



UPskill - JAVA

Linguagem de Programação Java

EXERCÍCIO TRIBUTÁVEIS

Considerando o projeto **Maven** fornecido, chamado **Tributaveis**, faça o seguinte:

1. Analise a classe **Veiculo** (simplificada) para representar veículos caracterizados pela matrícula, cilindrada e cor.
2. Na classe principal, chamada **MainTributaveis**, **crie e mostre** no ecrã uma instância da classe **Veiculo** com matrícula **22-33-CC**, **vermelho** e com cilindrada de **1000 cc**.
3. Crie a interface Java **Cores**, para representar o conjunto de cores: {azul, cinzento, vermelho, verde}.
4. Crie e mostre no ecrã uma nova instância da classe **Veiculo** com matrícula **44-55-DD**, **azul** e com cilindrada de **2500 cc**. Use a interface **Cores** para definir a cor do novo veículo.
5. Implemente a interface **Cores** na classe **Veiculo**.
6. Represente a classe **Veiculo** e a interface **Cores** num diagrama de classes, em notação UML.
7. Crie e mostre no ecrã uma nova instância da classe **Veiculo** com matrícula **11-22-BB**, **verde** e com cilindrada **1400 cc**. Especifique a cor do novo veículo através da classe **Veiculo**.
8. Analise a classe **Moradia** (simplificada) fornecida para representar moradias caracterizadas pela morada, área e cor.
9. Implemente a interface **Cores** na classe **Moradia**.
10. Atualize o diagrama de classes.
11. Crie e mostre no ecrã uma instância da classe **Moradia** com a morada **Rua do Bocage**, com cor **cinzenta** e com área de **90** metros quadrados.
12. Crie e imprima o conteúdo de uma **variável** capaz de guardar **qualquer referência** das instâncias existentes.
13. Altere as classes, **Veiculo** e **Moradia**, de forma a facilitar, através do **polimorfismo**, o cálculo do **valor do imposto** aplicado às instâncias dessas classes. Considere as seguintes formas de calcular o valor do imposto de um:
 - **Veiculo**: 15 € para cilindrada inferior a 1500 cc e 40€, caso contrário. Assumir que estes valores poderão ser alterados no futuro;
 - **Moradia**: igual ao dobro da área em metros quadrados.

14. Crie um **contentor** de objetos do tipo **array**, chamado *tributaveis*, para armazenar todas as instâncias existentes.
15. Guarde no contentor todas as instâncias criadas.
16. Mostre no ecrã o **total do imposto** aplicado às instâncias do contentor.
17. Crie e invoque um **método** para imprimir no ecrã o imposto aplicado a um objeto recebido por parâmetro, do tipo **Veiculo** ou **Moradia**.
18. **Crie**, de forma automática, uma classe de **teste** para a classe **Moradia**. Verifique a criação da classe **MoradiaTest** na pasta **Test Packages**. Esta classe inclui já alguns métodos de teste.
19. **Altere** o método de teste do cálculo do imposto, de forma a testar o cálculo do imposto a pagar por uma moradia com área de 90 metros quadrados.
20. **Execute** o teste à classe **Moradia** e verifique o resultado obtido.
21. **Analise** a cobertura de testes através do *plugin* Maven **JaCoCo** (Java Code Coverage). Para isso, acrescente no ficheiro **pom.xml**, o elemento **<build>** que se encontra a seguir.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.7.7.201606060606</version>
      <executions>
        <execution>
          <goals>
            <goal>prepare-agent</goal>
          </goals>
        </execution>
        <execution>
          <id>report</id>
          <phase>prepare-package</phase>
          <goals>
            <goal>report</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```