalhados * salarioDiario;
    }

    // toString incluindo o cálculo do salário
    @Override
    public String toString() {
        return codigo + "-" + nome + "-" + contacto + "-" + departamento + "-" +
            genero + "-" + estadoCivil + "-" + diasTrabalhados + "-" +
            salarioDiario + "-" + calcularSalario();
    }
}

```

ControllerFuncionario.java

```
import java.io.*;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class ControllerFuncionario {
    static Scanner k = new Scanner(System.in);

    public static void adicionarFuncionario() throws IOException {
        FileWriter fw = new FileWriter("funcionarios.txt", true);
        BufferedWriter bw = new BufferedWriter(fw);

        System.out.println("Codigo: ");
        int codigo = Integer.parseInt(k.next());

        System.out.println("Nome: ");
        String nome = k.next();

        System.out.println("Contacto: ");
        String contacto = k.next();

        System.out.println("Departamento: ");
        String departamento = k.next();

        System.out.println("Gênero: ");
        String genero = k.next();

        System.out.println("Estado Civil: ");
        String estadoCivil = k.next();

        System.out.println("Dias Trabalhados: ");
        int dias = k.nextInt();

        System.out.println("Salário Diário: ");
        double salarioDiario = k.nextDouble();

        Funcionario funcionario = new Funcionario(codigo, nome, contacto, departamento, genero,
            estadoCivil, dias,
            salarioDiario);

        bw.write(funcionario.toString());
        bw.newLine();
        bw.close();
        fw.close();
        System.out.println("FUNCIONÁRIO ADICIONADO COM SUCESSO!");
    }

    public static ArrayList<Funcionario> listaFuncionarios() throws IOException {
        ArrayList<Funcionario> funcionarios = new ArrayList<>();
        FileReader fr = new FileReader("funcionarios.txt");
        BufferedReader br = new BufferedReader(fr);
        String linha;

        while ((linha = br.readLine()) != null && !linha.isEmpty()) {
            String[] dados = linha.split("-");
            if (dados.length ≥ 8) {
```



```

        System.out.print("Novo Departamento: ");
        f.setDepartamento(k.next());
        break;
    case '4':
        System.out.print("Novo Gênero: ");
        f.setGenero(k.next());
        break;
    case '5':
        System.out.print("Novo Estado Civil: ");
        f.setEstadoCivil(k.next());
        break;
    case '6':
        System.out.print("Novos Dias Trabalhados: ");
        f.setDiasTrabalhados(k.nextInt());
        break;
    case '7':
        System.out.print("Novo Salário Diário: ");
        f.setSalarioDiario(k.nextDouble());
        break;
    case '0':
        break;
    default:
        System.out.println("Opção inválida!");
    }
} while (opcao != '0');
}
}

FileWriter fw = new FileWriter("funcionarios.txt");
BufferedWriter bw = new BufferedWriter(fw);
for (Funcionario f : funcionarios) {
    bw.write(f.toString());
    bw.newLine();
}
bw.close();
fw.close();

if (encontrado) {
    System.out.println("FUNCIONÁRIO ACTUALIZADO COM SUCESSO!");
} else {
    System.out.println("FUNCIONÁRIO NÃO ENCONTRADO!");
}
}

public static void removerFuncionario(int codigo) throws IOException {
    ArrayList<Funcionario> funcionarios = listaFuncionarios();
    boolean removido = funcionarios.removeIf(f → f.getCodigo() == codigo);

    FileWriter fw = new FileWriter("funcionarios.txt");
    BufferedWriter bw = new BufferedWriter(fw);
    for (Funcionario f : funcionarios) {
        bw.write(f.toString());
        bw.newLine();
    }
    bw.close();
    fw.close();
}

```

```

        if (removido) {
            System.out.println("FUNCIONÁRIO REMOVIDO COM SUCESSO!");
        } else {
            System.out.println("FUNCIONÁRIO NÃO ENCONTRADO!");
        }
    }
}

public static void ordenarFuncionarios() throws IOException {
    ArrayList<Funcionario> funcionarios = listaFuncionarios();
    Collections.sort(funcionarios, (f1, f2) → Integer.compare(f1.getCodigo(),
f2.getCodigo()));

    FileWriter fw = new FileWriter("funcionarios.txt");
    BufferedWriter bw = new BufferedWriter(fw);
    for (Funcionario f : funcionarios) {
        bw.write(f.toString());
        bw.newLine();
    }
    bw.close();
    fw.close();
    System.out.println("FUNCIONÁRIOS ORDENADOS COM SUCESSO!");
}
}

```

Projecto 2-CRUD com ArrayList e ficheiro de objecto;

ViewFuncionario, java

```

import java.util.Scanner;
public class ViewFuncionario {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) throws Exception {
        int opcao;
        do {
            System.out.println("++ MENU FUNCIONÁRIO ++");
            System.out.println("1 - Adicionar");
            System.out.println("2 - Listar");
            System.out.println("3 - Actualizar");
            System.out.println("4 - Remover");
            System.out.println("5 - Ordenar");
            System.out.println("0 - Sair");
            System.out.print("Opção: ");
            opcao = sc.nextInt();
            switch (opcao) {
                case 1 → ControllerFuncionario.adicionarFuncionario();
                case 2 → ControllerFuncionario.listarFuncionarios();
                case 3 → {
                    System.out.print("Código do funcionário: ");
                    ControllerFuncionario.actualizarFuncionario(sc.nextInt());}
                case 4 → {
                    System.out.print("Código do funcionário: ");
                    ControllerFuncionario.removerFuncionario(sc.nextInt());
                }
                case 5 → ControllerFuncionario.ordenarFuncionarios();
                case 0 → System.out.println("SAIU DO MENU.");
                default → System.out.println("Opção inválida.");
            }
        } while (opcao ≠ 0);
    }
}

```


Funcionario.java

```
import java.io.Serializable;

public class Funcionario implements Comparable<Funcionario>, Serializable {
    private int codigo;
    private String nome;
    private String contacto;
    private String departamento;
    private String genero;
    private String estadoCivil;
    private int diasTrabalhados;
    private double salarioDiario;
    public Funcionario(int codigo, String nome, String contacto, String departamento,
        String genero, String estadoCivil, int diasTrabalhados, double salarioDiario) {
        this.codigo = codigo;
        this.nome = nome;
        this.contacto = contacto;
        this.departamento = departamento;
        this.genero = genero;
        this.estadoCivil = estadoCivil;
        this.diasTrabalhados = diasTrabalhados;
        this.salarioDiario = salarioDiario;
    }
    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getContacto() {
        return contacto;
    }
    public void setContacto(String contacto) {
        this.contacto = contacto;
    }
    public String getDepartamento() {
        return departamento;
    }
    public void setDepartamento(String departamento) {
        this.departamento = departamento;
    }
    public String getGenero() {
        return genero;
    }
    public void setGenero(String genero) {
        this.genero = genero;
    }
    public String getEstadoCivil() {
        return estadoCivil;
    }
}
```

```

public void setEstadoCivil(String estadoCivil) {
    this.estadoCivil = estadoCivil;
}
public int getDiasTrabalhados() {
    return diasTrabalhados;
}
public void setDiasTrabalhados(int diasTrabalhados) {
    this.diasTrabalhados = diasTrabalhados;
}
public double getSalarioDiario() {
    return salarioDiario;
}
public void setSalarioDiario(double salarioDiario) {
    this.salarioDiario = salarioDiario;
}
public double calcularSalario() {
    return diasTrabalhados * salarioDiario;
}
public int compareTo(Funcionario outro) {
    return Integer.compare(this.codigo, outro.codigo);
}
public String toString() {
    return codigo + " - " + nome + " - " + contacto + " - " + departamento + " - " +
        genero + " - " + estadoCivil + " - " + diasTrabalhados + " dias - Salário: " +
        calcularSalario();
}
}

```

ControllerFuncionario.java

```

import java.io.*;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class ControllerFuncionario {
    static Scanner k = new Scanner(System.in);

    public static ArrayList<Funcionario> lerFuncionarios() {
        ArrayList<Funcionario> lista = new ArrayList<>();

        try (ObjectInputStream ler = new ObjectInputStream(new
        FileInputStream("funcionarios.bin"))) {
            while (true) {
                lista.add((Funcionario) ler.readObject());
            }
        } catch (EOFException e) {
            // fim do ficheiro
        } catch (Exception e) {
            System.out.println("Erro ao ler: " + e.getMessage());
        }

        return lista;
    }
}

```

```

    public static void salvarFuncionarios(ArrayList<Funcionario> funcionarios) throws
IOException {
        try (ObjectOutputStream escrever = new ObjectOutputStream(new
FileOutputStream("funcionarios.bin"))) {
            for (Funcionario f : funcionarios) {
                if (f != null) {
                    escrever.writeObject(f);
                }
            }
        }
    }

    public static void adicionarFuncionario() throws IOException {
        ArrayList<Funcionario> lista = lerFuncionarios();

        System.out.print("Código: ");
        int codigo = k.nextInt();

        System.out.print("Nome: ");
        String nome = k.next();

        System.out.print("Contacto: ");
        String contacto = k.next();

        System.out.print("Departamento: ");
        String departamento = k.next();

        System.out.print("Gênero: ");
        String genero = k.next();

        System.out.print("Estado Civil: ");
        String estadoCivil = k.next();

        System.out.print("Dias Trabalhados: ");
        int dias = k.nextInt();

        System.out.print("Salário Diário: ");
        double salario = k.nextDouble();

        Funcionario novo = new Funcionario(codigo, nome, contacto, departamento, genero,
estadoCivil, dias, salario);
        lista.add(novo);

        salvarFuncionarios(lista);
        System.out.println("FUNCIONÁRIO ADICIONADO!");
    }

    public static void listarFuncionarios() {
        ArrayList<Funcionario> lista = lerFuncionarios();

        if (lista.isEmpty()) {
            System.out.println("Nenhum funcionário encontrado.");
        } else {
            for (Funcionario f : lista) {
                System.out.println(f);
            }
        }
    }

```

```

}

public static void atualizarFuncionario(int codigo) throws IOException {
    ArrayList<Funcionario> lista = lerFuncionarios();

    for (Funcionario f : lista) {
        if (f != null && f.getCodigo() == codigo) {
            System.out.println("1-Nome\n2-Contacto\n3-Departamento\n4-Gênero\n5-Estado\n6-Dias\n7-Salário\n8-Sair");
            char opcao;
            do {
                System.out.print("Opção: ");
                opcao = k.next().charAt(0);
                switch (opcao) {
                    case '1':
                        System.out.print("Nome: ");
                        f.setNome(k.next());
                        break;
                    case '2':
                        System.out.print("Contacto: ");
                        f.setContacto(k.next());
                        break;
                    case '3':
                        System.out.print("Departamento: ");
                        f.setDepartamento(k.next());
                        break;
                    case '4':
                        System.out.print("Gênero: ");
                        f.setGenero(k.next());
                        break;
                    case '5':
                        System.out.print("Estado Civil: ");
                        f.setEstadoCivil(k.next());
                        break;
                    case '6':
                        System.out.print("Dias: ");
                        f.setDiasTrabalhados(k.nextInt());
                        break;
                    case '7':
                        System.out.print("Salário: ");
                        f.setSalarioDiario(k.nextDouble());
                        break;
                }
            } while (opcao != '8');
        }
    }

    salvarFuncionarios(lista);
    System.out.println("FUNCIONÁRIO ACTUALIZADO.");
}

public static void removerFuncionario(int codigo) throws IOException {
    ArrayList<Funcionario> lista = lerFuncionarios();

    lista.removeIf(f -> f != null && f.getCodigo() == codigo);

    salvarFuncionarios(lista);
}

```

```

        System.out.println("FUNCIONÁRIO REMOVIDO.");
    }

    public static void ordenarFuncionarios() throws IOException {
        ArrayList<Funcionario> lista = lerFuncionarios();
        Collections.sort(lista);
        salvarFuncionarios(lista);
        System.out.println("FUNCIONÁRIOS ORDENADOS.");
    }
}

```

OUTPUT

```

○ tony@pop-os:~/Documents/GitHub/P00/GRUP0/projeto2$ /usr/bin/env /usr/lib/jvm/java-21
sages -cp /home/tony/.config/Code/User/workspaceStorage/572054b30149b300f550893fd955f9
io
++ MENU FUNCIONÁRIO ++
1 - Adicionar
2 - Listar
3 - Atualizar
4 - Remover
5 - Ordenar
0 - Sair
Opção: 1
Código: 123
Nome: Antonio
Contacto: 878984953
Departamento: IT
Gênero: Masculino
Estado Civil: Solteiro
Dias Trabalhados: 4
Salário Diário: 40500
FUNCIONÁRIO ADICIONADO!
++ MENU FUNCIONÁRIO ++
1 - Adicionar
2 - Listar
3 - Atualizar
4 - Remover
5 - Ordenar
0 - Sair
Opção: ^[[21~^[[21~^[[21~

```

Adicionar funcionario

Opção: 2

123 - Antonio - 876754 - Maputo - Masculino - Solteiro - 5 dias - Salário: 1715.0

123 - Antonio - 878984953 - IT - Masculino - Solteiro - 4 dias - Salário: 162000.0

++ MENU FUNCIONÁRIO ++

1 - Adicionar

2 - Listar

3 - Actualizar

4 - Remover

5 - Ordenar

0 - Sair

Opção: ^[[21~

Listagem de Funcionario