

**UNIVERSIDADE PRESBITERIANA MACKENZIE**  
**CIÊNCIA DA COMPUTAÇÃO (2023.2)**  
**ESTRUTURA DE DADOS II – 04D**

**Revisão: Java e Orientação a Objetos**

NOME	TIA
Jonatas Garcia de Oliveira	42181232
Pedro Henrique Araujo Farias	32246994

**EXERCÍCIO 01**

**Código-fonte do arquivo *Avaliacao.java***

```
public class Avaliacao {
    String nome;
    String tipoAvaliacao;
    float notaMaxima;

    // Item A
    public Avaliacao() {
        this("", "", 0.0f);
    }

    public Avaliacao(String nameArg, String typeArg, float maxGradeArg) {
        this.nome = nameArg;
        this.tipoAvaliacao = typeArg;
        this.notaMaxima = maxGradeArg;
    }

    // Item B
    public float getFinalPercent() {
        return notaMaxima * 10;
    }

    // Item C
    @Override
    public String toString() {
        return "Nome: " + nome + " | Tipo: " + tipoAvaliacao + " | Nota
máxima: " +
        notaMaxima + " | Percentual final: " + getFinalPercent() + "% ";
    }
}
```

## Código-fonte do arquivo *Principal.java*

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class Principal {
    public static void main(String[] args) {
        String nameArg = "", typeArg = "";
        Scanner input = new Scanner(System.in);
        Avaliacao evalArray[] = new Avaliacao[3];
        float maxGradeArg = 0.0f;
        int count = 0;

        while(count < 3) {
            boolean flag = true;
            int numOfEval = count + 1;
            System.out.println("\n|----- AVALIAÇÃO " + numOfEval +
" -----|");

            System.out.print("Insira o nome: ");
            nameArg = input.nextLine();
            System.out.print("Insira o tipo: ");
            typeArg = input.nextLine();

            while(flag) {
                try {
                    System.out.print("Insira a nota máxima: ");
                    maxGradeArg = input.nextFloat();
                    input.nextLine();
                    flag = false;
                } catch (InputMismatchException invalidInput) {
                    System.err.println("\nEntrada de dados inválida.
Tente novamente.");
                    input.nextLine();
                }
            }

            Avaliacao aux = new Avaliacao(nameArg, typeArg, maxGradeArg);
            evalArray[count] = aux;
            count++;
        }

        input.close();
        System.out.println("\n|----- AVALIAÇÕES REGISTRADAS -----
-----|");

        for(int i = 0; i < evalArray.length; i++) {
            System.out.println(evalArray[i].toString());
        }
    }
}
```

```
}  
}
```

## EXECUÇÃO

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
Microsoft Windows [versão 10.0.19045.3324]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
F:\Mackenzie\4Semestre\EstruturaDeDadosII\Exercicio01>javac Principal.java  
  
F:\Mackenzie\4Semestre\EstruturaDeDadosII\Exercicio01>java Principal  
  
|----- AVALIAÇÃO 1 -----|  
Insira o nome: Avaliacao01  
Insira o tipo: Tipo01  
Insira a nota máxima: 5.0  
  
Entrada de dados inválida. Tente novamente.  
Insira a nota máxima: 5,0  
  
|----- AVALIAÇÃO 2 -----|  
Insira o nome: Avaliacao02  
Insira o tipo: Tipo02  
Insira a nota máxima: 2,5  
  
|----- AVALIAÇÃO 3 -----|  
Insira o nome: Avaliacao03  
Insira o tipo: Tipo03  
Insira a nota máxima: 2,5  
  
|----- AVALIAÇÕES REGISTRADAS -----|  
Nome: Avaliacao01 | Tipo: Tipo01 | Nota máxima: 5.0 | Percentual final: 50.0% ;  
Nome: Avaliacao02 | Tipo: Tipo02 | Nota máxima: 2.5 | Percentual final: 25.0% ;  
Nome: Avaliacao03 | Tipo: Tipo03 | Nota máxima: 2.5 | Percentual final: 25.0% ;  
  
F:\Mackenzie\4Semestre\EstruturaDeDadosII\Exercicio01>
```

## EXERCÍCIO 02

### Código-fonte do arquivo *Automovel.java*

```
public class Automovel {  
    private String placa;  
    private String marca;  
    private int ano;  
    private float preco;  
  
    // Métodos Construtores  
    public Automovel() { // Sem parâmetros  
        this("", "", 0, 0.0f);  
    }  
}
```

```

    public Automovel(String placaArg, String marcaArg, int anoArg, float
precoArg) {
        this.placa = placaArg;
        this.marca = marcaArg;
        this.ano = anoArg;
        this.preco = precoArg;
    }

    // Métodos Getters e Setters
    public String getPlaca() {
        return placa;
    }

    public void setPlaca(String placaArg) {
        this.placa = placaArg;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marcaArg) {
        this.marca = marcaArg;
    }

    public int getAno() {
        return ano;
    }

    public void setAno(int anoArg) {
        this.ano = anoArg;
    }

    public float getPreco() {
        return preco;
    }

    public void setPreco(float precoArg) {
        this.preco = precoArg;
    }

    // Override do Método toString()
    @Override
    public String toString() {
        return "Placa: " + placa + " | " + "Marca: " + marca + " | " +
            "Ano: " + ano + " | " + "Preço: R$" + preco + ",";
    }
}

```

## Código-fonte do arquivo *Principal.java*

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class Principal {
    // Item A
    private static final int MAX_CARROS = 100;
    private static Automovel autoArray[] = new Automovel[MAX_CARROS];

    // Item D
    public static String getCarsOf(String marcaArg) {
        StringBuilder buffer = new StringBuilder();

        for(int i = 0; i < autoArray.length; i++) {
            if(autoArray[i] != null &&
autoArray[i].getMarca().equals(marcaArg)) {
                buffer.append(autoArray[i] + "\n");
            }
        }

        return buffer.toString();
    }

    // Item E
    public static float getAverageOf(Automovel[] array) {
        float total = 0.0f;
        int arrayLength = 0;
        int count = 0;

        while(array[count] != null) {
            total += array[count].getPreco();
            arrayLength++;
            count++;
        }

        return (total/(float)arrayLength);
    }

    // Item F
    public static int getNumAfter(int anoArg) {
        int count = 0, numAfter = 0;

        while(autoArray[count] != null) {
            if(autoArray[count].getAno() == anoArg) {
                numAfter++;
            }

            count++;
        }
    }
}
```

```

    }

    return numAfter;
}

// Item G
public static float getAverageBetween(int startYear, int finalYear) {
    float total = 0.0f;
    int count = 0, numOfCarsBetween = 0;

    while(autoArray[count] != null) {
        int currentCarYear = autoArray[count].getAno();

        if(currentCarYear >= startYear && currentCarYear <=
finalYear) {
            total += autoArray[count].getPreco();
            numOfCarsBetween++;
        }

        count++;
    }

    return (total/(float) numOfCarsBetween);
}

public static void main(String[] args) {
    int numOfCars = 0, anoArg = 0, afterYear = 0, startYearArg = 0,
finalYearArg = 0;
    Scanner input = new Scanner(System.in);
    String placaArg = "", marcaArg = "";
    float precoArg = 0.0f;
    boolean flag = true;

    // Item B
    while(flag) {
        try {
            System.out.print("Insira o número de carros (Máximo =
100): ");
            numOfCars = input.nextInt();

            if(numOfCars >= 0 && numOfCars <= MAX_CARROS) {
                flag = false;
            } else {
                System.out.println("Número inválido. Tente
novamente.");
            }

            input.nextLine();
        } catch (InputMismatchException error) {

```

```

        System.err.println("Entrada de dados inválida. Tente novamente.");
        input.nextLine();
    }
}

for(int i = 0; i < numOfCars; i++) {
    int numOfAuto = i + 1;
    System.out.println("\n|----- AUTOMÓVEL " + numOfAuto +
" -----|");
    System.out.print("Insira a placa do automóvel: ");
    placaArg = input.nextLine();
    System.out.print("Insira a marca do automóvel: ");
    marcaArg = input.nextLine();

    flag = true;
    while(flag) {
        try {
            System.out.print("Insira o ano do automóvel: ");
            anoArg = input.nextInt();
            input.nextLine();
            flag = false;
        } catch (InputMismatchException error) {
            System.err.println("\nEntrada de dados inválida. Tente novamente.");
            input.nextLine();
        }
    }

    flag = true;
    while(flag) {
        try {
            System.out.print("Insira o preço do automóvel: ");
            precoArg = input.nextFloat();
            input.nextLine();
            flag = false;
        } catch (InputMismatchException error) {
            System.err.println("\nEntrada de dados inválida. Tente novamente.");
            input.nextLine();
        }
    }

    Automovel aux = new Automovel(placaArg, marcaArg, anoArg,
precoArg);
    autoArray[i] = aux;
}

// Item C

```

```

        System.out.println("\n|----- AUTOMÓVEIS REGISTRADOS -----
-----|");
        for(int j = 0; j < numOfCars; j++) {
            System.out.println(autoArray[j].toString());
        }

        System.out.print("\nInsira uma marca de automóvel: ");
        String marcaInput = input.nextLine();
        System.out.println("\n|----- AUTOMÓVEIS DA MARCA " +
marcaInput +
        " -----|");
        System.out.println(getCarsOf(marcaInput));

        System.out.println("Preço médio dos carros registrados: R$" +
getAverageOf(autoArray) + "\n");

        flag = true;
        while(flag) {
            try {
                System.out.print("Insira um ano: ");
                afterYear = input.nextInt();
                input.nextLine();

                if(afterYear < 0) {
                    System.out.println("Ano inválido. Tente novamente");
                } else {
                    flag = false;
                }
            } catch (InputMismatchException error) {
                System.err.println("Entrada de dados inválida (apenas
números inteiros). Tente novamente.");
                input.nextLine();
            }
        }

        afterYear++;
        System.out.println("Número de automóveis fabricados em " +
afterYear +
        " (um ano após o ano informado): " + getNumAfter(afterYear) +
        "\n");

        flag = true;
        while(flag) {
            try {
                System.out.print("Insira o ano inicial: ");
                startYearArg = input.nextInt();
                input.nextLine();

                if(startYearArg < 0) {

```



```

        System.out.println("Ano inválido. Tente novamente.");
    } else {
        flag = false;
    }
} catch (InputMismatchException error) {
    System.err.println("Entrada de dados inválida (apenas
números inteiros). Tente novamente.");
    input.nextLine();
}

}

flag = true;
while(flag) {
    try {
        System.out.print("Insira o ano final: ");
        finalYearArg = input.nextInt();
        input.nextLine();

        if(finalYearArg < 0 || finalYearArg < startYearArg) {
            System.out.println("Ano inválido. Tente novamente.");
        } else {
            flag = false;
        }
    } catch (InputMismatchException error) {
        System.err.println("Entrada de dados inválida (apenas
números inteiros). Tente novamente.");
        input.nextLine();
    }
}

System.out.println("\nMédia dos preços dos carros fabricados de "
+ startYearArg +
    " até " + finalYearArg + ": R$" + getAverageBetween(startYearArg,
finalYearArg));

    input.close();
}
}

```

## EXECUÇÃO

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Microsoft Windows [versão 10.0.19045.3324]  
(c) Microsoft Corporation. Todos os direitos reservados.

F:\Mackenzie\4Semestre\EstruturaDeDadosII\Exercicio02>javac Principal.java

F:\Mackenzie\4Semestre\EstruturaDeDadosII\Exercicio02>java Principal  
Insira o número de carros (Máximo = 100): 5

|----- AUTOMÓVEL 1 -----|  
Insira a placa do automóvel: A  
Insira a marca do automóvel: Honda  
Insira o ano do automóvel: 2015  
Insira o preço do automóvel: 35000

|----- AUTOMÓVEL 2 -----|  
Insira a placa do automóvel: B  
Insira a marca do automóvel: Ford  
Insira o ano do automóvel: 2020  
Insira o preço do automóvel: 50000

|----- AUTOMÓVEL 3 -----|  
Insira a placa do automóvel: C  
Insira a marca do automóvel: Fiat  
Insira o ano do automóvel: 2009  
Insira o preço do automóvel: 20000

|----- AUTOMÓVEL 4 -----|  
Insira a placa do automóvel: D  
Insira a marca do automóvel: Honda  
Insira o ano do automóvel: 2010  
Insira o preço do automóvel: 30000

|----- AUTOMÓVEL 5 -----|  
Insira a placa do automóvel: E  
Insira a marca do automóvel: Ford  
Insira o ano do automóvel: 2021  
Insira o preço do automóvel: 60000

|----- AUTOMÓVEIS REGISTRADOS -----|  
Placa: A | Marca: Honda | Ano: 2015 | Preço: R\$35000.0;  
Placa: B | Marca: Ford | Ano: 2020 | Preço: R\$50000.0;  
Placa: C | Marca: Fiat | Ano: 2009 | Preço: R\$20000.0;  
Placa: D | Marca: Honda | Ano: 2010 | Preço: R\$30000.0;

Preço médio dos carros registrados: R\$39000.0

Insira um ano: 2008  
Número de automóveis fabricados em 2009 (um ano após o ano informado): 1

Insira o ano inicial: 2010  
Insira o ano final: 2020

Média dos preços dos carros fabricados de 2010 até 2020: R\$38333.332

F:\Mackenzie\4Semestre\EstruturaDeDadosII\Exercicio02>|

## EXERCÍCIO 03

### Código-fonte do arquivo *Candidato.java*

```
public class Candidato {
    private int numCandidato;
    private String cpf;
    private String nome;
    private int idade;
    private String sexo;
    private String generoMusical;

    public Candidato() {
        this(0, "\\0", "\\0", 0, "\\0", "\\0");
    }

    public Candidato(int numeroArg, String cpfArg, String nomeArg, int
idadeArg, String sexoArg, String genMusArg) {
        this.numCandidato = numeroArg;
        this.cpf = cpfArg;
        this.nome = nomeArg;
        this.idade = idadeArg;
        this.sexo = sexoArg;
        this.generoMusical = genMusArg;
    }

    public int getNumCandidato() {
        return numCandidato;
    }

    public void setNumCandidato(int numArg) {
        this.numCandidato = numArg;
    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpfArg) {
        this.cpf = cpfArg;
    }

    public String getNome() {
        return nome;
    }

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

```

    public int getIdade() {
        return idade;
    }

    public void setIdade(int idadeArg) {
        this.idade = idadeArg;
    }

    public String getSexo() {
        return sexo;
    }

    public void setSexo(String sexoArg) {
        this.sexo = sexoArg;
    }

    public String getGenMus() {
        return generoMusical;
    }

    public void setGenMus(String genMusArg) {
        this.generoMusical = genMusArg;
    }

    @Override
    public String toString() {
        return "Número: " + numCandidato + " | " + "CPF: " + cpf + " | "
+ "Nome: " +
        nome + " | " + "Idade: " + idade + " | " + "Sexo: " + sexo +
" | " +
        "Gênero Musical: " + generoMusical + ";";
    }
}

```

### Código-fonte do arquivo *Juri.java*

```

public class Juri {
    private String nomeCompleto;
    private String nomeArtistico;
    private String generoMusicalPrincipal;

    public Juri() {
        this("\0", "\0", "\0");
    }

    public Juri(String nomeCompletoArg, String nomeArtisticoArg, String
genMusPrinArg) {

```

```

        this.nomeCompleto = nomeCompletoArg;
        this.nomeArtístico = nomeArtísticoArg;
        this.generoMusicalPrincipal = genMusPrinArg;
    }

    public String getNomeCompleto() {
        return nomeCompleto;
    }

    public void setNomeCompleto(String nomeCompletoArg) {
        this.nomeCompleto = nomeCompletoArg;
    }

    public String getNomeArtístico() {
        return nomeArtístico;
    }

    public void setNomeArtístico(String nomeArtísticoArg) {
        this.nomeArtístico = nomeArtísticoArg;
    }

    public String getGenMusPrin() {
        return generoMusicalPrincipal;
    }

    public void setGenMusPrin(String genMusPrinArg) {
        this.generoMusicalPrincipal = genMusPrinArg;
    }

    @Override
    public String toString() {
        return "Nome Completo: " + nomeCompleto + " | " + "Nome
Artístico: " +
            nomeArtístico + " | " + "Gênero Musical Principal: " +
            generoMusicalPrincipal + ",";
    }
}

```

### Código-fonte do arquivo *Concurso.java*

```

public class Concurso {
    private String nome;
    private Candidato[] candidatos;
    private Juri[] jurados;

    public Concurso() {
        this("\0", null, null);
    }
}

```

```

    }

    public Concurso(String nomeArg, Candidato[] candidatosArg, Juri[]
juriArg) {
        this.nome = nomeArg;
        this.candidatos = candidatosArg;
        this.jurados = juriArg;
    }

    public String filtraGeneroMusical(String generoMusical){
        StringBuilder filtrados = new StringBuilder();
        for (int i = 0; i < candidatos.length; i++){
            if(generoMusical.equals(candidatos[i].getGenMus())){
                filtrados.append(candidatos[i].getNome());
            }
        }
        for (int i = 0; i < jurados.length;i++){
            if(generoMusical.equals(jurados[i].getGenMusPrin())){
                filtrados.append(" - ");
                filtrados.append(jurados[i].getNomeArtistico());
            }
        }
        return filtrados.toString();
    }

    public String filtraSexoIdade(String sexo, int idadeMin, int
idadeMax) {
        StringBuilder filtrados = new StringBuilder();
        for(int i = 0; i < candidatos.length; i++) {
            if(sexo.equals(candidatos[i].getSexo()) && candidatos[i]
instanceof Candidato) {
                if(idadeMax < candidatos[i].getIdade() && idadeMin >
candidatos[i].getIdade()) {
                    filtrados.append(candidatos[i].getNome());
                }
            }
        }

        return filtrados.toString();
    }

    public int contaCandidatos(String sexo){
        int contador = 0;
        for (int i =0; i < candidatos.length; i++){
            if(sexo.equals(candidatos[i].getSexo())){
                contador += 1;
            }
        }
    }

```

```

        return contador;
    }

    @Override
    public String toString() {
        StringBuilder buffer = new StringBuilder();

        buffer.append("CONCURSO " + nome);
        buffer.append("\n\nCANDIDATOS\n");
        for (int i = 0; i < candidatos.length; i++){
            buffer.append(candidatos[i] + "\n");
        }
        buffer.append("\nJURI\n");
        for (int i=0; i < jurados.length; i++){
            buffer.append(jurados[i] + "\n");
        }

        return buffer.toString();
    }

    public static void main(String[] args) {
        Candidato candidato01 = new Candidato(1, "123.456.564.67", "Zeca
Pagodes", 69, "Masculino", "Samba");
        Candidato candidato02 = new Candidato(2, "394.234.543.32",
"Pipokinha", 23, "Feminino", "Funk");
        Candidato candidato03 = new Candidato(3, "779.223.442.13",
"Manoel Gomes", 37, "Masculino", "MPB");
        Candidato[] candidatos = {candidato01, candidato02, candidato03};

        Juri jurado01 = new Juri("Julio", "Julão da massa", "Rap");
        Juri jurado02 = new Juri("Jordan Americo", "JorJor", "House");
        Juri jurado03 = new Juri("Amelia Aparecida", "Amelecida", "MPB");
        Juri[] jurados = {jurado01, jurado02, jurado03};

        Concurso concurso = new Concurso("The Voice Chernobyl",
candidatos, jurados);
        System.out.println(concurso);

        System.out.println("Candidatos e jurados que tocam 'Funk'");
        System.out.println(concurso.filtroGeneroMusical("Funk"));
        System.out.println("\nCandidatos e jurados que tocam 'MPB'");
        System.out.println(concurso.filtroGeneroMusical("MPB"));
        System.out.println("\nCandidatos 'HOMENS' que tem entre 50 e 80
anos");
        System.out.println(concurso.filtroSexoIdade("Masculino", 80,
50));
        System.out.print("\nQuantidade de candidatos do sexo Masculino:
");
        System.out.println(concurso.contaCandidatos("Masculino"));
    }
}

```

```

        System.out.print("Quantidade de candidatos do sexo Feminino: ");
        System.out.println(concurso.contaCandidatos("Feminino"));
    }
}

```

## EXECUÇÃO

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [versão 10.0.19045.3324]
(c) Microsoft Corporation. Todos os direitos reservados.

F:\Mackenzie\4Semestre\EstruturaDeDadosII\ConcursoMusical>cd src

F:\Mackenzie\4Semestre\EstruturaDeDadosII\ConcursoMusical\src>javac Concurso.java

F:\Mackenzie\4Semestre\EstruturaDeDadosII\ConcursoMusical\src>java Concurso
CONCURSO The Voice Chernobyl

CANDIDATOS
Número: 1 | CPF: 123.456.564.67 | Nome: Zeca Pagodes | Idade: 69 | Sexo: Masculino | Gênero Musical: Samba;
Número: 2 | CPF: 394.234.543.32 | Nome: Pipokinha | Idade: 23 | Sexo: Feminino | Gênero Musical: Funk;
Número: 3 | CPF: 779.223.442.13 | Nome: Manoel Gomes | Idade: 37 | Sexo: Masculino | Gênero Musical: MPB;

JURI
Nome Completo: Julio | Nome Artístico: Julão da massa | Gênero Musical Principal: Rap;
Nome Completo: Jordan Americo | Nome Artístico: JorJor | Gênero Musical Principal: House;
Nome Completo: Amelia Aparecida | Nome Artístico: Amelecida | Gênero Musical Principal: MPB;

Candidatos e jurados que tocam 'Funk'
Pipokinha

Candidatos e juris que tocam 'MPB'
Manoel Gomes - Amelecida

Candidatos 'HOMENS' que tem entre 50 e 80 anos
Zeca Pagodes

Quantidade de candidatos do sexo Masculino: 2
Quantidade de candidatos do sexo Feminino: 1

F:\Mackenzie\4Semestre\EstruturaDeDadosII\ConcursoMusical\src>

```