

Mestrado Integrado em Engenharia de Comunicações

Métodos de Programação II

1º Ano

Ano lectivo 2008/2009

1º Teste Teórico

22/4/2009 Duração: 1h30

Leia atentamente as seguintes perguntas. Tente responder claramente a todas as perguntas desenvolvendo somente aquilo que é pedido.

1. Considere a seguinte definição de classe em Java1.5:

```
public class Carro
{
    private int kms;
    private int idade;
    private double valor;
    private String matricula;

    Carro(int k, int a, double v, String m)
    { this.kms = k; this.idade = a; this.valor = v; this.matricula = m; }

    public int getKms()
    { return this.kms;}
    public int getIdade()
    { return this.idade;}
    public double getValor()
    { return this.valor;}
    public String getMat()
    { return this.matricula;}
    public void setIdade()
    { this.idade++;}
    public void setKms(int inc)
    { this.kms += inc;}

    public double valor_actual()
    { return( (this.valor) - (this.valor *(this.idade / 10)) – (this.valor *(this.kms/1 000 000) ); }

    public Carro clone()
    { return new Carro(this.kms, this.idade, this.valor, this.matricula);}
}
```

1.1 Apresente o método `equals()` para a classe `Carro`.

1.2 Escreva uma classe `Desportivo` que é subclasse de `Carro`. Neste tipo de carro regista-se também a potência do motor (em HP). Considere que o valor do `Desportivo` desvaloriza 10% por ano durante os primeiros 5 anos e 5% por ano do valor inicial durante os 20 anos seguintes (não tendo assim em conta o número de quilómetros). Apresente o construtor com os parâmetros para inicializar as variáveis de instância, o construtor de cópia (que recebe como parâmetro um objecto do tipo `Desportivo`), o método `clone()` e a nova implementação de `valor_actual()` para este tipo de carro.

1.3 Escreva uma classe `Garagem` que guarda uma colecção de carros de diferentes tipos. Nela deve constar os seguintes elementos:

- as variáveis de instância necessárias para armazenar carros,
- o construtor para criar uma nova garagem,
- um método para adicionar um novo carro à garagem,
- um método para identificar a matrícula da viatura mais antiga,
- e finalmente, um método para retornar a avaliação, em euros, da garagem em veículos armazenados.

O docente responsável

Paulo Azevedo

Correcção da Prova

1.1.

```
public boolean equals( Carro e)
{
    if(e == this) return(true);
    if(e == null) return(false);
    return( this.getKms() == e.getKms() && this.getIdade() == e.getIdade() &&
this.getValor() == e.getValor() && this.getMat().equals(e.getMat()) );
}
```

1.2.

```
public class Desportivo extends Carro
{
    private int hp;

    Desportivo(int k, int a, double v, String m, int h)
    { super(k,a,v,m); this.hp = h;}

    Desportivo(Desportivo d)
    { super(d.getKms(), d.getIdade(), d.getValor(), d.getMat()); this.hp = d.getHp();}

    public int getHp()
    { return this.hp;}

    public double valor_actual()
    { double temp = super.getValor();
      if(super.getIdade() < 6)
          temp = temp * super.getIdade() * 0.1
      if(super.getIdade() < 26)
          temp = temp * (super.getIdade() - 5) * 0.05;
      else
          temp = 0.0;

      return temp;
    }
    public Desportivo clone()
    { return new Desportivo(this); }
    // ou em alternativa:
    // public Desportivo clone()
    // {return new Desportivo(super.getKms(), super.getIdade(), super.getValor(), super.getMat(), this.hp); }
}
```

1.3.

```
public class Garagem
{
    private ArrayList<Carro> lista;

    Garagem()
    { this.lista = new ArrayList<Carro>(); }

    public void insere(Carro novo)
    { this.lista.add(novo.clone()); }

    public String maisantiga()
    {
        Carro temp = this.lista.get(0);
        for(Carro c: this.lista)
            if(c.getIdade() > temp.getIdade())
                { temp = c;}

        return(temp.getMat());
    }

    public double avaliação()
    {
        double total = 0.0;
        for(Carro c: this.lista)
            total += c.valor_actual();

        return(total);
    }
}
```