

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, 1969));

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

```

        if (dis.alunoInscrito(est3.nMec()))
            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.nMec() + ", " + bls3.nome()
                + " já está inscrito nesta disciplina!");
        if (dis.delAluno(bls2.nMec()))
            System.out.println(bls2.nome() + " Removido!");

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

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

        System.out.println("\n Listagem de Estudantes:");
        for (Estudante e : dis.getAlunos())
            System.out.println(e);

        System.out.println("\n Listagem de Bolseiros:");
        for (Estudante e : dis.getAlunos("Bolsheiro"))
            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

Listagem de Estudantes:
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, Bolsheiro 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, Bolsheiro com bolsa de 745 euros

Listagem de Bolseiros:
Maria, BI: 8976543, Nascido na Data: 12/5/1976, Estudante NMec: 104, Inscrito em Data:
25/10/2007, Bolsheiro com bolsa de 745 euros
Duarte, BI: 32423512, Nascido na Data: 6/1/1976, Estudante NMec: 106, Inscrito em Data:
25/10/2007, Bolsheiro 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) Crie uma nova classe ColecaoFiguras que suporta um conjunto (coleção) de Figuras não repetidas. Implemente os seguintes métodos:

```
public ColecaoFiguras(double maxArea)    // O construtor define a área máxima da
                                         coleção de figuras
public boolean addFigura(Figura f)       // Adiciona uma figura à coleção
public boolean delFigura(Figura f)       // Remove uma figura da coleção
public double areaTotal()                // Retorna a área total das figuras
public boolean exists(Figura f)          // Verifica se uma figura existe na coleção
public String toString()                 // Retorna as características da coleção
public Figura[] getFiguras()             // Retorna uma lista com todas
                                         as figuras da coleção
public Ponto[] getCentros()              // Retorna uma lista com todos os centros
                                         das figuras da coleção
```

- b) Teste as classes desenvolvidos com o seguinte programa:

```
public class Test {
    public static void main(String[] args) {
        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);

        ColecaoFiguras col = new ColecaoFiguras(42.0);           // MaxArea

        System.out.println(col.addFigura(c2));                   // true
        System.out.println(col.addFigura(r1));                   // true
        System.out.println(col.addFigura(r1));                   // false
        System.out.println(col.addFigura(r2));                   // true
        System.out.println(col.addFigura(c1));                   // true
        System.out.println(col.addFigura(q2));                   // true
        System.out.println(col.addFigura(q1));                   // false
        System.out.println(col.delFigura(r1));                   // true
        System.out.println(col.addFigura(q1));                   // true

        System.out.println("\nÁrea Total da Lista de Figuras: " + col.areaTotal());

        System.out.println("\nLista de Figuras:");
        for (Figura f: col.getFiguras())
            System.out.println(f);

        System.out.println("\n\nCirculos na Lista de Figuras:");
        for (Figura f: col.getFiguras())
            if (f instanceof Circulo)
                System.out.println(f);

        System.out.println("\n\nCentro das Figuras:");
        for (Ponto p: col.getCentros())
            System.out.println(p);
    }
}
```

Verifique se obteve o seguinte resultado:

```
true
true
false
true
true
true
false
true
true

Área Total da Lista de Figuras: 39.132741228718345

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

Problema 4.3

Utilizando os conceitos de Polimorfismo, reescreva o programa desenvolvido nas aulas anteriores relativamente ao sistema de informação para vídeo-clube.