Universidade Federal da Fronteira Sul
Ciência da Computação
Programação I

Avaliação 1

Nome: _____

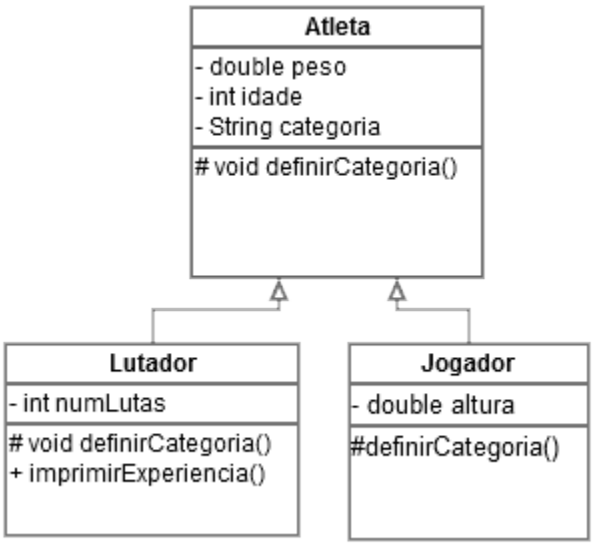
Data: _____

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) (2,5) Observe o diagrama UML abaixo. Implemente as três classes, criando os getters e setters e observando as descrições dos métodos abaixo:



`definirCategoria()`:

- na superclasse, este método deve definir a categoria de acordo com a idade (até 10 anos "infantil", de 11 a 18 "juvenil" e acima disso "adulto".
- para um `Lutador`, a classificação deve se dar de acordo com o peso (até 54 "pluma", de 55 a 60 "leve", de 61 a 75 "meio-leve", acima disso "pesado").
- para um Jogador, a categoria deve respeitar a altura (acima de 1,90 "atacante", abaixo disso "levantador").

`imprimirExperiencia()`:

- método específico da classe `Lutador`. Deve imprimir o nível de experiência baseado no número de lutas (menos de 10 "iniciante", entre 11 e 20 "intermediário' e acima disso "experiente").

2) (1,5) Dadas as classes abaixo:

```
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);
  }
}
```

```
class Martelo extends Ferramenta {

  public Martelo() {
    super();
  }

  public void ativar() {
    super.ativar();
    Sys.out.println("Martelando!");
  }

  public void imprimeDesc() {
    Sys.out.print(getMarca() + " mar.");
  }
}
```

e o seguinte código:

```
Ferramenta f = new Ferramenta();
Martelo m = new Martelo();

f.setMarca("Tramontina");
m.setMarca("Black&Decker");
```

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

Comando	Impressão na tela
`System.out.println(f.getMarca());`	
`System.out.println(m.getMarca());`	
`f.ativar();`	
`System.out.println(m.getInfos());`	
`m.ativar();`	
`f.imprimeDesc();`	
`m.imprimeDesc();`	

3) (2,0) Dado o código da classe abaixo:

```
class BlurayDisc {
      private String rotulo;
      protected int id;
      protected float densidade;
      public char[] dados;

      public BlurayDisc(String rotulo, int id) {
            setIdentificação(rotulo, id);
            this.dados = new char[4096];
      }

      public void setIdentificação(String rotulo, int id) {
            this.rotulo = rotulo;
            this.id = id;
      }

      private String getRotulo() {
            return this.rotulo;
      }
}
```

Dadas as variáveis abaixo, indique se cada uma dos comandos seguintes funciona. Se houver algum erro nesses comandos, diga qual é esse erro (e justifique).

```
BlurayDisc c;
BlurayDisc h = new BlurayDisc("Teste", 1);
String st = "teste";
float num;
```

a) `c.dados[10] = ' i ';`
b) `h.setIdentificacao(st, num);`
c) `c.setIdentificacao(12, "Outro rótulo");`
d) `c = new BlurayDisc();`
e) `h.rotulo = "Outro rótulo";`
f) `num = 2.34f;`
g) `num = h.densidade;`
h) `st = h.getRotulo();`
i) `c = new BlurayDisc(st, 234);`
j) `c.dados[100] = "Dado posição 100";`

4) (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 herdou 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. O operador _____ permite acessar membros da superclasse.

5) (2,5) A classe abaixo representa uma compra num software de comércio. A classe está com o código de seus métodos incompleto. Escreva o código faltante dos métodos da classe `Compra` e escreva um programa em Java que instancie um objeto da classe `Compra`, leia do teclado as informações necessárias para preencher todas as propriedades do objeto e, por último, imprima essas informações através do método `imprimeResumo()`.

```
class Compra {
      private String nomeProduto;
      private float preço;
      private int quantidade;

      public Compra(String produto, float preço, int quantidade) {}

      public void setNomeProduto(String novo) {}

      public String getNomeProduto() {}

      public void setPreço(float preço) {}

      public float getPreço() {}

      public void setQuantidade(int quantidade) {}

      public int getQuantidade() {}

      // Imprime todas as propriedades da classe
      public void imprimeResumo() {}
}
```