

OBS: As questões cuja resposta não tem referência, são de elaboração própria

1.1. Subclasse / Superclasse

1.2. Reutilização de código

1.3. Atributos / Métodos

1.4. Subclasse / Superclasse

2. Para indicar que a classe herda um construtor de alguma outra classe, no caso de uma classe que “não herda de ninguém” o super aparece para indicar que ela herda de “Object” em java. O `super()`; pode conter parâmetros que se encaixem com os de um dos métodos construtores da Superclasse.

3. O erro evidente é que a superclasse “Conta” possui um construtor que recebe 2 parâmetros além de um erro com variável, atribuindo-a com um desses valores e na subclasse “Poupança” o super recebe 2 parametros porem o próprio construtor da classe não recebe esses mesmos 2 parâmetros, apenas 1. A correção seria tanto atribuir os parâmetros aos atributos utilizando o `this`. tanto passar mais um parâmetro no construtor da subclasse.

6.1. DEmary

6.2. Pai

6.3. As superclasses possuem atributos e métodos de mesmo nome mas com valores diferentes, o que pode causar confusão na hora de utilizar os conceitos de herança. Logo quando queremos acessar um método ou um atributo da superclasse o python utiliza a precedência na ordem em que aparece declarada.

7. A sobrecarga de métodos em java diz respeito a escrita de um mesmo método que tem assinatura diferente e/ou ações diferentes, já a sobreposição é quando se altera o conteúdo de um método já predefinido, alterando seu comportamento. Polimorfismo e herança.

8. Não, seria praticamente como declarar dois atributos com o mesmo nome em uma mesma classe, o que é praticamente impossível. Por exemplo: `int num`, `double num`. Não identificaria qual dos atributos estaríamos utilizando no tempo de execução, já que atributos não possuem assinatura, não existe sobrecarga de atributo.

9.1. Se ela deve implementar um método ela precisa herdar de uma classe e não pode ser uma classe abstrata, sendo assim obrigada a fazer um `@Override` em um método predefinido em uma classe.

9.2 Definir um método intermediário que permita a chamada desse método indiretamente.

10. `toString()`; é um método que pode ser implementado em todas as classes, ele pode ser sobrescrito e sua função é retornar uma string de uma classe. Object em java é a classe da

qual todas as classes herdam, o toString é um método definido em object, portanto todas as classes o possuem mesmo que não tenha sobrescrito.

11.1 Deve ser sobrescrito quando há a necessidade de representar a classe em uma string específica.

```
@Override  
public static void toString(){  
    return "Atributo: " + atributo  
}
```

11.2. Quando por exemplo a superclasse possui um método generalizado para um comportamento mas ele não é apropriado para subclasse. Por exemplo a classe Animal; Todos os animais fazem algum som, mas um cachorro late.

11.3 Uma classe que herda de uma classe abstrata deve implementar o método abstrato da superclasse, logo é necessário a sobrescrita do método, seja qual for o comportamento.

12. Por que cupom promocional é uma constante, portanto não pode ser alterada, portanto não possui um método set().

13. Não codifiquei por que é paia, só pra alterar a forma como manipula pra ArrayList.