

# Aula Prática 3

## Objectivos

- Utilização de herança simples.
- Overriding* de métodos.
- A referência `super()` e `this()`.
- Implementação dos métodos *toString* e *equals*.

## Problema 3.1

Utilizando o trabalho desenvolvido no exercício 1.2 da aula prática 1:

- a) Construa uma nova classe Estudante derivada da classe Pessoa, acrescentando os métodos e atributos necessários para aceder e guardar o número mecanográfico (int) e a data de inscrição (Data) na instituição de ensino. Note que o número mecanográfico deverá ser atribuído automaticamente (e sequencialmente a partir do 100) quando da criação de um novo estudante.

A estrutura simplificada das classes deverá ser a seguinte:

```
public class Pessoa {
    //...
    public String nome() {...}
    @Override public String toString() {...}
    // ...
}

public class Estudante extends Pessoa{
    //... definição de atributos

    Estudante(String iNome, int iBI, Data iDataNasc, Data iDataInsc);
    Estudante(String iNome, int iBI, Data iDataNasc);
                                // Nota: neste caso deve assumir a data atual

    int nMec() {...}           // Retorna o número mecanográfico
    // ... acrescentar métodos necessários
}
```

- b) Crie a classe Bolseiro que deverá incluir um atributo com o montante da bolsa. Defina novos métodos ou reescreva os métodos que julgar conveniente. Acrescente métodos get/set associados ao valor da bolsa.
- c) Implemente o método "`@ Override public String toString()`" em todas as classes. Por exemplo, para a classe Pessoa, deve retornar: "Andreia, BI: 9855678 Data: 18/7/1974"
- d) Teste o trabalho desenvolvido com o seguinte programa:

```
public class Test {
    public static void main(String[] args) {
        Estudante est = new Estudante ("Andreia", 9855678, new Data(18, 7, 1974));
        Bolseiro bls = new Bolseiro ("Maria", 8976543, new Data(11, 5, 1976));
        bls.setBolsa(745);
    }
}
```

```

        System.out.println("Estudante:" + est.nome());
        System.out.println(est);

        System.out.println("Bolseiro:" + bls.nome() + ", NMec: " + bls.nMec()
                           + ", Bolsa:" + bls.bolsa());
        System.out.println(bls);
    }
}

```

O resultado deverá ser:

```

Estudante: Andreia
Andreia, BI: 9855678, Nasc. Data: 18/7/1974, NMec: 100, inscrito em
Data: 29/9/2014
Bolseiro: Maria, NMec: 101, Bolsa:745
Maria, BI: 8976543, Nasc. Data: 11/5/1976, NMec: 101, inscrito em
Data: 29/9/2014, Bolsa: 745

```

## Problema 3.2

Utilizando Herança, reescreva o programa desenvolvido na primeira aula relativamente às figuras geométricas. Pretende-se trabalhar com as seguintes classes: Ponto, Figura, Circulo, Quadrado e Rectângulo:

- Comece por desenhar um diagrama com a relação hierárquica entre elas. Defina os atributos a incluir em cada Classe.
- Teste o trabalho desenvolvido 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);
        Circulo c3 = new Circulo (c1);

        System.out.println(c1 + "tem area: " + c1.area()
                           + " e perimetro: " + c1.perimetro());
        System.out.println(c3 + "tem area: " + c3.area()
                           + " e perimetro: " + c3.perimetro());
        System.out.println("c1 equals to c3? -> " + c1.equals(c3));           // True

        Quadrado q1 = new Quadrado(2);
        Quadrado q2 = new Quadrado(3,4,2);
        Quadrado q3 = new Quadrado(q2);

        System.out.println(q1 + "tem area: " + q1.area()
                           + " e perimetro: " + q1.perimetro());
        System.out.println(q3 + "tem area: " + q3.area()
                           + " e perimetro: " + q3.perimetro());
        System.out.println("q1 equals to q3? -> " + q1.equals(q3));           // False

        Rectangulo r1 = new Rectangulo(2,3);
        Rectangulo r2 = new Rectangulo(3,4,2,3);
        Rectangulo r3 = new Rectangulo(r2);

        System.out.println(r1 + "tem area: " + r1.area()
                           + " e perimetro: " + r1.perimetro());
        System.out.println(r3 + "tem area: " + r3.area()

```

```

        + " e perimetro: " + r3. perimetro());
    System.out.println("r2 equals to r3? -> " + r2.equals(r3));        // True
}
}

```

### Problema 3.3

Considere as seguintes entidades: Ligeiro, Motociclo, Pesado de mercadorias, Pesado de passageiros, Condutor (Pessoa com Carta). Admita ainda um conjunto de características que pode associar a estas entidades, tais como:

- Cilindrada
- Potência
- Lotação
- Tipo de Carta (A, B, C, D)
- Peso bruto

Escreva um conjunto de classes em Java, atributos, métodos e associações, que representem adequadamente estes conceitos. Teste as classes desenvolvidas.

### Problema 3.4

Utilizando os conceitos de Herança, reescreva o programa desenvolvido na aula número dois relativamente ao sistema de informação para vídeo-clube.