



Universidade Federal da Fronteira Sul
Ciência da Computação
Programação I

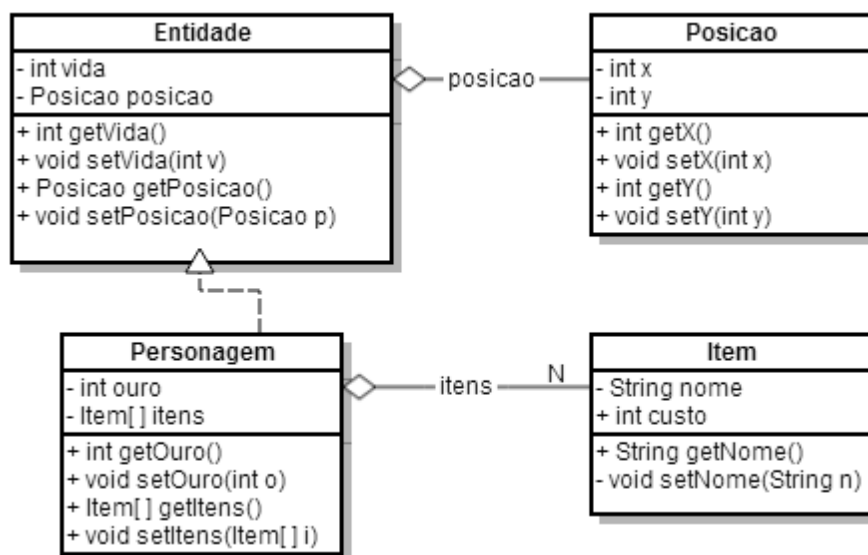
Avaliação 1

Nome: _____ Data: 17/06/2013

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 7h30, 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) (2,0) Dado o diagrama UML, implemente os seguintes métodos:

a) `reposicionaInativos()` - método *estático* pertencente à classe `Utils`. Recebe como parâmetro um vetor de objetos `Personagem`; percorre esse vetor alterando a posição `x` e `y` para `-1` e `-1` de todos os objetos cuja vida seja menor que zero, ou a posição `x` seja maior que 640 ou a posição `y` seja maior que 480. O método não retorna valor.

b) `podeUsarItem()` - método pertencente à classe `Personagem`. Recebe como parâmetro uma `String` que representa o nome de um `Item`. O método retorna `true` se o personagem possui, em seu vetor de itens, um item cujo nome foi informado E se o personagem possui ouro suficiente para pagar pelo custo desse item a ser usado. Ex.: se o personagem possui 10 de ouro e o custo do item é 5, então o item pode ser usado. Se o personagem não possui o item em questão ou não possui ouro suficiente, o método retorna `false`.

2) (1,5) Utilize o diagrama UML da questão 1, sendo que para a classe `Personagem` assumo que a propriedade `itens` (e seus getters/setter) **não existe mais**. Escreva os métodos **construtores** necessários para as classes para que o trecho de código abaixo funcione e inicialize **todas** as propriedades dos objetos instanciados. Você **não** pode criar métodos que não sejam construtores (como getters e setters), apenas usar os já existentes.

```
int x = 5, y = 6, ouro = 10, vida = 500;
Entidade e = new Entidade(x, y, vida);
Entidade e2 = new Entidade(); // inicializa todas os atributos com zero
Personagem p = new Personagem(ouro, vida, x, y);
Personagem p2 = new Personagem(ouro, vida); // inicializa a posição em (0,0)
```

3) (2,0) Dado o código abaixo, qual a saída do programa na tela? Justifique sua resposta desenhando a representação na memória dos objetos e variáveis do programa:

```
class Main {
    public static void main(String[] args) {
        double value = 3.0;
        Sprite a, b, c, d;
        a = new Sprite();
        b = new Sprite();
        c = Sprite.newOne(b);
        a.reset(value);
        c.reset(45.6);
        b.id = 33;
        d = a;
        d.value = value;
        System.out.println(value);
        System.out.println(a.debug("a"));
        System.out.println(b.debug("b"));
        System.out.println(c.debug("c"));
        System.out.println(c.debug("d"));
    }
}

class Sprite {
    public double value;
    public int id;

    public static Sprite newOne(Sprite a) {
        a = new Sprite();
        Sprite b = new Sprite();
        return b;
    }

    public void reset(double v) {
        double value = 3.2;
        this.value = value + v;
        id = 9;
    }

    public String debug(String n) {
        return "Sprite " + n + ": value = " + this.value + ", id = " + this.id;
    }
}
```

4) (2,0) Utilize o diagrama UML da questão 1, sendo que para a classe `Personagem` assumo que a propriedade `itens` (e seus getters/setter) **não existe mais**. Faça um programa que instancie um objeto `Personagem` e mostre na tela todos os seus atributos. Os valores para cada um dos atributos do personagem deve ser lido do teclado.

5) (2,5) Dadas as classes abaixo, mostre o que será impresso na tela quando o programa Main for executado.

<pre> class Sprite { private int x; public int scaleY; public Sprite (int x, int s) { this.x = x; this.scaleY = s; } public int getX() { return this.x; } public void setX(int i) { this.x = i; } public void invertre() { this.x = -this.scaleY; System.out.println("x = " + getX()); } public String resumo() { return x + " " + scaleY; } public void imprime() { System.out.println(getX() + " " + scaleY); } } </pre>	<pre> class MovieClip extends Sprite { private float alpha; public MovieClip(float h) { super(10, 22); alpha = h; setX(12); } public void negativo() { invertre(); alpha = 0; System.out.println("Negativo!"); } public String resumo() { return "Movie: " + super.resumo(); } public void imprime() { System.out.println(getX() + " " + alpha); } } </pre>
<pre> class MovieBlock extends MovieClip { public MovieBlock(float h) { super(h); setX(40); } public String resumo() { return "Block: " + super.resumo(); } public void imprime() { super.imprime(); System.out.println("Block "+getX()); } public void invertre() { setX(900); System.out.println("b invertre"); } } </pre>	<pre> class Main { public static void main(String args[]) { Sprite s = new Sprite(1, 2); MovieClip m = new MovieClip(5.5f); MovieBlock b = new MovieBlock(7.3f); s.resumo(); s.invertre(); s.imprime(); m.imprime(); m.negativo(); m.resumo(); m.invertre(); System.out.println("b.x = " + b.getX()); b.imprime(); b.negativo(); b.imprime(); } } </pre>