

Aula Prática 4

Objectivos

Utilização de herança e polimorfismo.

Problema 4.1

Utilizando o trabalho desenvolvido no exercício 3.1 da aula prática 3:

- a) Crie a classe `Disciplina` com as seguintes características:
- o Nome da disciplina (`String`)
 - o Área científica (`String`)
 - o ECTS (`int`)
 - o Responsável (`Professor`) // Um professor é uma pessoa com `nfunc` e `dataAdmissao`
 - o Alunos (`Estudante[]`) // Vector dinâmico de Estudantes não "repetidos"

- e) Implemente os seguintes métodos:

```
public boolean addAluno(Estudante est)    // Adiciona um aluno à Disciplina
public boolean delAluno(int nMec)        // Remove um aluno da Disciplina
public boolean alunoInscrito(int nMec)    // Verifica se aluno está inscrito à
                                         disciplina
public int numAlunos()                   // Retorna o Nº de Alunos Inscritos
public String toString()                  // Retorna as características da Disciplina
public Estudante[] getAlunos()            // Retorna uma lista com todos
                                         os alunos da disciplina
public Estudante[] getAlunos(String tipo) // Retorna uma lista com todos os alunos
                                         da disciplina do subtipo "tipo"
```

- f) Teste o trabalho desenvolvido com o seguinte programa:

```
public class Test {

    public static void main(String[] args) {
        Estudante est1 = new Estudante("Andreia", 9855678, new Data(18, 7, 1974));
        Estudante est2 = new Estudante("Monica", 75266454, new Data(11, 8, 1978));
        Estudante est3 = new Estudante("Jose", 85265426, new Data(15, 2, 1976));
        Estudante est4 = new Estudante("Manuel", 85153442, new Data(1, 3, 1973));

        Bolseiro bls1 = new Bolseiro("Maria", 8976543, new Data(12, 5, 1976));
        Bolseiro bls2 = new Bolseiro("Xico", 872356522, new Data(21, 4, 1975));
        Bolseiro bls3 = new Bolseiro("Duarte", 32423512, new Data(6, 1, 1976));
        bls1.setBolsa(745);
        bls2.setBolsa(845);
        bls3.setBolsa(745);

        Professor pf1 = new Professor("Jose Manuel", 11223344, new Data(11, 9, 1999));

        Disciplina dis = new Disciplina("P5", "Informatica", 6, pf1);
        dis.addAluno(est1);
        dis.addAluno(est2);
        dis.addAluno(bls1);
    }
}
```

```

        if (dis.alunoInscrito(est3.getNMec()))
            System.out.println("\n" + est3
                + " \n\t-> ESTÁ inscrito na Disciplina");
        else
            System.out.println("\n" + est3
                + " \n\t-> NÃO ESTÁ inscrito na Disciplina");

        System.out.println("\nNº de Alunos Inscritos: " + dis.numAlunos());

        dis.addAluno(est3);
        dis.addAluno(bls2);
        dis.addAluno(est4);
        dis.addAluno(bls3);
        if (!dis.addAluno(bls3))
            System.out.println(bls3.getNMec() + ", " + bls3.getName()
                + " já está inscrito nesta disciplina!");
        if (dis.delAluno(bls2.getNMec()))
            System.out.println(bls2.getName() + " Removido!");

        System.out.println("\nNº de Alunos Inscritos: " + dis.numAlunos());

        System.out.println(dis + "\n");

        for (Estudante e : dis.getAlunos())
            System.out.println(e);
        for (Estudante e : dis.getAlunos("Bolseiro"))
            System.out.println(e);
    }
}

```

Verifique se obteve o seguinte resultado:

```

Jose, BI: 85265426, Nascido na Data: 15/2/1976, Estudante NMec: 102, Inscrito
em Data: 25/10/2007
-> NÃO ESTÁ inscrito na Disciplina

Nº de Alunos Inscritos: 3
106, Duarte já está inscrito nesta disciplina!
Xico Removido!

Nº de Alunos Inscritos: 6
Disciplina: P5 ( 6 ECTS) da Area de Informatica
Responsável: PROFESSOR: Jose Manuel, BI: 11223344, Nascido na Data: 11/9/1999,
NMec: 1, Admitido em Data: 25/10/2007
Existem 6 Alunos Inscritos

Andreia, BI: 9855678, Nascido na Data: 18/7/1974, Estudante NMec: 100,
Inscrito em Data: 25/10/2007
Monica, BI: 75266454, Nascido na Data: 11/8/1978, Estudante NMec: 101,
Inscrito em Data: 25/10/2007
Maria, BI: 8976543, Nascido na Data: 12/5/1976, Estudante NMec: 104, Inscrito
em Data: 25/10/2007, Bolseiro com bolsa de 745 euros
Jose, BI: 85265426, Nascido na Data: 15/2/1976, Estudante NMec: 102, Inscrito
em Data: 25/10/2007
Manuel, BI: 85153442, Nascido na Data: 1/3/1973, Estudante NMec: 103, Inscrito
em Data: 25/10/2007
Duarte, BI: 32423512, Nascido na Data: 6/1/1976, Estudante NMec: 106, Inscrito
em Data: 25/10/2007, Bolseiro com bolsa de 745 euros

Maria, BI: 8976543, Nascido na Data: 12/5/1976, Estudante NMec: 104, Inscrito
em Data: 25/10/2007, Bolseiro com bolsa de 745 euros
Duarte, BI: 32423512, Nascido na Data: 6/1/1976, Estudante NMec: 106, Inscrito
em Data: 25/10/2007, Bolseiro com bolsa de 745 euros

```

Problema 4.2

Utilizando o programa desenvolvido nas aulas anteriores relativamente às figuras geométricas. (classes Ponto, Figura, Circulo, Quadrado e Rectângulo):

a) Teste novamente as classes desenvolvidos com o seguinte programa:

```
public class Test {

    public static void main(String[] args) {
        Figura listaFig[] = new Figura[15];
        Circulo c1 = new Circulo(2);
        Circulo c2 = new Circulo(1, 3, 2);
        Quadrado q1 = new Quadrado(2);
        Quadrado q2 = new Quadrado(3, 4, 2);
        Rectangulo r1 = new Rectangulo(2, 3);
        Rectangulo r2 = new Rectangulo(3, 4, 2, 3);
        Rectangulo r3 = new Rectangulo(r1);

        listaFig[0] = c2;
        listaFig[1] = r1;
        listaFig[2] = r2;
        listaFig[3] = q2;
        listaFig[4] = c1;
        listaFig[5] = q1;

        System.out.println("\nLista de Figuras:");
        for (int i = 0; i < listaFig.length; i++) {
            if (listaFig[i] != null)
                System.out.println(listaFig[i]);
        }

        System.out.println("\nCirculos na Lista de Figuras:");
        for (int i = 0; i < listaFig.length; i++) {
            if (listaFig[i] != null && listaFig[i] instanceof Circulo)
                System.out.println(listaFig[i]);
        }

        System.out.println("\nCentro das Figuras:");
        for (int i = 0; i < listaFig.length; i++) {
            if (listaFig[i] != null)
                System.out.println(listaFig[i].getCentro());
        }

        System.out.println("\n" + r1 + " É IGUAL " + r2 + "? " + r1.equals(r2));
        System.out.println("\n" + r1 + " É IGUAL " + r3 + "? " + r1.equals(r3));
    }
}
```

Verifique se obteve o seguinte resultado:

```
Lista de Figuras:
Circulo de Centro x: 1.0, y:3.0 e de raio 2.0
Rectangulo de Centro x: 0.0, y:0.0, altura 3.0, comprimento 2.0
Rectangulo de Centro x: 3.0, y:4.0, altura 3.0, comprimento 2.0
Quadrado de Centro x: 3.0, y:4.0 e de lado 2.0
Circulo de Centro x: 0.0, y:0.0 e de raio 2.0
Quadrado de Centro x: 0.0, y:0.0 e de lado 2.0
```

```

Circulos na Lista de Figuras:
Circulo de Centro x: 1.0, y:3.0 e de raio 2.0
Circulo de Centro x: 0.0, y:0.0 e de raio 2.0

Centro das Figuras:
x: 1.0, y:3.0
x: 0.0, y:0.0
x: 3.0, y:4.0
x: 3.0, y:4.0
x: 0.0, y:0.0
x: 0.0, y:0.0

Rectangulo de Centro x: 0.0, y:0.0, altura 3.0, comprimento 2.0 É IGUAL
Rectangulo de Centro x: 3.0, y:4.0, altura 3.0, comprimento 2.0? false

Rectangulo de Centro x: 0.0, y:0.0, altura 3.0, comprimento 2.0 É IGUAL
Rectangulo de Centro x: 0.0, y:0.0, altura 3.0, comprimento 2.0? true

```

Problema 4.3

Utilizando os conceitos de Polimorfismo, re-escreva o programa desenvolvido nas aulas anteriores relativamente ao sistema de informação para videoclub.

Problema 4.4

Considere as seguintes entidades:

- ~~Calçado, caracterizado por um nome (string), fabricante (string) e um preço (double).~~
- ~~Bota, tipo de calçado caracterizada por um tipo (Cano, MeioCano).~~
- ~~Sapatilha, calçado desportivo caracterizado por um tipo (Atletismo, Futebol, Ginastica,...)~~
- ~~Chuteira, sapatilha de futebol caracterizada pelo tipoPitoes (Aluminio, Borracha, Plastico) e numeroPitoes (byte)~~
- ~~BotaDesporto, bota desportiva caracterizada por um tipo (Snowboard, Basket,...).~~
- ~~Catálogo, caracterizado por uma edição (string) e por representar um conjunto de calçado não repetido.~~

Todo o calçado do tipo desportivo (Sapatilha, Chuteira, BotaDesporto, etc) deve ter um método que retorna a sua finalidade (método “String getFinalidade()”). Por exemplo: “Futebol”, “Atletismo”, “Montanha”, etc.

- Represente adequadamente estas entidades. Crie construtores, os métodos set/get que lhe pareçam adequados, bem como métodos que sejam fundamentais para cada classe.
- Teste as classes desenvolvidas com a seguinte função main

```

public class TestCalçado {
    public static void main(String[] args) {
        Bota b1 = new Bota("Miss Fininha", "GoldFactory", 75, "Cano");
        Bota b2 = new Bota("Miss Fininha", "GoldFactory", 75, "Cano");
        BotaDesporto bd1 = new BotaDesporto("IceMountain", "NewGeneration",
                                            99.5, "MeioCano", "SnowBoard");
    }
}

```

```

Calcado c1 = new Chuteira("X75", "Naique", 69.0, "Borracha", (byte) 6);
Calcado c2 = new Chuteira("fX300", "GaloDesportivo", 74.9, "Aluminio", (byte) 5);
Sapatilha s1 = new Sapatilha ("XGold","Pluma", 49.99, "Ginastica");

Catalogo prope = new Catalogo("Planta do Pé - Primavera/Verão 2013");
System.out.println("----- Insercao -----");
System.out.print(prope.add(b1) + ", "); // true
System.out.print(prope.add(b2) + ", "); // false
System.out.print(prope.add(bd1) + ", "); // true
System.out.print(prope.add(c1) + ", "); // true
System.out.print(prope.add(c2) + ", "); // true
System.out.println(prope.add(s1)); // true

System.out.println("----- Listagem de Calcado Desportivo -----");
Calcado calDesp[] = prope.getCalcadoDesportivo();

System.out.println("Preço Total do Calçado Desportivo -> " +
                    precoCalcado(calDesp));

    printFinalidade(calDesp);
}

private static double precoCalcado(Calcado cal[]){
    // Implemente...
    // Retorna o custo total do calcado passado como argumento
}

private static void printFinalidade(Calcado cal[]){
    // Implemente...
    // Lista o calcado desportivo, incluindo a finalidade
}
}

```