

Universidade Federal de Pernambuco  
Centro de Informática

Paradigmas de Linguagens de Programação - Segunda Prova

André Santos  
04 de agosto de 2025

Nome: \_\_\_\_\_  
CPF: \_\_\_\_\_

1. (2,5 pontos) Dadas as seguintes classes:

```
public class Classe1 {  
    private static int x = 3;  
    private int y = 2;  
    public static void setX(int i) { x = i; }  
    public void setY(int i) { y = i; }  
    public static int getX() { return x; }  
    public int getY() { return y; }  
    public int function(int x) { return (this.x * y); }  
}  
  
public class Classe2 extends Classe1 {  
    public int function(int x) { return (getX() + getY() + x); }  
    public String function2(int x) { return "Hello"; }  
}  
  
public class Teste1 {  
    public static void main(String args[]) {  
        Classe1 a = new Classe1();  
        Classe1 b = new Classe2();  
        Classe2 c = new Classe2();  
        a.setX(5); b.setX(7); c.setY(9);  
        a.setY(4); b.setY(6); c.setY(8);  
        System.out.println("a.function = " + a.function(10));  
        System.out.println("b.function = " + b.function(11));  
        System.out.println("c.function = " + c.function(12));  
        a = b;  
        b = c;  
        System.out.println("a.function = " + a.function(13));  
        System.out.println("b.function = " + b.function(14));  
    }  
}
```

informe as linhas que serão impressas ao executar o programa Teste1:

```
a.function =  
b.function =  
c.function =  
a.function =  
b.function =
```

2. (2,5 pontos) Comparando os modificadores de acesso **public**, **protected** e **private** em Java, preencha a tabela abaixo, marcando V ou F nas afirmações verdadeiras ou falsas.

	<b>public</b>	<b>protected</b>	<b>private</b>
<b>permite</b> que um método ou atributo seja acessado somente na própria classe			
<b>permite</b> que um método ou atributo seja acessado por qualquer classe			
<b>permite</b> que um método ou atributo seja acessado por suas subclasses			
<b>impede</b> que um atributo seja alterado			

3. (2,5 pontos) Comparando Classes, Classes Abstratas e Interfaces em Java, preencha a tabela abaixo, marcando V ou F nas afirmações verdadeiras ou falsas.

	Classes	Interfaces	Classes Abstratas
<b>podem</b> declarar métodos sem sua implementação			
é possível criar objetos com seu tipo, usando a palavra reservada "new"			
<b>podem</b> implementar construtores			
<b>tem</b> que declarar pelo menos um atributo			

4. (1,0 ponto) Sobre generics em Java podemos afirmar (marque V ou F, verdadeiro ou falso):

- (      ) serve somente para declarar atributos genéricos;
- (      ) podem ser usados para declarar métodos com tipos genéricos;
- (      ) permite criar classes com tipos genéricos;
- (      ) são usados para declarar e tratar exceções;
- (      ) só podem ser usados com métodos estáticos;

5. (1,5 pontos) declare e implemente uma exceção **ParametroZeroException** e modifique o código de **classe2** na primeira questão para que essa exceção que você criou seja levantada quando o parâmetro passado para **function2** for igual a zero.