



Universidade Federal da Fronteira Sul
Ciência da Computação
Programação I

Avaliação 1

Nome: _____ Data: _____

1. A prova pode ser feita a lápis, porém o professor se dará ao direito de não aceitar reclamações relativas à correção.
2. Início da prova 07h30, término 10h00. Manter celulares desligados!
3. Coloque o seu nome nas folhas de resposta.
4. A compreensão das questões faz parte da prova.

Boa prova!

Questões

1) (1,0) Dado o código das classes abaixo, qual será a saída do programa na tela? Justifique.

```
class Tester {
    private int var;

    public Tester(double var) { this.var = (int)var; }
    public Tester(int var) { this("hello"); }
    public Tester() { System.out.println("good-bye"); }

    public Tester(String s) {
        this();
        System.out.println(s);
    }
}

class MainTester {
    public static void main(String[] args) {
        Tester t = new Tester(5);
    }
}
```

2) (1,0) Dado o código abaixo:

```
interface X { void go(); }
interface Y { void somewhere(); }
abstract class B {
    protected int getHash() { return 89; }
    abstract protected void now();
    abstract public void rightNow();
}
```

Implemente o código da classe chamada C, que deve herdar de B e implementar as interfaces X e Y. Você pode deixar o corpo dos métodos sem código. Ex.: `void teste() { ... }`

3) (1,5) Dadas as classes abaixo:

<pre>class Ferramenta { private String marca; public Ferramenta() { this.marca="Nenhuma"; } public String getMarca() { return this.marca; } public void setMarca(String s) { this.marca = s; } public void ativar() { Sys.out.println("Ativada!"); } public String getInfos() { return "Tool " + this.marca; } public void imprimeDesc() { Sys.out.print("M: "+ this.marca); } }</pre>	<pre>class Martelo extends Ferramenta { public Martelo(String s) { super(); setMarca(s); } public void ativação() { ativar(); Sys.out.println("Martelando!"); } public void imprime() { Sys.out.print(getMarca() + " mar."); } }</pre>
--	---

e o seguinte código:

```
Ferramenta f = new Ferramenta();
Martelo m = new Martelo("Black&Decker");

f.setMarca("Tramontina");
```

mostre o que será imprimido na tela para cada uma das chamadas abaixo:

Comando	Impressão na tela
<code>System.out.println(f.getMarca());</code>	
<code>System.out.println(m.getMarca());</code>	
<code>f.ativar();</code>	
<code>System.out.println(m.getInfos());</code>	
<code>m.ativação();</code>	
<code>f.imprimeDesc();</code>	
<code>m.imprimeDesc();</code>	
<code>m.imprime();</code>	

4) (1,5) Dado o código abaixo, qual a saída do programa na tela? Justifique sua resposta desenhando e explicando a representação na memória dos elementos do programa:

```
class MyClass {
    public static void main(String[] args) {
        double d = 2.7;
        Box b = new Box();
        b.doMinus(d);
        System.out.println(d);
        System.out.println(b.explain());
    }
}

class Box {
    public double d;
    private int i;

    public void doMinus(double d) {
        d = d - 10;
        i = 9;
    }
    public String explain() {
        return "Box: " + this.i + " " + this.d;
    }
}
```

5) (1,5) Preencha as lacunas de acordo com os conceitos ou termos da programação orientada a objetos.

- a. Uma _____ serve como um modelo para a criação de objetos.
- b. O ato de alocar um espaço de memória para um objeto de uma determinada classe através do operador `new` chama-se _____.
- c. Os membros de uma classe que definem seu estado/características são os _____.
- d. Os membros de uma classe que definem seu comportamento e/ou funcionalidades são os _____.
- e. O _____ é um método especial, público e sem retorno, executado sempre no momento da criação de um objeto.
- f. O mecanismo de _____ permite que uma classe reutilize as propriedades e/ou métodos já definidos em outra classe mais genérica. A classe que recebeu as características chama-se _____ e a classe que foi estendida chama-se _____.
- g. Um membro precedido do modificador _____ é visível somente dentro da classe onde foi declarado. Um membro _____ é visível na própria classe, em suas subclasses ou outras classes do mesmo pacote. Já um membro _____ é acessível por qualquer outra classe de qualquer pacote.
- h. O conceito de _____ diz que uma classe deve funcionar como uma caixa-preta: não precisamos conhecer os detalhes internos de sua implementação, apenas conhecer sua interface pública.
- i. Os atributos precedidos do modificador _____ possuem um valor que é compartilhado por todas as instâncias de uma classe. Já os métodos que possuem o mesmo modificador não exigem a instanciação de um objeto, podendo ser chamados diretamente através do nome da classe.
- j. O operador _____ permite referenciar membros da própria classe, sendo utilizado, por exemplo, para resolver ambiguidades entre nomes do membro e de parâmetros com o mesmo nome.
- k. Uma classe que possui a palavra-chave _____ na sua declaração não pode ser instanciada.
- l. Um atributo precedido da palavra _____ não pode ter seu valor alterado em tempo de execução.

6) (2,0) O código abaixo contém as propriedades que descrevem um post de um blog. Não há o código dos métodos. Escreva o código dos métodos necessários para ambas as classes e escreva um programa que instancie um objeto Post, leia do teclado as informações necessárias para preencher todas as propriedades do objeto e, por último, imprima essas informações através do método `imprimeResumo()`. As classes **devem** conter métodos getters/setters.

```
class Autor {
    private String nome;
    private int id;
}
class Post{
    private Autor autor;
    private String texto;

    // Imprime todas as propriedades da classe no formato:
    public void imprimeResumo() {}
}
```

7) (1,5) Implemente uma classe chamada Parser que contenha um método estático `analize()`. O método deve receber como parâmetro um vetor de Strings e retornar quantas vezes a palavra "token" ou qualquer outra palavra que inicie pela letra 't' aparem no vetor.