

Universidade Federal de Pernambuco
Centro de Informática

Paradigmas de Linguagens de Programação - Segunda Prova

André Santos
19 de Abril de 2023

Nome: _____ CPF: _____

1. (2,0 pontos) Dada a seguinte classe:

```
public class Classe1 {  
    private static int x = 40;  
    private int y = 70;  
    public static void setX(int i) { x = i; }  
    public void setY(int i) { y = i; }  
    public int getY() { return y; }  
    public static int getX() { return x; }  
}
```

Qual a saída do seguinte programa:

```
public class Teste1 {  
    public static void main(String args[]) {  
        Classe1 a = new Classe1();  
        Classe1 b = new Classe1();  
        a.setY(5);  
        b.setY(6);  
        a.setX(1);  
        b.setX(2);  
        System.out.println("a.y = " + a.getY());  
        System.out.println("b.y = " + b.getY());  
        System.out.println("a.x = " + a.getX());  
        System.out.println("b.x = " + b.getX());  
        System.out.println("Classe1.x = " + Classe1.getX());  
    }  
}
```

a) a.y = 5 b.y = 6 a.x = 1 b.x = 2 Classe1.x = 40	b) a.y = 5 b.y = 6 a.x = 40 b.x = 40 Classe1.x = 40	c) a.y = 5 b.y = 6 a.x = 2 b.x = 2 Classe1.x = 2	d) a.y = 5 b.y = 6 a.x = 1 b.x = 2 Classe1.x = 2
---	---	--	--

2. (1,0 ponto) Se as três ocorrências da palavra reservada "static" forem retiradas do código de Classe1, o que aconteceria?
- a) nem a classe Classe1 nem a classe Teste1 compilam
 - b) os códigos de Classe1 e de Teste1 compilam mas ocorre um erro na execução
 - c) o código de Classe1 compila, mas ao tentar compilar o código de Teste1 ocorrerá um erro [Não é possível acessar Classe1.getX sem que getX seja estático]
 - d) Classe1 e Teste1 compilam corretamente e geram o mesmo resultado da questão anterior
 - e) Classe1 e Teste1 compilam corretamente e geram um resultado diferente da questão anterior

3. (2,0 pontos) O código abaixo troca a posição de dois elementos (índice x e índice y) de um array de inteiros. Com base nele, mostre que alterações seriam necessárias no código do método trocaposicao para termos uma versão dele usando *generics*, de forma que um mesmo código possa funcionar para arrays de Integer, Boolean, Character, Double etc.

```
public class TesteArray {
    public static int[] trocaposicao(int v[], int x, int y) {
        int w = v[x];
        v[x] = v[y];
        v[y] = w;
        return v;
    }
    // resposta:
    public static <T> T[] trocaposicao(T v[], int x, int y) {
        T w = v[x];
        v[x] = v[y];
        v[y] = w;
        return v;
    }
}
```

4. (1,0 ponto) O trecho de código abaixo cria um objeto array e dois objetos String.

```
...
String[] students = new String[10]; [uma referência criada]
String studentName1 = "Peter Parker"; [uma referência criada]
String studentName2 = "Clark Kent"; [uma referência criada]
students[0] = studentName1; [uma referência criada]
students[1] = studentName2; [uma referência criada]
studentName1 = null; [uma referência removida]
```

Quantas referências para esses objetos existem depois que esse trecho de código é executado?

- a) 11 b) 3 c) 4 d) 5 e) 12

5. (3,0 pontos) Comparando Classes, Classes Abstratas e Interfaces em Java, preencha a tabela abaixo com as respostas com suas características, respondendo **Sim** ou **Não**. [0,2 pontos por cada resposta correta]

	Classes	Classes Abstratas	Interfaces
podem declarar atributos	Sim	Sim	Não
podem declarar métodos sem sua implementação	Não	Sim	Sim
podem declarar métodos e sua implementação	Sim	Sim	Não
permitem sobrescrever métodos herdados de uma superclasse	Sim	Sim (*)	Não
permitem que objetos sejam criados com seu tipo, usando a palavra reservada "new"	Sim	Não	Não

(*) nesse item foi aceita qualquer resposta pois não falamos dessa possibilidade nas aulas.

6. (1,0 ponto) Sobre Exceptions em Java podemos afirmar que (marque todas as corretas, se houver): [0,2 pontos por cada item respondido corretamente]
- a) algumas exceções são pré-definidas, como RuntimeException e NullPointerException;
 - b) novas exceções podem ser criadas através de classes que implementam a interface Exception; [errado: deve herdar da classe Exception]
 - c) o programador pode criar novas exceções verificadas;
 - d) o programador pode criar novas exceções não verificadas; [para o item (d) foi aceita qualquer resposta pois não vimos esse assunto]
 - e) os métodos não precisam informar na sua declaração se podem gerar exceções verificadas; [errado: devem informar com a palavra *throws* seguida das exceções que podem ser levantadas]