

Lista 6 - Programação de Computadores

Ayron Luigi de paiva (ayronpaiva@outlook.com)

31 de março de 2021

1. Responda as perguntas abaixo:

a) Explique com suas palavras por que uma classe abstrata não pode ser instanciada

A classe abstrata não pode ser instanciada pois ela é uma classe apenas de referência de outras, e não poderá ter um objeto instanciado, sendo caracterizada como uma classe complementar .

b) Explique, com suas palavras, por que interfaces não podem ter construtores.

Uma interface não possui uma implementação instanciada, portanto, nenhum construtor, além de possuir somente métodos referêciais, sua funcionalidade é parecida com os métodos abstratos.

c) Explique com suas palavras por que não podemos ter construtores declarados com a palavra-chave abstract.

Primeiramente para criar um construtor o mesmo deve ter o nome da classe referente, e nome de Classe sempre começara com letra maiúscula portanto abstract seria inválido, em segundo que a palavra abstract é reservada para indicar classes abstratas, assim proibida de ser nomes de métodos ou variáveis.

2. Identifique e explique o(s) erro(s) na classe abaixo:

```
1 public class Produto {  
2     private String identificacao;  
3     private double custoDeFabricacao;  
4     Produto(String i, double c) {  
5         identificacao = i;  
6         custoDeFabricacao = c;  
7     }  
8     abstract public String toString();  
9     abstract public void novoCusto(double nc);  
10 }
```

Código 1: Classe Produto

Visto que Produto não é uma classe abstrata, não poderemos ter métodos abstratos, portanto a linha 8 e 9 apresentam erros, outro ponto a se observar que não é um erro mas sim uma boa prática de programação, seria a utilização de this, em frente às variáveis instanciadas, portanto as linhas 5 e 6 deveriam ficar this.identificacao=i; e this.custoDeFabricacao=c;

3. Identifique e explique o(s) erro(s) na classe abaixo:

```

1 public abstract class Dispositivo {
2     private String nome;
3     private long capacidadeEmBytes;
4     public Dispositivo(String n, long c) {
5         nome = n;
6         capacidadeEmBytes = c;
7     }
8     abstract public String toString();
9     abstract public double capacidadeEmMegabytes();
10 }

```

Código 2: Classe Dispositivo

```

1 public class DiscoOtico extends Dispositivo {
2     public DiscoOtico(long c){
3         super("Disco tico ", 241172480L);
4     }
5     public String toString() {
6         return "Dispositivo:" + nome + " Capacidade:" + c;
7     }
8 }

```

Código 3: Classe DiscoOtico

O primeiro erro (linha 1) identificado está na classe filha DiscoOtico onde que ao tentar fazer a herança de Dispositivo a mesma obriga a criação de 2 métodos obrigatórios na classe (linha 8 e 9) e no caso DiscoOtico não possui o método capacidadeEmMegabytes acarretando em um erro, o outro possível erro está no método toString da classe filha também, onde são chamadas as variáveis nome e capacidade da classe herdada, mas as mesmas são do atributo private e não possuem o encapsulamento correto, impossibilitando o uso, o certo seria se na classe Dispositivo fosse implementados os métodos getter e setters.