

Suporte para GitHub:

Carregando arquivos pelo navegador: https://drive.google.com/open?id=1Klf0HCJcB_4Q5B7efMXrs_YYDXGwH89U

Tutorial Basico GitHub com Eclipse e EGit Usando Chave SSH: <http://www.youtube.com/watch?v=fFB5azTSGZw>

Usando Github com Github Desktop em Projetos Eclipse: <http://www.youtube.com/watch?v=EgHljYyS4U>

Usando Github com SSH no Terminal Linux com chave gerada no Eclipse: <http://www.youtube.com/watch?v=0s699q5Sja4>

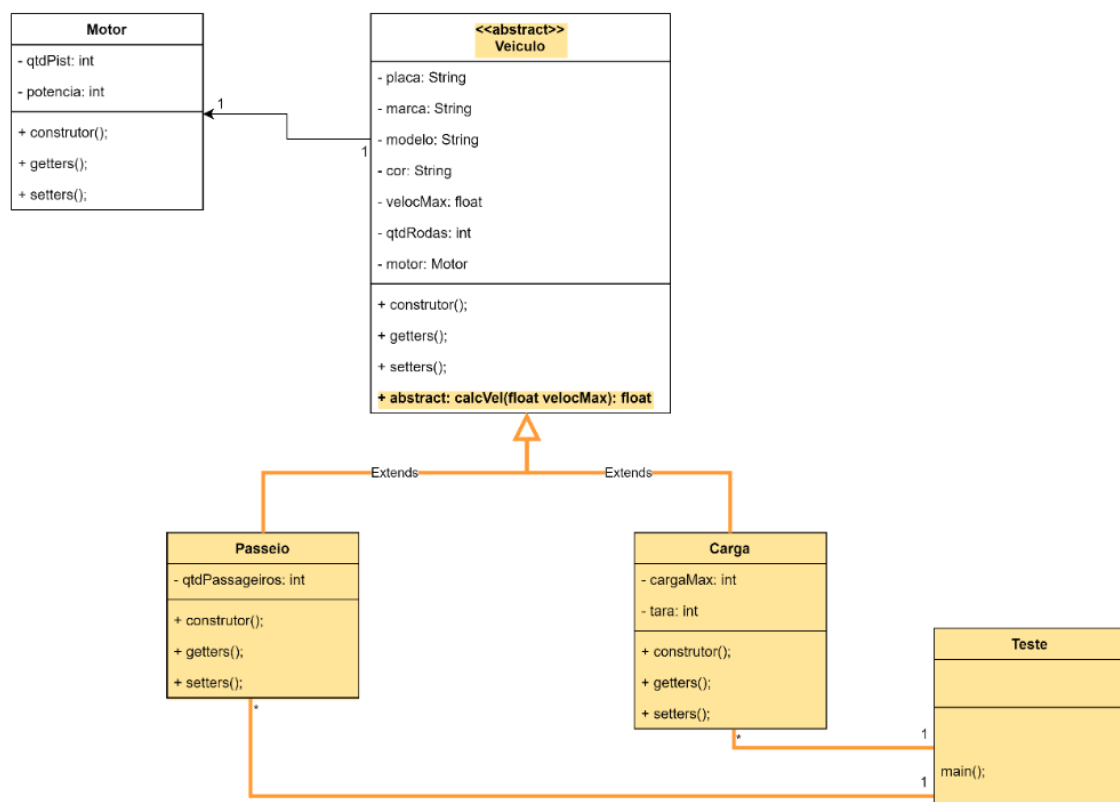
Usando Github com SSH no Terminal Windows com chave gerada no Eclipse: <http://www.youtube.com/watch?v=DaydwPB2WSI>

Para cada exercício, criar um projeto no Github para submeter à tarefa.

Para cada exercício, criar uma classe no package view que se possa testar as implementações que se foram pedidas nos objetos

Modelagens devem estar em um arquivo em uma pasta no projeto que será submetido ao Github

1. Criar um Java Project para implementar o diagrama abaixo, onde a classe teste deve implementar pelo menos 2 veículos de passeio e 2 de carga:



Considere as seguintes limitações:

- A velocidade dos veículos de passeio se dá em metros/hora (1 km/h = 1000 m/h)
- A velocidade dos veículos de passeio se dá em cm/hora (1 km/h = 100000 cm/h)
- Garanta que nunca ocorra:
 - As classes **Passeio** e **Carga** nunca devem ser extendidas
 - Nenhum setter, de nenhuma classe poderá ser sobrescrito

2. Considere uma aplicação em Java que gerencie a contratação de funcionários. A empresa tem 3 tipos de funcionários: Funcionários de Tempo Integral, de Tempo Parcial e Temporários.

- Todos os funcionários tem id, nome e salário
 - Funcionários de Tempo integral tem banco de horas;
 - Funcionários de Tempo Parcial tem hora de entrada e hora de saída;
 - Funcionários Temporários tem dia, mês e ano para o fim do contrato;
- A classe de controle tem a operação de contratação, que, inicializa o funcionário com seus atributos e retorna o funcionário
- A classe main chama o novo funcionário e imprime seus atributos, simulando a gravação em um BD.

3. Considere uma aplicação feita para calcular impostos urbanos nas cidades do país. O cálculo não é muito complexo, mas varia de cidade para cidade. Inicialmente, se fará para São Paulo, Belo Horizonte, Porto Alegre e Curitiba.

- São Paulo → $\text{Imposto} = \text{área total} * 10 + \text{número de cômodos} * 2$
- Belo Horizonte → $\text{Imposto} = \text{área total} * 7 + \text{número de quartos} * 4$
- Porto Alegre → $\text{Imposto} = \text{área total} * 7,5 + \text{área da garagem} * 2,5$
 - Se não houver garagem → $\text{Imposto} = \text{área total} * 8$
- Curitiba → $\text{Imposto} = \text{área total} * 5$
 - Se idade do imóvel > 50 anos, somar idade * 3
 - Se idade do imóvel < 20 anos, somar idade * 2
 - Se 20 anos > idade do imóvel > 50 anos, somar idade * 2,5

Fazer a solução em Java e criar o diagrama de pacotes da solução proposta usando o Strategy Design. Criar uma classe view para apresentar a aplicação do Strategy.