

UNIVERSIDADE DE AVEIRO
DEPARTAMENTO DE ELETRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA
ATP1 de Programação Orientada a Objetos
23 de março de 2018
Duração: 1h00

Nome _____ N° mec. _____

- I. [7] Relativamente às perguntas 1 a 14, assinale na tabela seguinte um X na coluna “V” para as declarações que estão corretas e na “F” para as que estão incorretas. Note que estas questões têm por base a linguagem Java. Cada uma destas perguntas vale 0.5 valores e cada resposta errada desconta 0,25 valores. Questões não respondidas valem 0.

	V	F
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

- Existem 3 tipos de dados primitivos em Java.
- Devido à utilização da máquina virtual de Java (JVM), é possível executar o mesmo programa em qualquer plataforma de hardware e software que possua uma versão da JVM.
- A conversão de um valor com menor capacidade de armazenamento para um valor com maior capacidade de armazenamento é automática (upcasting).
- Vários objetos de uma classe em Java têm o mesmo comportamento mas podem guardar dados diferentes.
- Os métodos da classe String do tipo `replace`, `substring`, `toLowerCase`, etc. alteram o conteúdo da própria String em que estão a ser chamados.
- O parâmetro de visibilidade `private` impede que um atributo ou método possa ser acedido fora da classe onde está implementado.
- As variáveis estáticas, ou variáveis de classe, são comuns a todos os objetos dessa classe.
- É possível definir múltiplos construtores numa classe, desde que tenham argumentos diferentes.
- Na composição de classes, criamos objetos de outras classes dentro da classe nova.
- Podemos invocar um método estático de uma classe sem instanciar um objeto dessa classe.
- A expressão `c1 == c2` verifica se as duas variáveis referenciam o mesmo objeto em memória.
- Na herança de classes criamos um tipo novo que é uma extensão da classe existente, herdando a interface da classe existente, mas adicionando código novo para acrescentar mais funcionalidades.
- A classe derivada tem acesso a todos os membros da classe base que não sejam privados.
- Uma classe em Java só pode derivar de uma classe.

II. [8] Considere os programas seguintes e indique o que é impresso no terminal (use as linhas vazias que se seguem aos programas). Não existem espaços no conteúdo a ser impresso.

```
public class XPT01b {  
    public static void main(String[] args) {  
        String x = "23,marco,2018";  
        int n = 0, p;  
        String k = x + '+' + 6 + 'k';  
        System.out.println(k);  
        String y[] = x.split(",");  
        System.out.println(y.length);  
        for(String s : y){  
            n = n + s.length();  
        }  
        System.out.println(n);  
        String z = new String();  
        for(p = 0 ; p < x.length() ; p++){  
            char c = x.charAt(p);  
            if(c != ','){  
                z = z + c;  
            }  
        }  
        System.out.println(z);  
        System.out.println(p);  
    }  
}
```

Resultado da execução do programa:

```
class OneClass{
    private String x, y;
    private static int z = 10;
    public OneClass(){
        this.x = "poo";
        this.y = "prog";
        z = z + 10;
    }
    public OneClass(String s){
        this.x = s;
        this.y = s + z;
        z = z + 10;
        System.out.println("myclass=" + z);
    }
    public String f(OneClass o){
        if(this.x.compareTo(o.x) < 0)
            return this.x;
        else
            return o.y;
    }
    public static int g(){
        return z;
    }
    public String toString(){
        return this.x + "-" + this.y;
    }
}
public class XPT02b {
    public static void main(String[] args) {
        OneClass obj1 = new OneClass("ola");
        OneClass obj2 = new OneClass("aveiro");
        System.out.println(obj1.f(obj2));
        System.out.println(obj2.f(obj1));
        System.out.println(OneClass.g());
        System.out.println("'" + obj1 + "'," + obj2);
    }
}
```

Resultado da execução do programa:

III. [5] Considere as classes declaradas abaixo e o programa de teste (tabela à esquerda). Tenha em atenção o que foi impresso depois da execução do programa (tabela à direita) e inclua o código necessário nos espaços em branco para que seja possível a sua execução. Não acrescente métodos nem atributos.

```
class MyClass1b{
    private double s;
    public MyClass1b(double s){

        _____ = s;}
    public void f(){

        this.s += _____;}
    public double getS(){

        return _____;}

    public String _____(){

        return _____;}
}
class MyClass2b extends MyClass1b{
    private int x;
    public MyClass2b(double s, int x){

        _____;

        _____ = x;}
    public MyClass2b(){

        _____;

        _____ = 5;}
    public void f(){

        _____;

        this.x += _____;}

    public double _____(){
        return this.x;}

    public String _____(){

        return _____;}
}
```

```
public class XPT03b {
    public static void main(String[] args) {
        MyClass1b obj1 = new MyClass1b(25.30);
        System.out.printf("%.1f\n", obj1.getS());
        MyClass1b obj2 = new MyClass2b(10.0, 5);
        System.out.printf("%.1f\n", obj2.getS());
        MyClass2b obj3 = new MyClass2b();
        System.out.printf("%.1f\n", obj3.getX());
        System.out.printf("%.1f\n", obj3.getS());
        System.out.println(obj1);
        System.out.println(obj2);
        System.out.println(obj3);
    }
}
```

```
terminal> java XPT03b
```

```
25.3
10.0
5.0
5.1
MyClass1b:25
MyClass2b:10.0,5
MyClass2b:5.1,5
```