

# Programação Orientada a Objetos

## Construção de Objetos

### Exercícios



Joinville Batista Junior

### Exemplo

```
package entidade;

public class Aluno {

    public enum EstadoCivil {solteiro, casado, separado, viúvo};

    private String nome;
    private int ano_nascimento;
    private char sexo; // valor default = '\u0000'
    private EstadoCivil estado_civil;
    private boolean estrangeiro;

    public Aluno(String nome, int ano_nascimento, char sexo, EstadoCivil estado_civil,
        boolean estrangeiro) {
        this.nome = nome;
        this.ano_nascimento = ano_nascimento;
        this.sexo = sexo;
        this.estado_civil = estado_civil;
        this.estrangeiro = estrangeiro;
    }
}
```

## Exemplo

```
public String getNome() {  
    return nome;  
}  
  
public void setNome(String nome) {  
    this.nome = nome;  
}  
  
public int getAnoNascimento() {  
    return ano_nascimento;  
}  
  
public void setAnoNascimento(int ano_nascimento) {  
    this.ano_nascimento = ano_nascimento;  
}  
  
public char getSexo() {  
    return sexo;  
}
```

## Exemplo

```
public void setSexo(char sexo) {  
    this.sexo = sexo;  
}  
  
public EstadoCivil getEstadoCivil() {  
    return estado_civil;  
}  
  
public void setEstadoCivil(EstadoCivil estado_civil) {  
    this.estado_civil = estado_civil;  
}  
  
public boolean isEstrangeiro() {  
    return estrangeiro;  
}  
  
public void setEstrangeiro(boolean estrangeiro) {  
    this.estrangeiro = estrangeiro;  
}  
}
```

## Exemplo

```
private String toStringSexo() {
    if ((sexo == 'M') || (sexo == 'm')) return "masculino";
    if ((sexo == 'F') || (sexo == 'f')) return "feminino";
    return "indefinido";
}

private String toStringEstadoCivil () {
    String sexo_string = toStringSexo();
    if (sexo_string.equals("masculino")) return estado_civil + "";
    if (sexo_string.equals("feminino")) {
        switch(estado_civil) {
            case solteiro: return "solteira";
            case casado: return "casada";
            case separado: return "separada";
            case viúvo: return "viúva";
            default: return "indefinido";
        }
    }
    return "indefinido";
}
```

## Exemplo

```
private String toStringEstrangeiro() {
    if (!estrangeiro) return "";
    String sexo_string = toStringSexo();
    if (sexo_string.equals("masculino")) return "estrangeiro";
    if (sexo_string.equals("feminino")) return "estrangeira";
    return "indefinido";
}

public String toString() {
    String dados_aluno;
    dados_aluno = "nome: " + getNome()
        + " -- ano de nascimento " + getAnoNascimento()
        + " -- sexo: " + toStringSexo()
        + " -- estado civil: " + toStringEstadoCivil();
    String estrangeiro = toStringEstrangeiro();
    If (!estrangeiro.isEmpty()) dados_aluno += " -- " + estrangeiro;
    return dados_aluno;
}
}
```

## Exemplo

```
package controle;

import entidade.Aluno;
import entidade.Aluno.EstadoCivil;
import java.util.ArrayList;

public class ImpressãoAlunos {

    private static Aluno[] listaAluno;

    public static void main(String[] args) {
        listaAluno = criarAlunos();
        imprimirAlunos ("Alunos Cadastrados:", listaAluno);
        ArrayList<Aluno> selecionados_listaAluno = selecionarAlunos();
        imprimirAlunos ("Alunos Selecionados:",
            selecionados_listaAluno.toArray(new Aluno[selecionados_listaAluno.size()]));
    }
}
```

## Exemplo

```
private static Aluno[] criarAlunos() {
    Aluno[] listaAluno = new Aluno[8];
    listaAluno[0] = new Aluno ("Ana Julia", 1995, 'F', EstadoCivil.solteiro, false);
    listaAluno[1] = new Aluno ("Joaquim", 1990, 'M', EstadoCivil.casado, true);
    listaAluno[2] = new Aluno ("Ana Ligia", 1998, 'F', EstadoCivil.solteiro, false);
    listaAluno[3] = new Aluno ("Mateus", 1991, 'M', EstadoCivil.solteiro, false);
    listaAluno[4] = new Aluno ("Livia", 1985, 'F', EstadoCivil.casado, true);
    listaAluno[5] = new Aluno ("Roberto", 1990, 'M', EstadoCivil.casado, false);
    listaAluno[6] = new Aluno ("Ana Maria", 1991, 'F', EstadoCivil.solteiro, false);
    listaAluno[7] = new Aluno ("Sandro", 1992, 'M', EstadoCivil.solteiro, true);
    return listaAluno;
}
```

## Exemplo

```
private static ArrayList<Aluno> selecionarAlunos() {
    int ano_inicial = 1990;
    int ano_final = 1995;
    EstadoCivil estado_civil = Aluno.EstadoCivil.solteiro;
    char estrangeiro = 'X';
    System.out.println("--- Filtros de seleção\n  ano inicial: "
        + ano_inicial + " - ano final: " + ano_final + " - estado civil: "
        + estado_civil + " - estrangeiro = " + estrangeiro + "\n");
    return filtrarAlunos(ano_inicial, ano_final, estado_civil, estrangeiro);
}
```

## Exemplo

```
private static ArrayList<Aluno> filtrarAlunos(int ano_inicial,
    int ano_final, EstadoCivil estado_civil, char estrangeiro) {
    ArrayList<Aluno> selecionados_listaAluno = new ArrayList();
    for (int n = 0; n < listaAluno.length; n++) {
        int ano_nascimento = listaAluno[n].getAnoNascimento();
        EstadoCivil estado_civil_aluno = listaAluno[n].getEstadoCivil();
        boolean é_estrangeiro = listaAluno[n].isEstrangeiro();
        if ((ano_inicial > -1) && (ano_nascimento < ano_inicial)) continue;
        if ((ano_final > -1) && (ano_nascimento > ano_final)) continue;
        if ((estado_civil != null) && (estado_civil != estado_civil_aluno)) continue;
        if ((estrangeiro == 'T') && (!é_estrangeiro)) continue;
        if ((estrangeiro == 'F') && (é_estrangeiro)) continue;
        selecionados_listaAluno.add(listaAluno[n]);
    }
    return selecionados_listaAluno;
}
```

## Exemplo

```
private static void imprimirAlunos (String cabeçalho, Aluno[] listaAluno) {  
    System.out.println("---- " + cabeçalho);  
    for (Aluno itemAluno : listaAluno) {  
        System.out.println(itemAluno.toString());  
    }  
    System.out.println();  
}  
  
}
```

## Lista de Exercícios

1.1 – Adapte o exemplo da apostila:

- a) substituindo a classe Aluno pela classe Disciplina, com os seguintes dados: título, carga horária total, carga horária prática, tipo (obrigatória, optativa, eletiva), reuni (sim ou não);
- b) e substituindo os filtros de impressão por: carga teórica mínima, tipo, reuni.

1.2 – Adapte o exemplo da apostila:

- a) substituindo a classe Aluno por uma classe de sua escolha, que tenha pelo menos: um dado do tipo enumerado e um dado booleano;
- b) e definindo os filtros que julgar necessário, incluído pelo menos o dado enumerado e o dado booleano.

## Lista de Exercícios

1.3 – Adapte o exercício 1.1:

- a) incluindo o filtro de pesquisa “substring inicial do título”;
- b) e utilizando o método “boolean startsWith (String)” da classe String para verificar se um string começa com um determinado substring.

1.4 – Adapte o exemplo da apostila:

- a) criando um construtor que só inicializa: nome, ano\_nascimento e sexo;
- b) e alterando o método criarAlunos para utilizar esse construtor e métodos de alteração, para criar e inicializar os demais dados de cada aluno após a sua criação.

## Lista de Exercícios

1.5 - Adapte o exemplo da apostila:

- a) substituindo todas as utilizações de array por ArrayList;
- b) e usando somente iterações do tipo: for (int n = 0; n < conjunto.size(); n++).

1.6 – Adapte o exercício 1.5:

- a) usando somente iterações do tipo: for (Aluno itemAluno : listaAlunos).

1.7 – Adapte o exemplo da apostila:

- a) incluindo o método ordenarAlunos, para ordenar os alunos criados com seus nomes em ordem alfabética;
- b) utilize o método “int compareToIgnoreCase(String)” da classe String para comparar lexicograficamente dois strings, ignorando se as letras dos strings são maiúsculas ou minúsculas.