|  |  |
| --- | --- |
|  | **AARE**: **Paradigmas de Linguagens de Programação**  **Professor**: Dr. Ausberto S. Castro V.  **Orientação a Objetos** |

**Prova JAVA**

Nome Completo: Gabriel Viana de Almeida Data: 12 de June de 2023

1. **(1 pontos)** Responda detalhadamente as seguintes questões (**exemplifique**):
   1. Qual é a diferença entre uma classe e um objeto?

Classe é uma estrutura de abstração que representa os métodos e atributos de um objeto, já o objeto é uma entidade, uma representação ou um conceito da realidade. Por exemplo, temos uma classe **celular** com atributos de “marca” “modelo”, seus métodos seriam “ligar”, “desligar”. Podemos criar um objeto *Samsung*  que os atributos seriam: marca -> samsung, modelo -> A30. E ele poderia realizar as operações de ligar e desligar. Assim como qualquer outro objeto criado a partir da classe **celular.**

* 1. Explique como se define uma classe em Java

Uma classe em java segue padrões: Primeiro, defini seu acesso (**publico** - pode ser acessada por todos, **private** - apenas é acessado dentro da própria classe, **protected** -> permite acesso além da própria classe, como também das subclasses.). Após isso, é dito o nome da classe, em seguida são definidos seus atributos e seus tipos. Por fim, é definido os construtores da classe, para inicializá-la, e há as definições dos métodos.

* 1. Por que uma classe é uma abstração lógica?

Classe é uma abstração porque representa um molde de algum objeto com seus atributos e métodos. Ela representa seus comportamentos e características se generalizar.

* 1. Como os objetos são criados?

São criados por meio de instância da classe. Por exemplo, suponha a existência de uma classe gato que possui como atributos: nome e raça.. Ao instancia-lo deveremos passar os parâmetros **nome** e **raça**

* 1. O que é coleta de lixo e como ela funciona?

A coleta de lixo foi criada para gerenciar a memória automaticamente para objetos, sua função é liberar memória ocupada que não são mais utilizadas. A coleta de lixo facilita na criação do código, uma vez que não precisa se preocupar em mexer na memória manualmente.

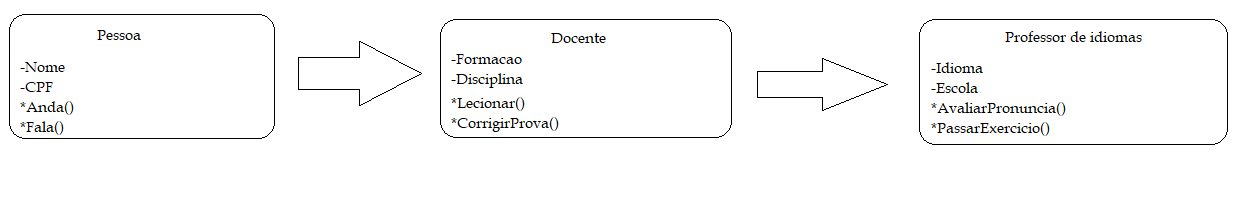
1. **(1 pontos)** Desenhar todas as classes, subclasses e superclasses associadas aos seguintes objetos, indicando em cada caso, pelo menos 2 atributos e 2 métodos, e a relação de herança entre classes

LEGENDA:

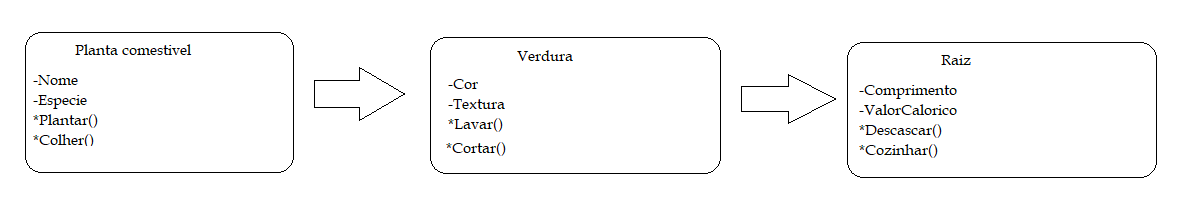
( - ) são atributos

( \* ) são métodos

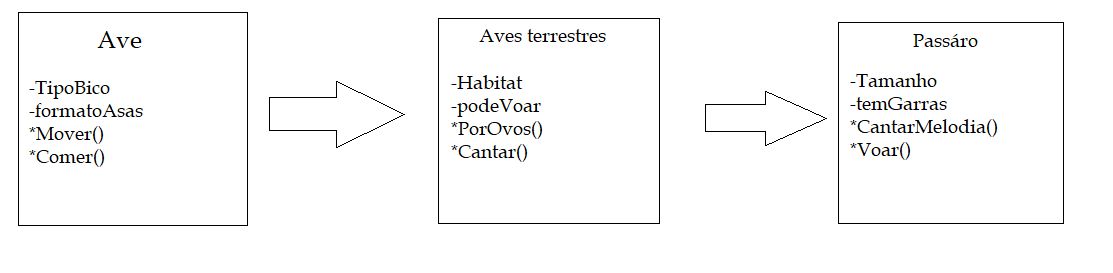
* 1. Docente



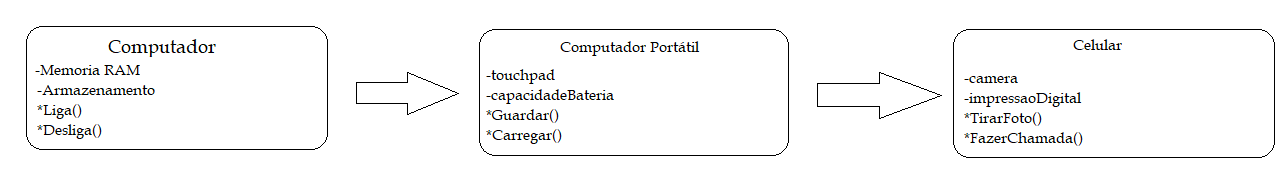
* 1. Verdura



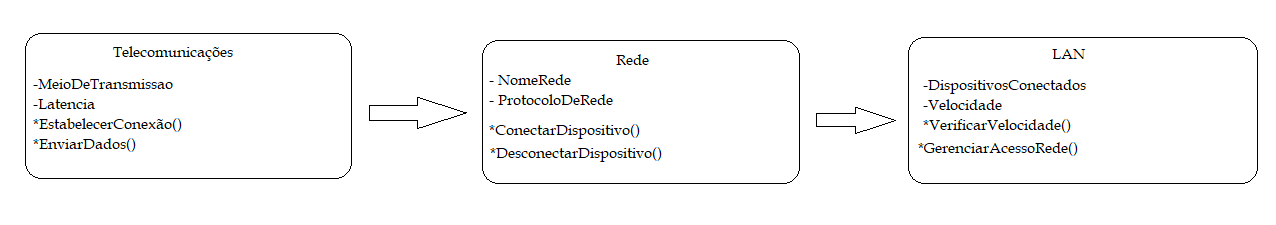
* 1. Ave



* 1. Computador

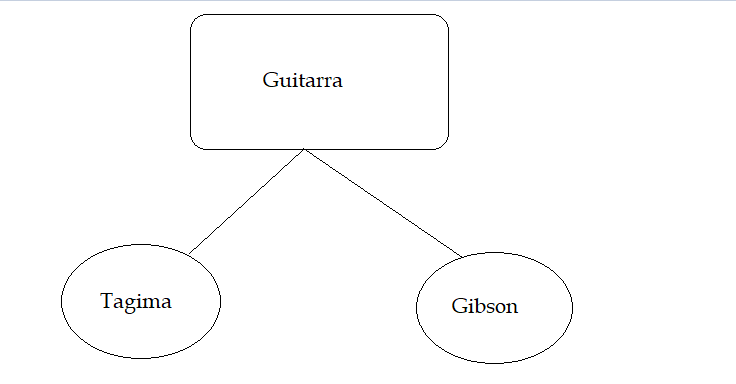
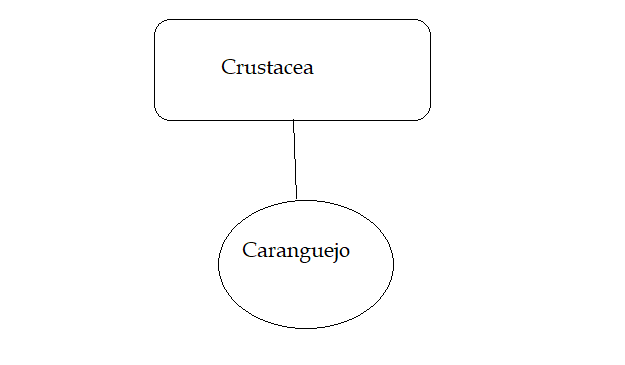
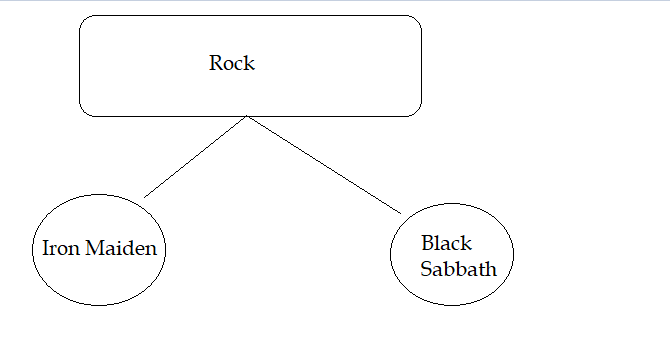
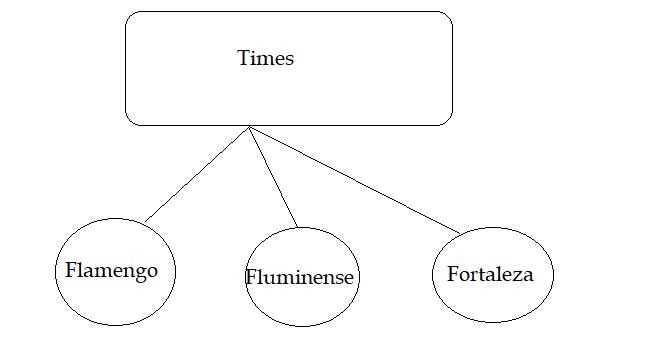
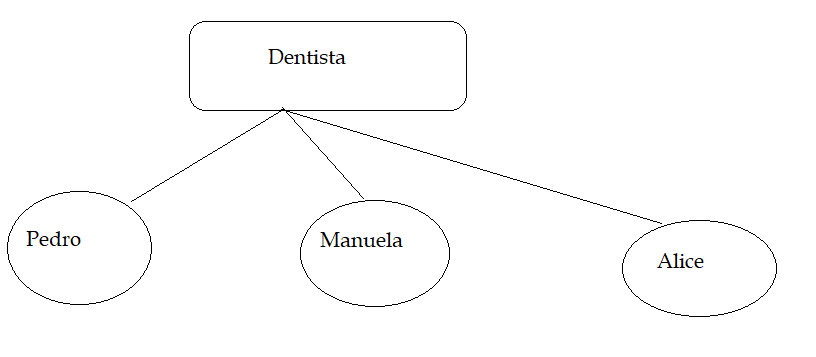


* 1. Rede

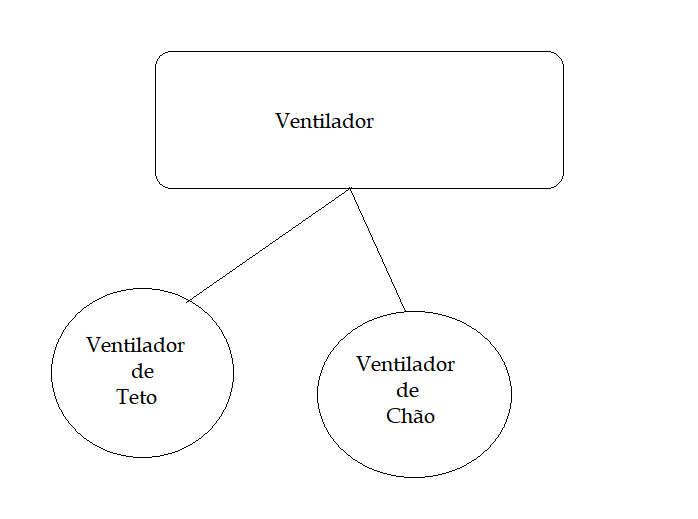
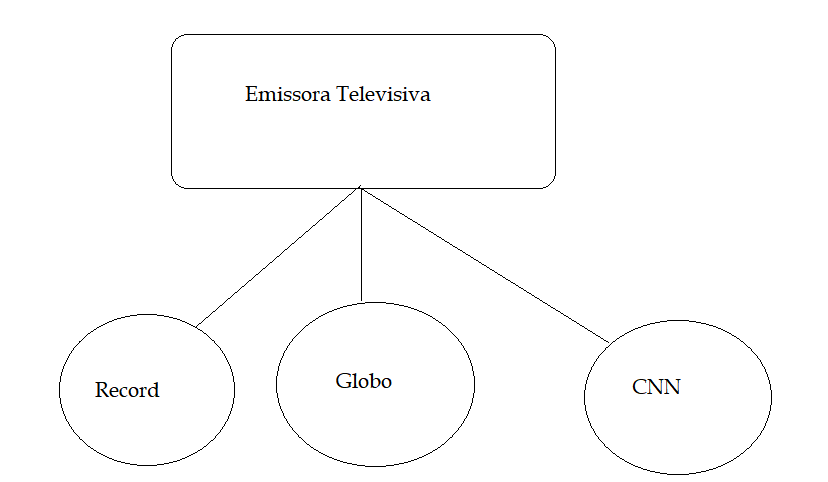
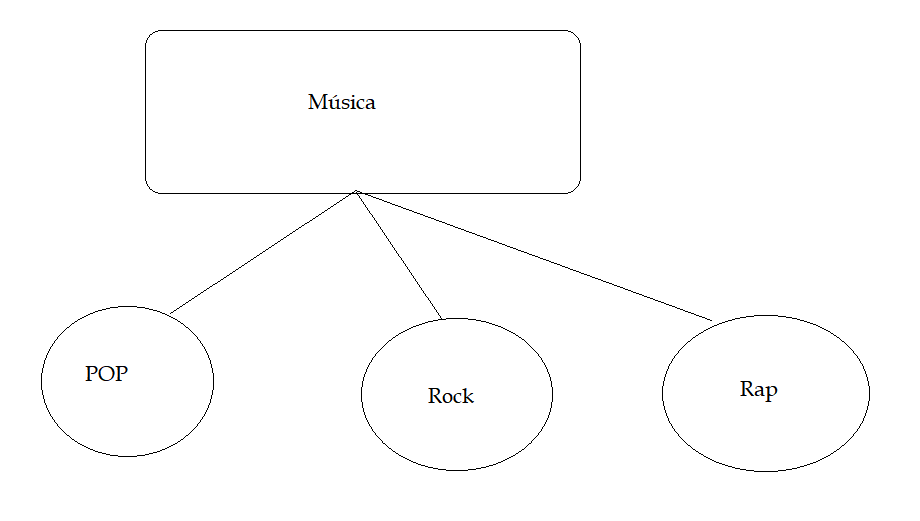
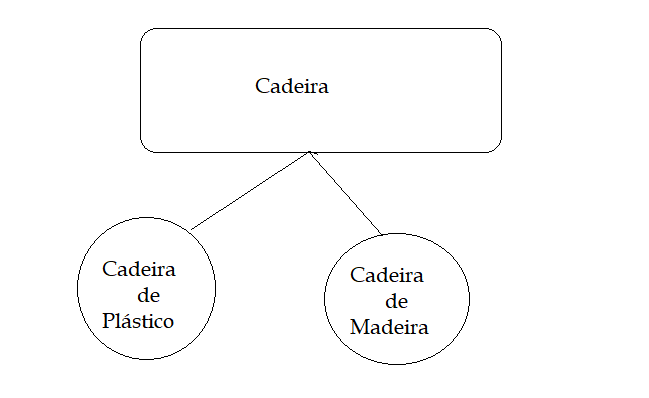
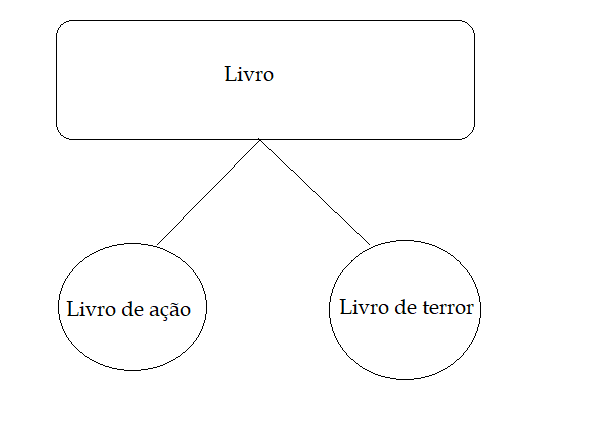


1. **(01 ponto)** Dar cinco exemplos de cada operação abstrata em orientação a objetos. Ilustre com imagens ou diagramas

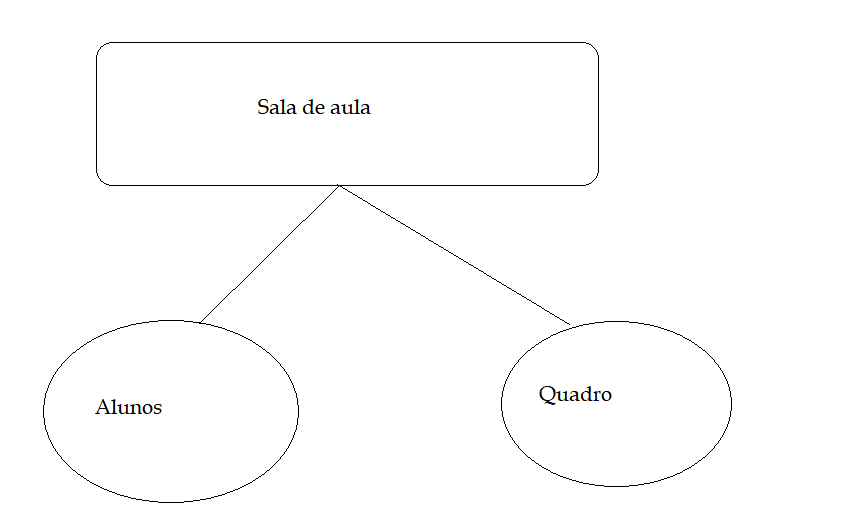
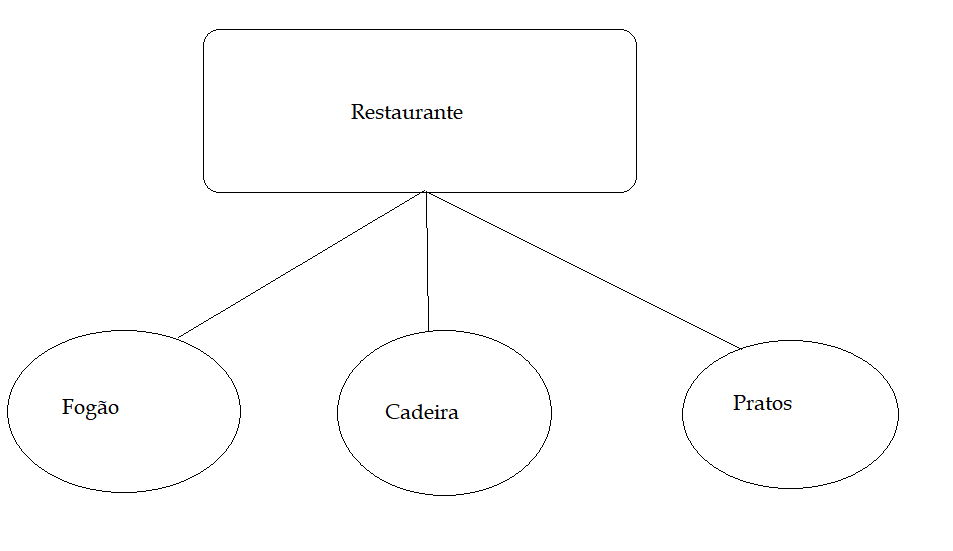
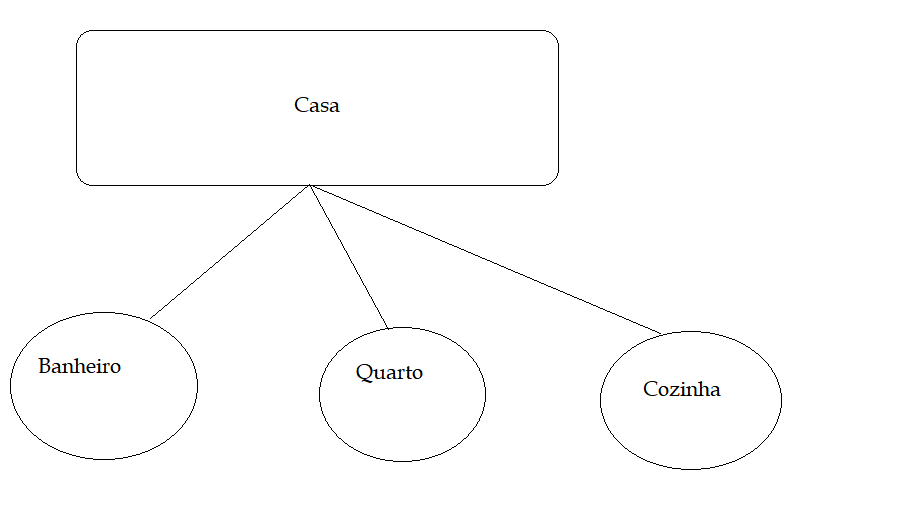
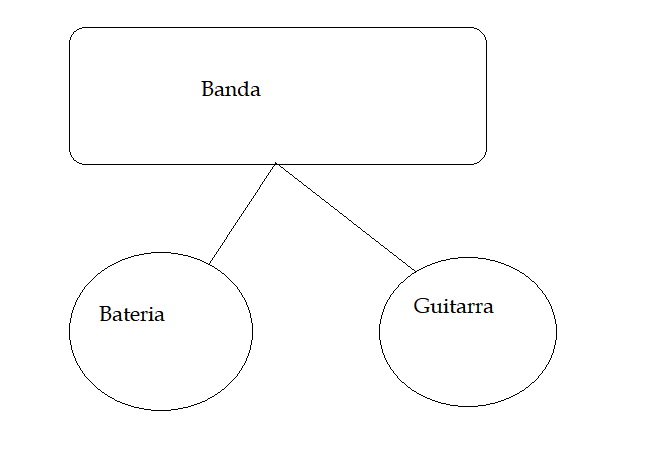
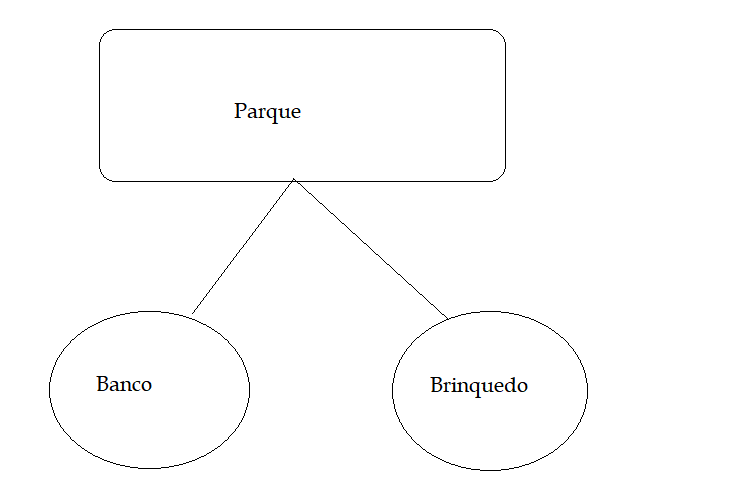
* Classificação



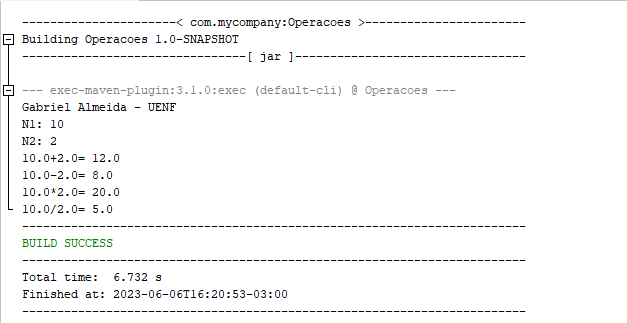
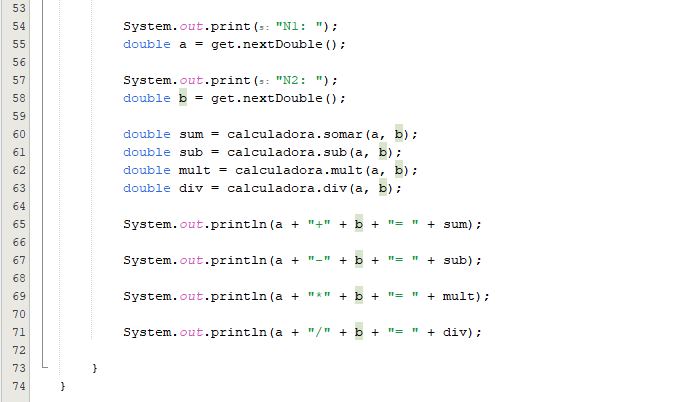
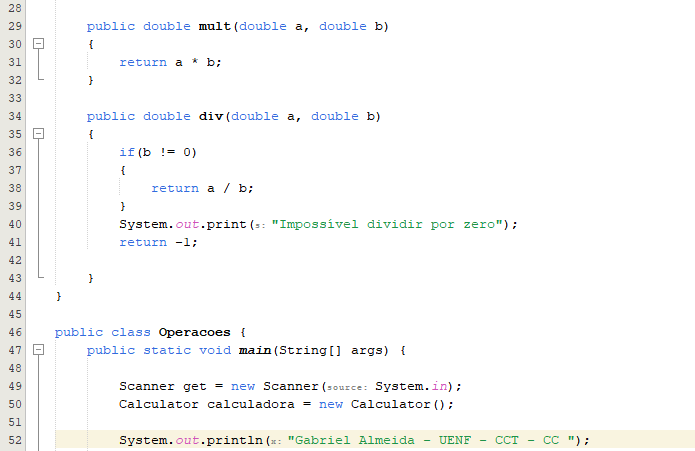
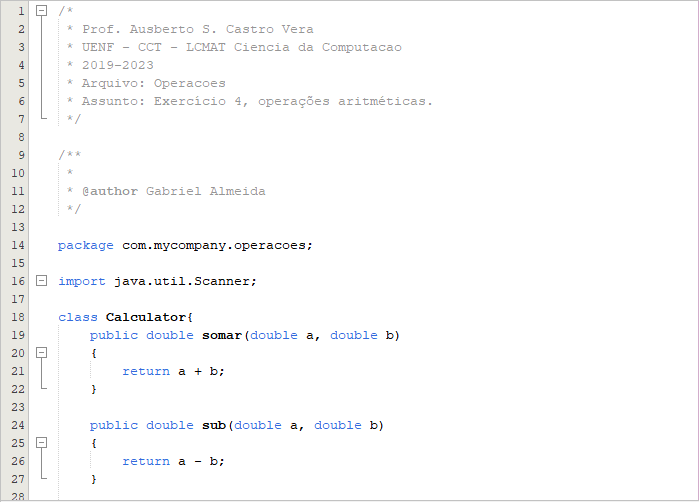
* Generalização



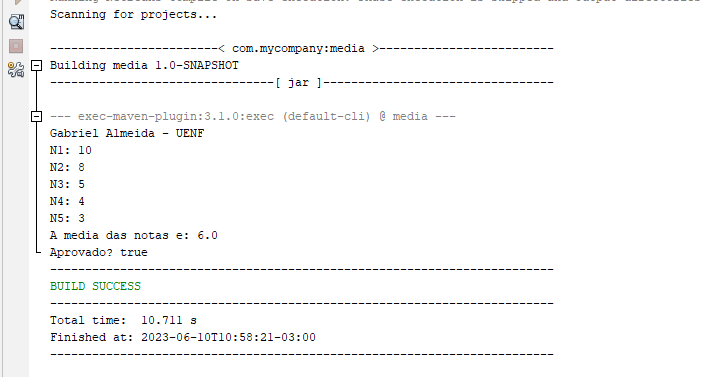
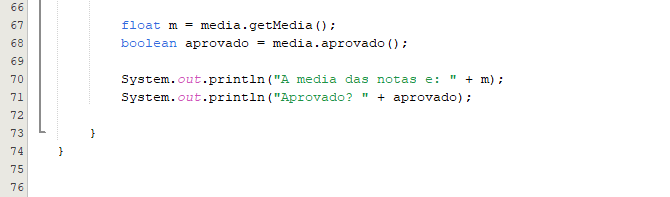
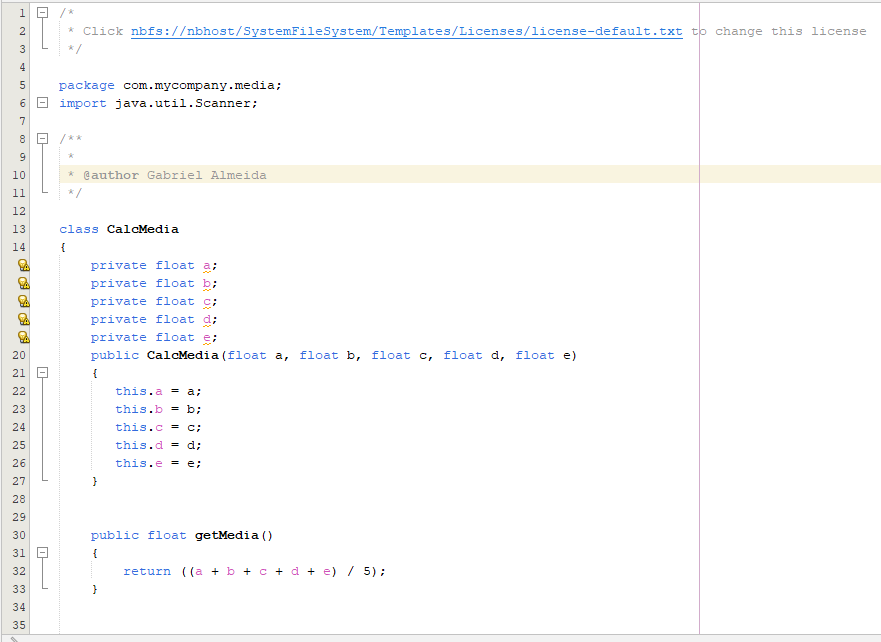
* Agregação



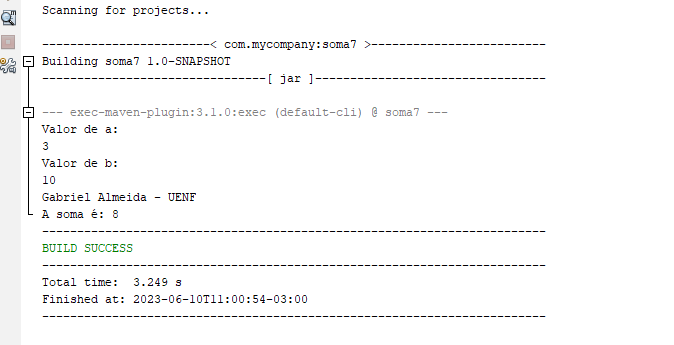
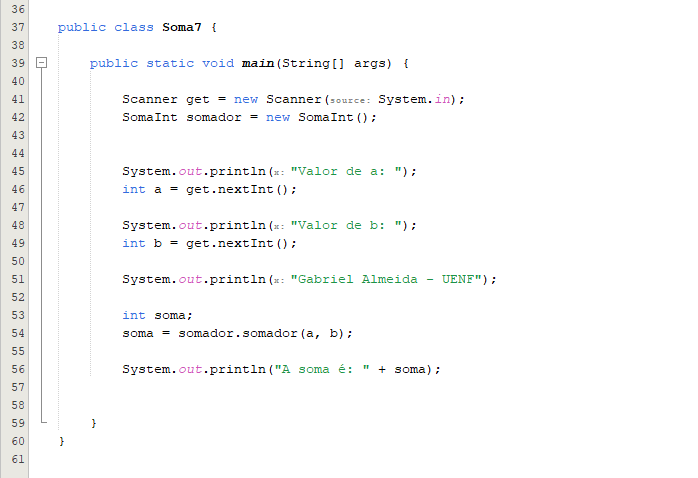
1. **(01 ponto)** Escreva um programa em Java que execute as quatro operações aritméticas de números reais



1. **(01 ponto)** Escreva um programa JAVA que calcule a média de cinco notas. O programa deve mostrar o valor da média e indicar se foi aprovado ou reprovado

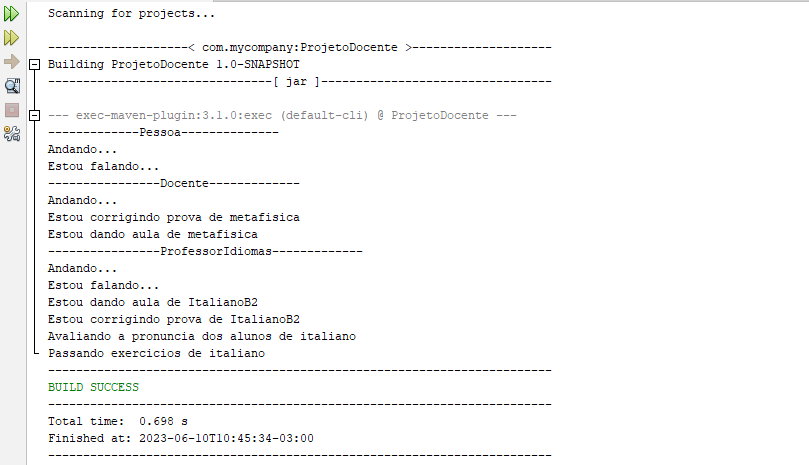
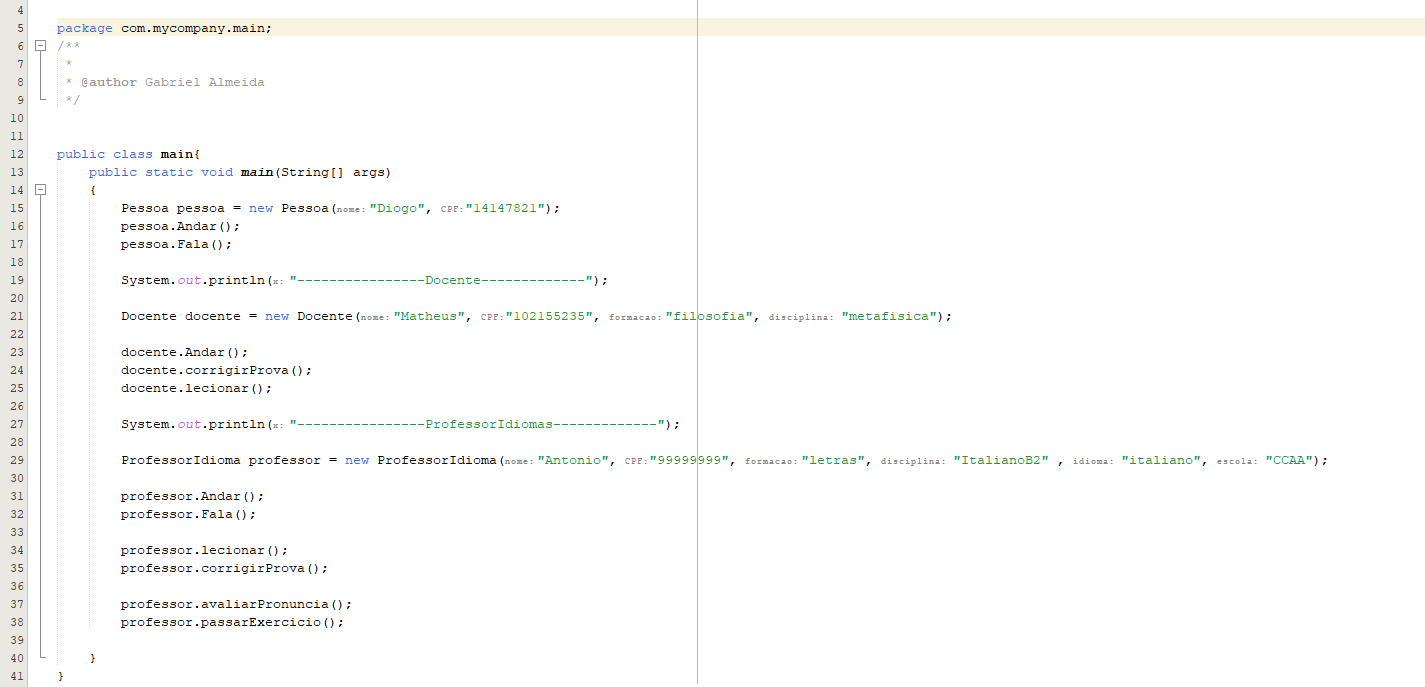
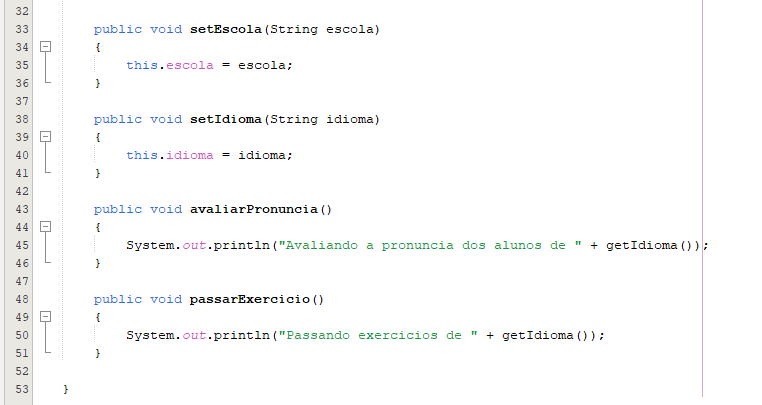
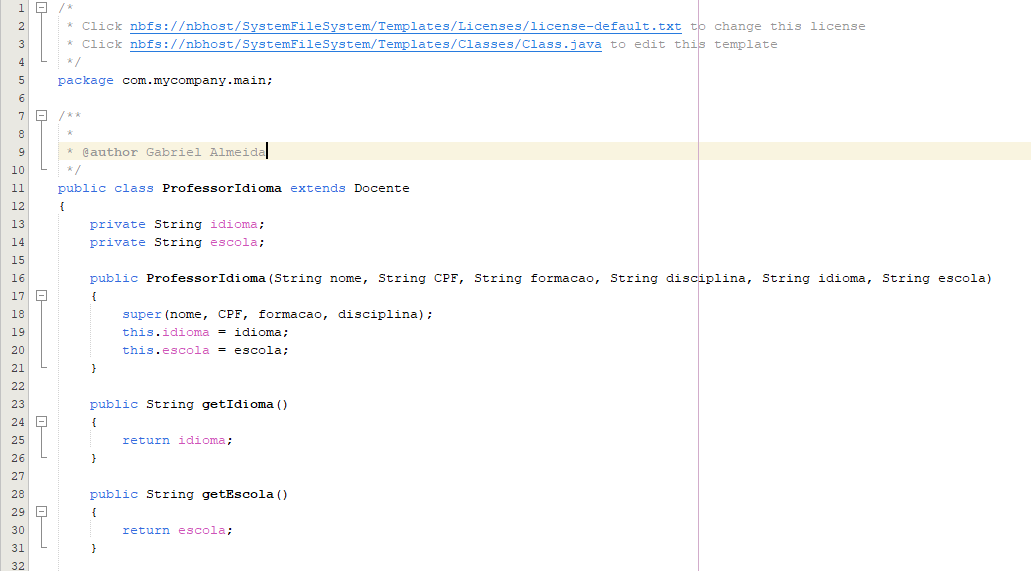
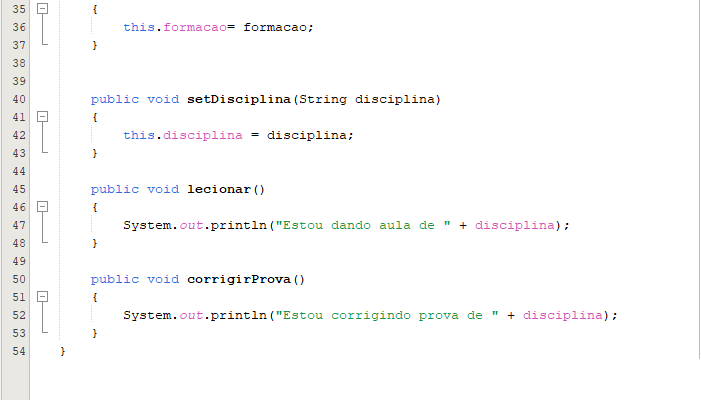
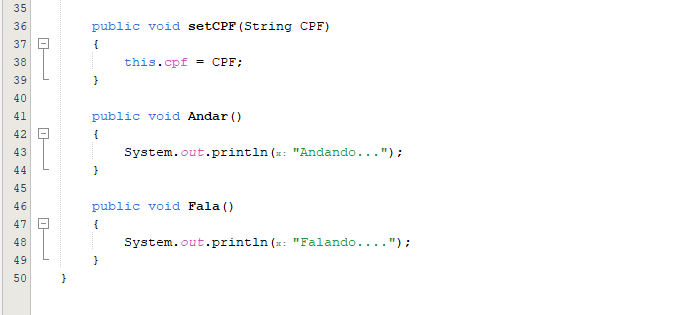


1. **(01 ponto)** Escreva um programa JAVA que calcule a soma dos inteiros maiores o iguales a um número inteiro A e menores ou iguais a um inteiro B.

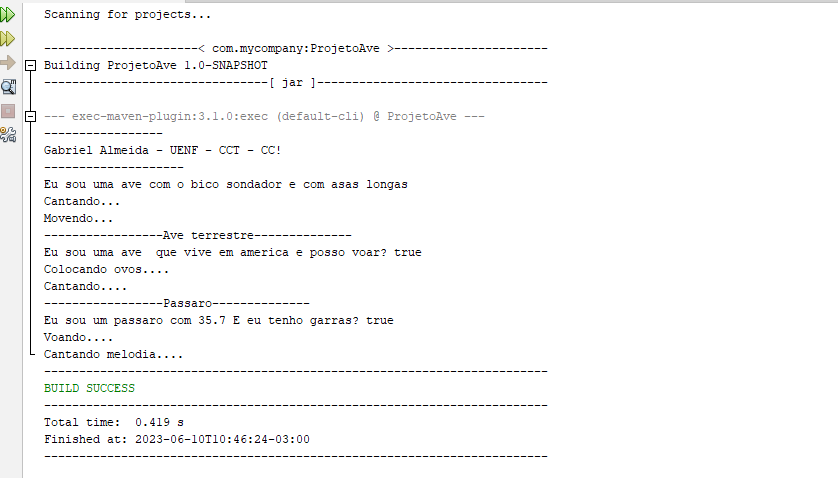
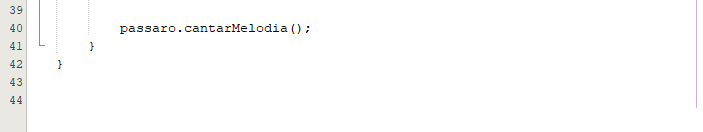
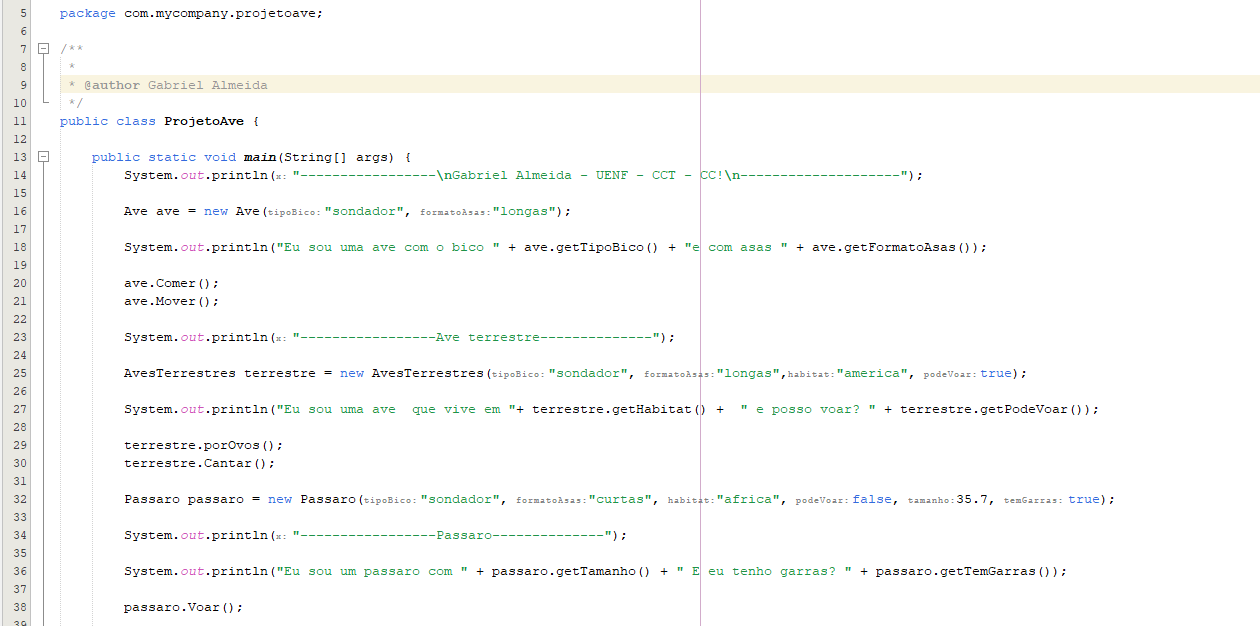
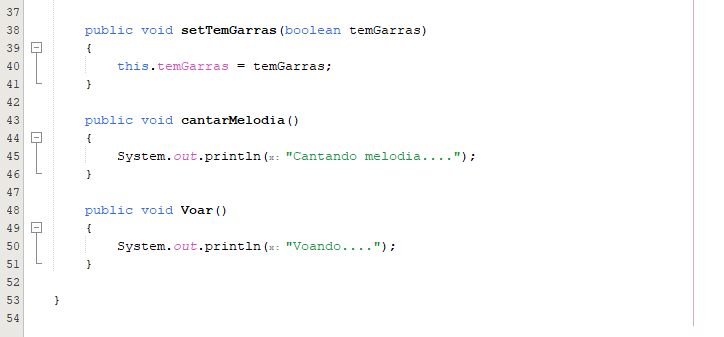
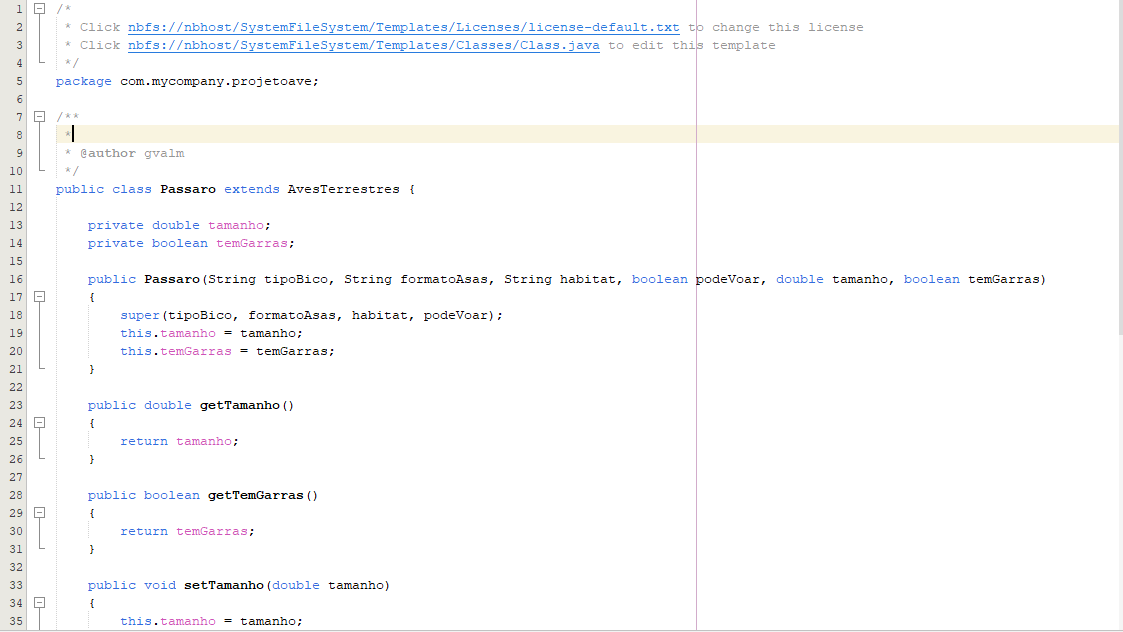
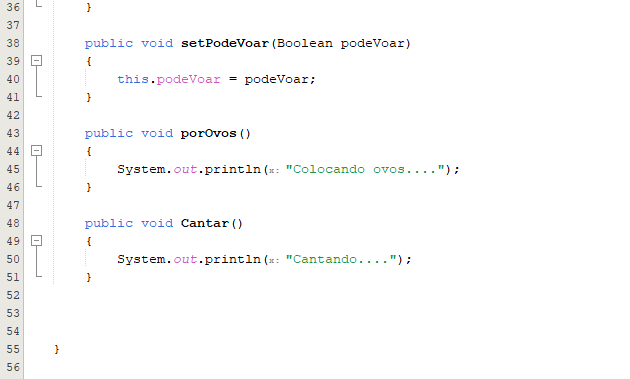
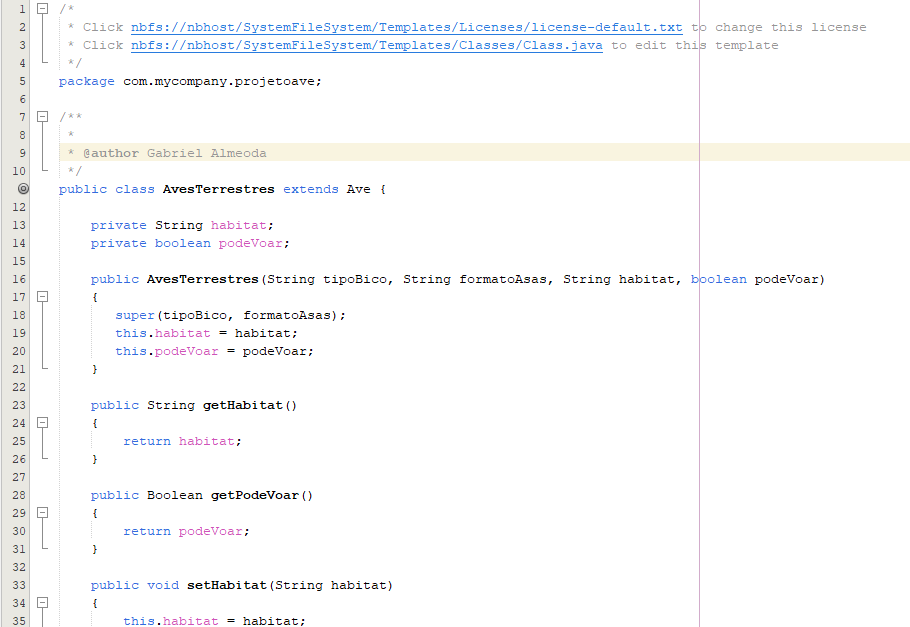
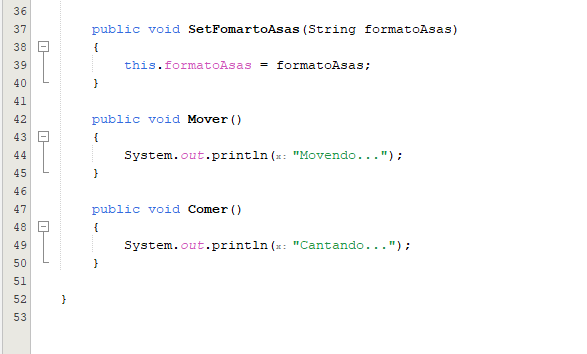


1. **(2 pontos)** Para cada caso da questão 2, escrever um programa JAVA, mostrando as classes, as operações (métodos), a criação de objetos e outras características da Programação Orientada a Objetos.

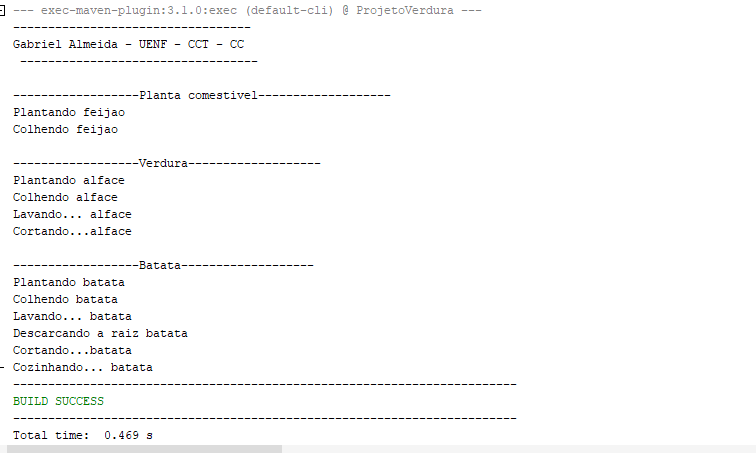
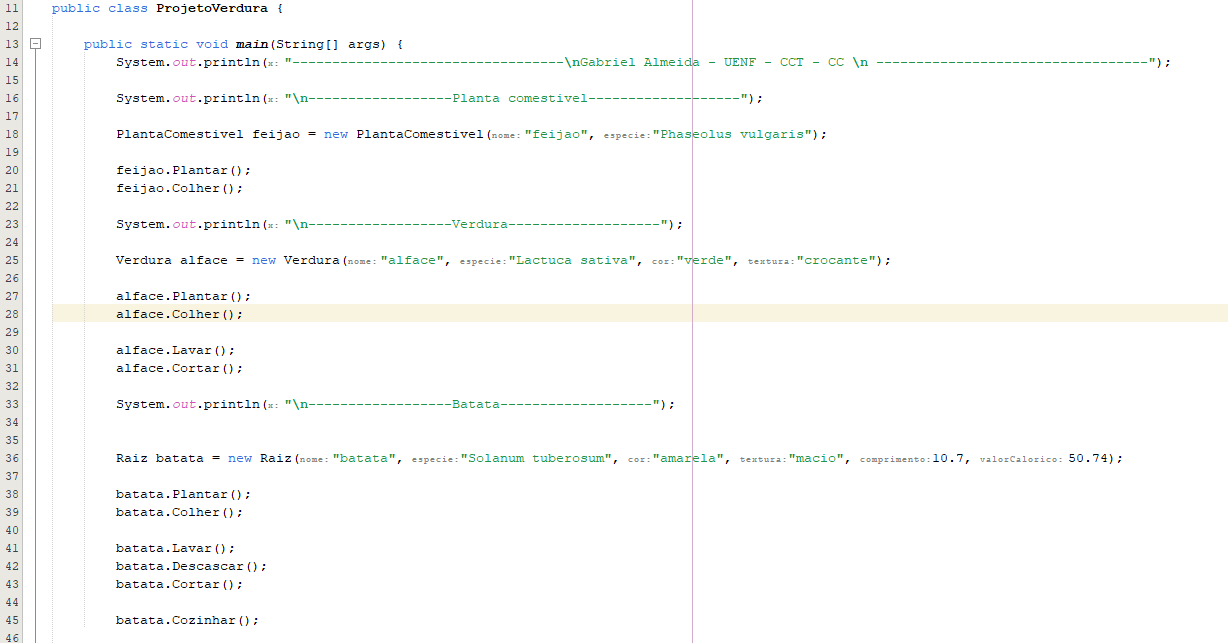
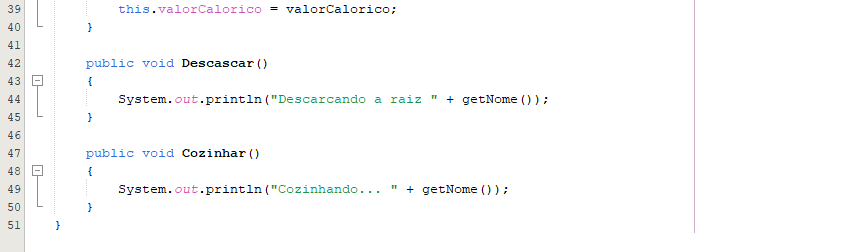
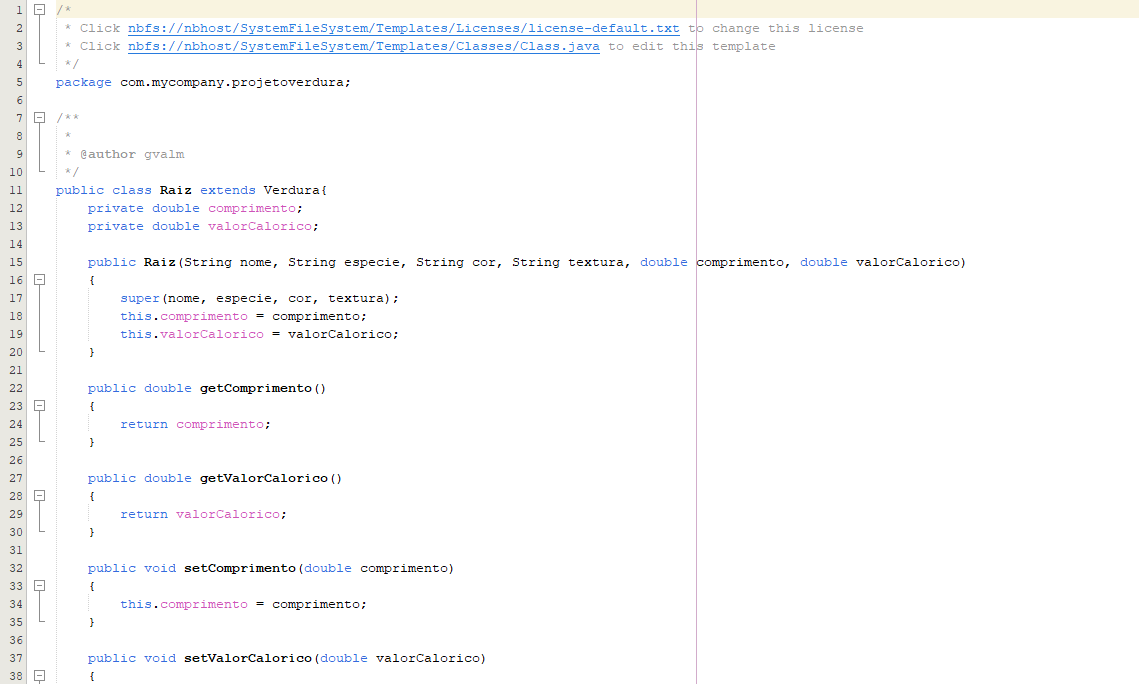
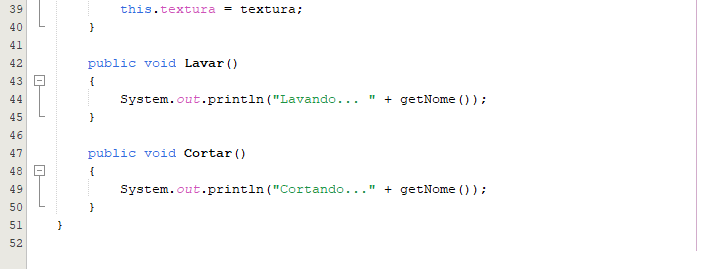
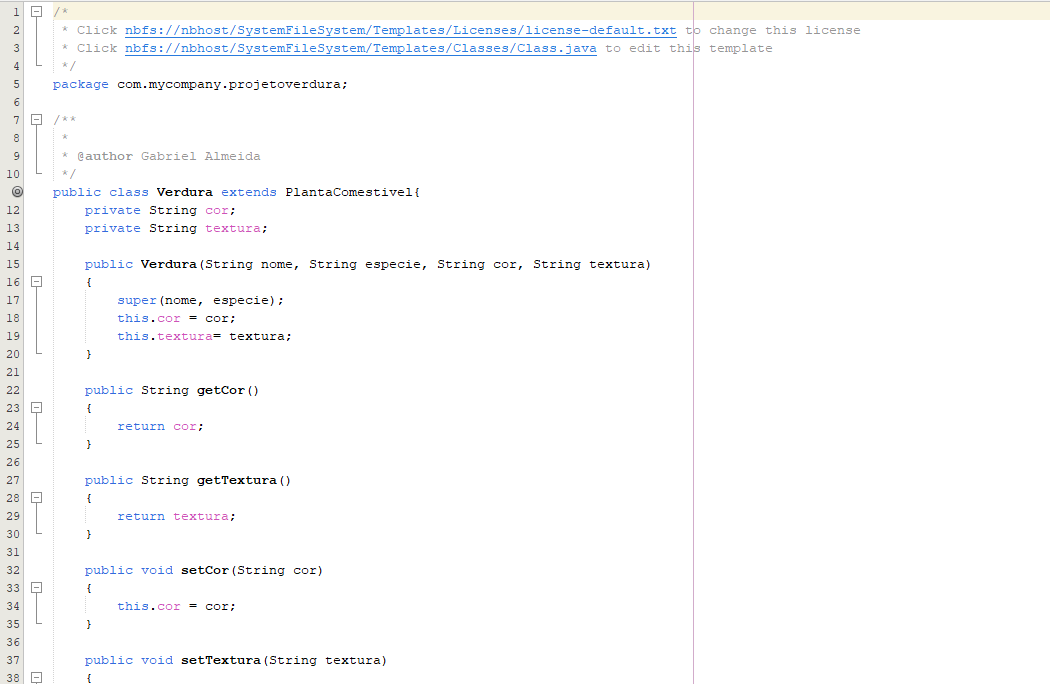
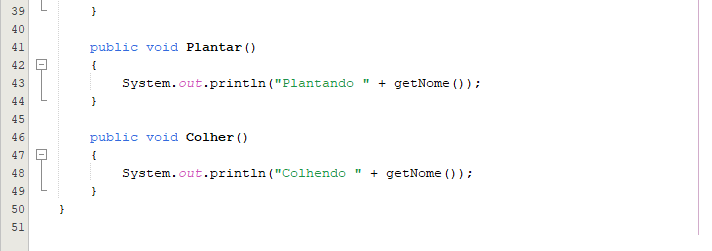
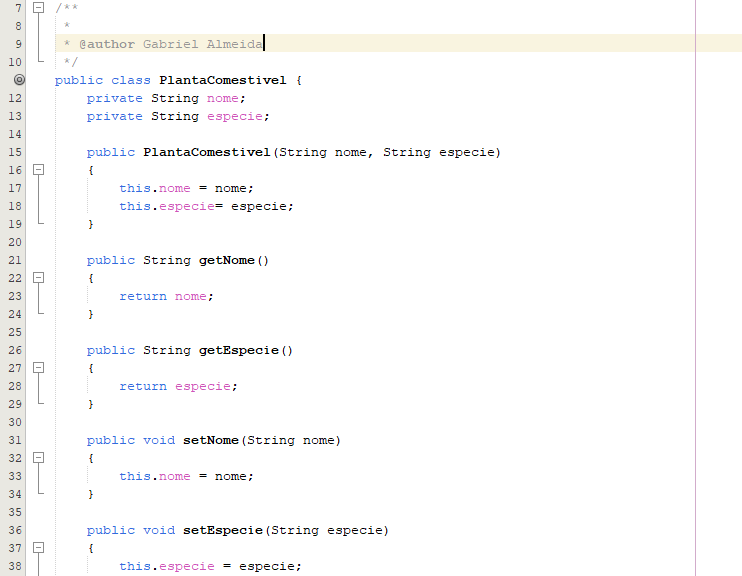
Pessoa:



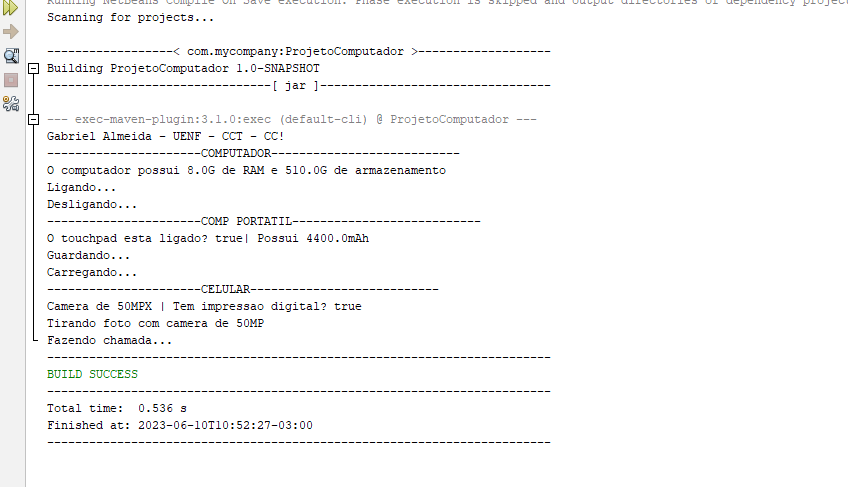
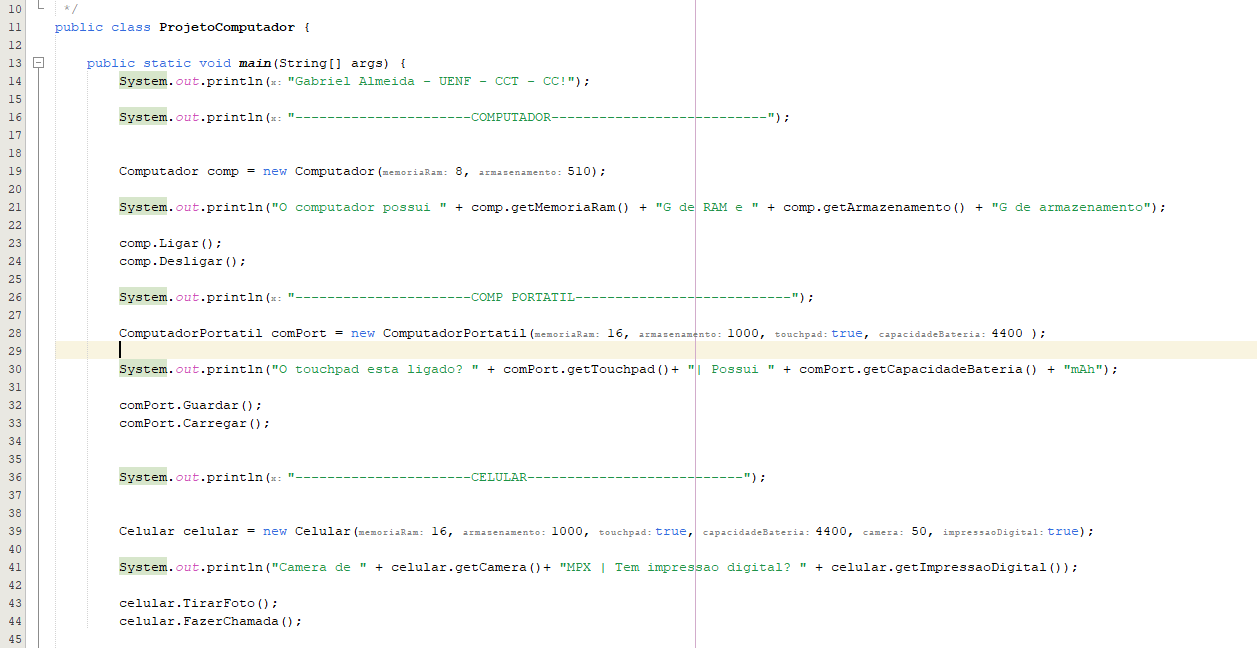
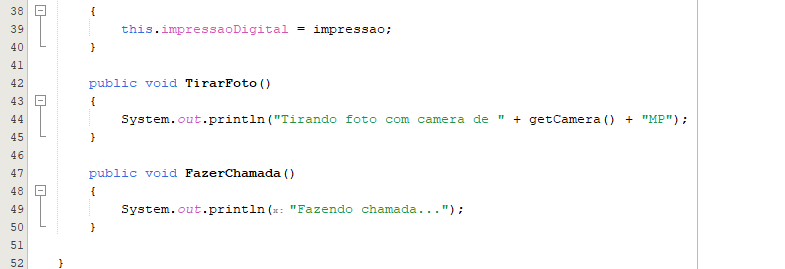
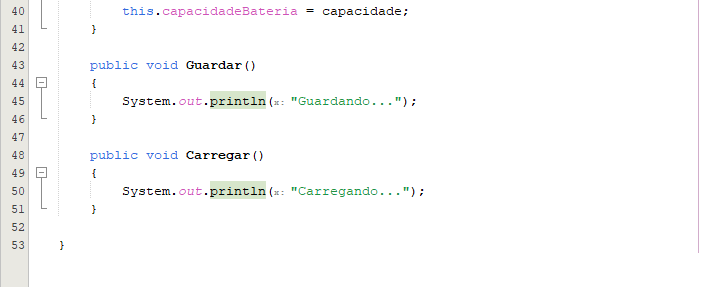
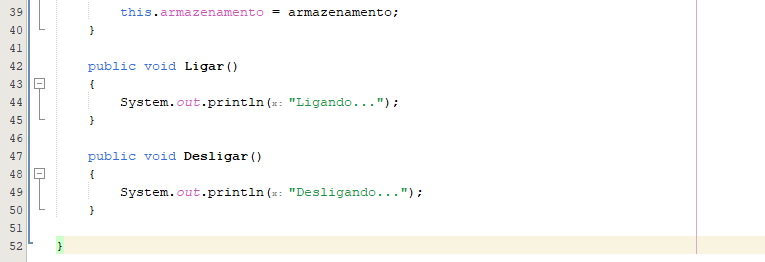
AVE:



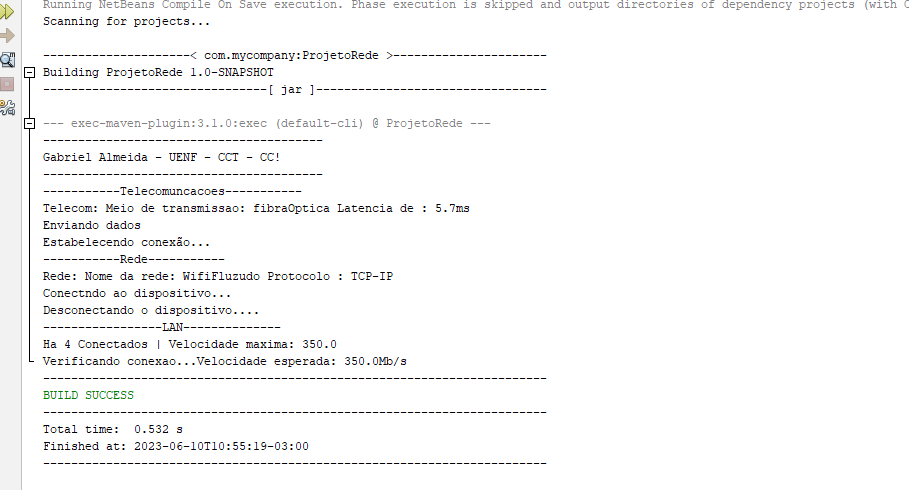
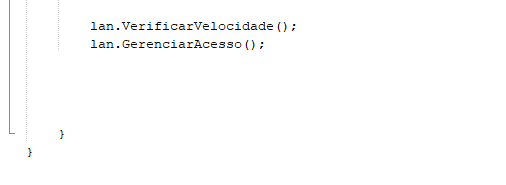
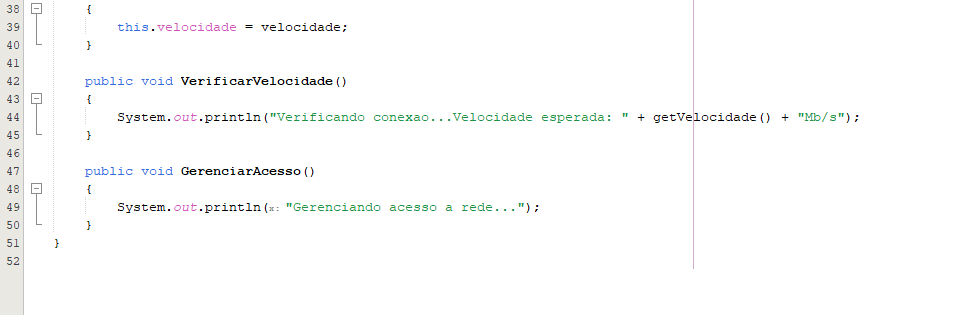
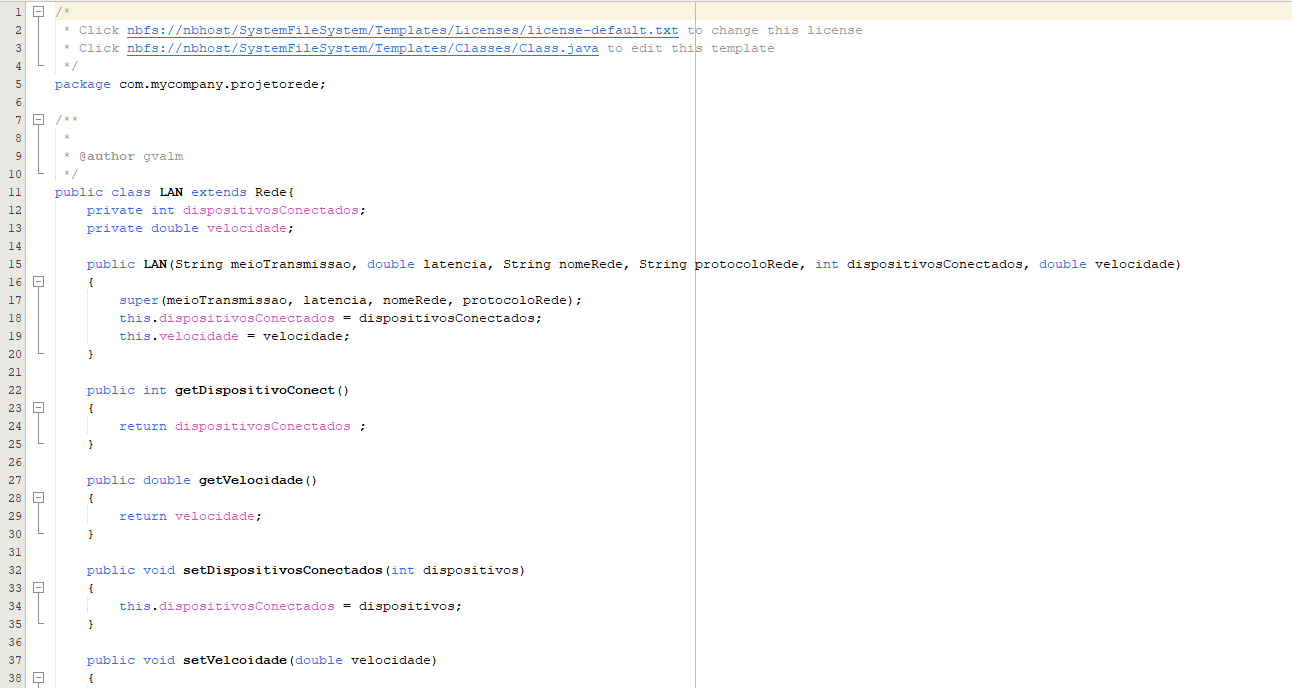
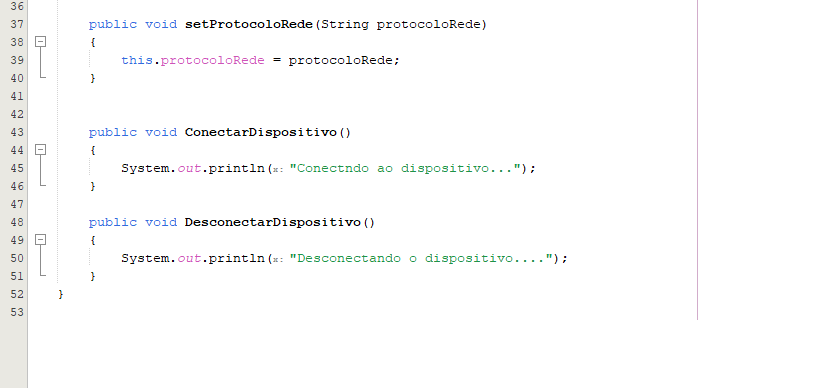
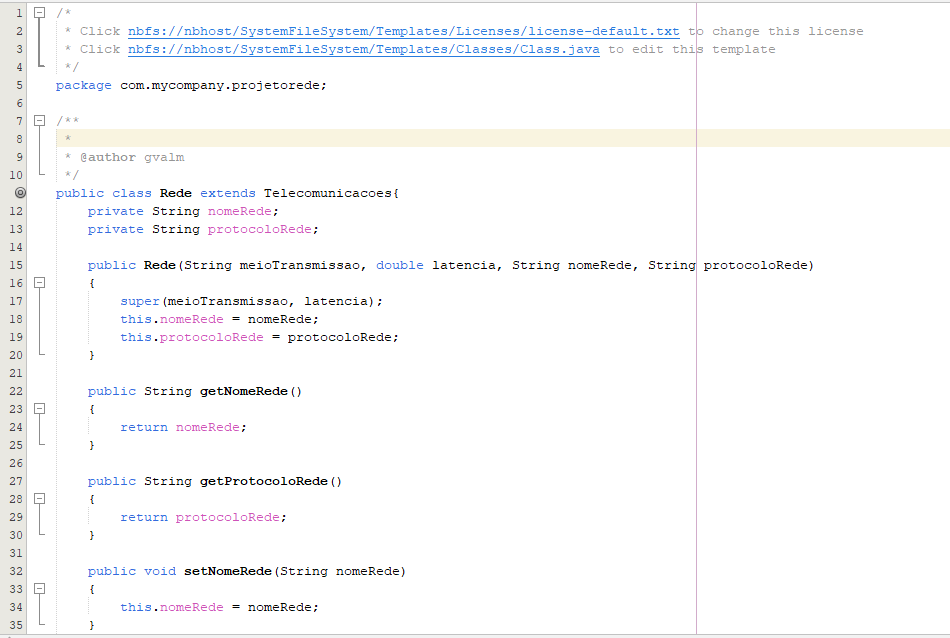
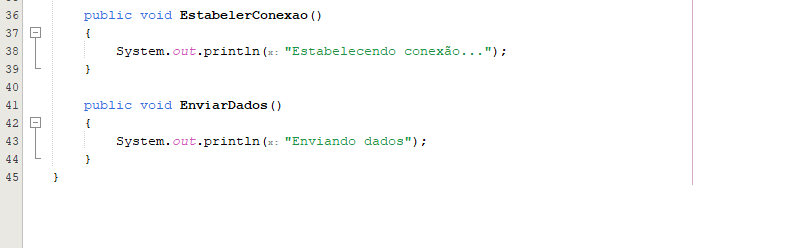
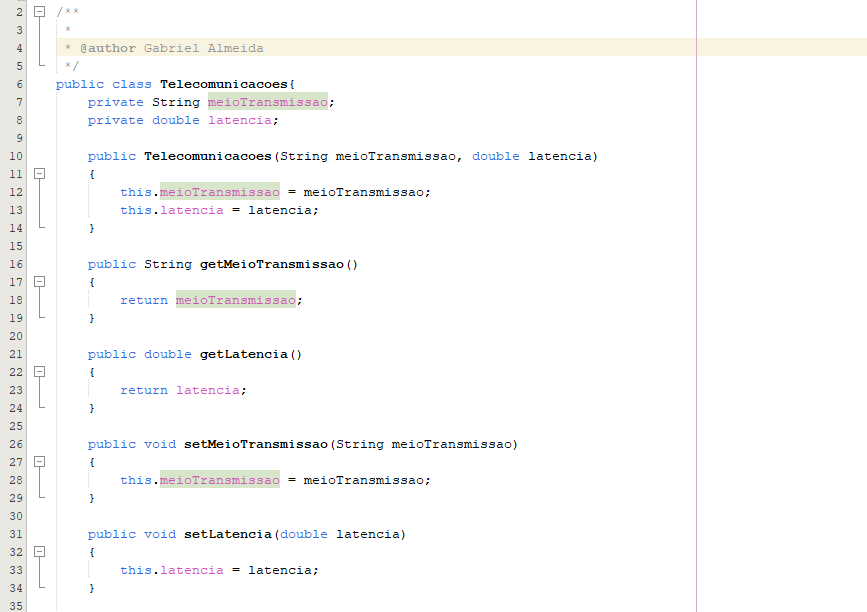
VERDURA



COMPUTADOR



REDE



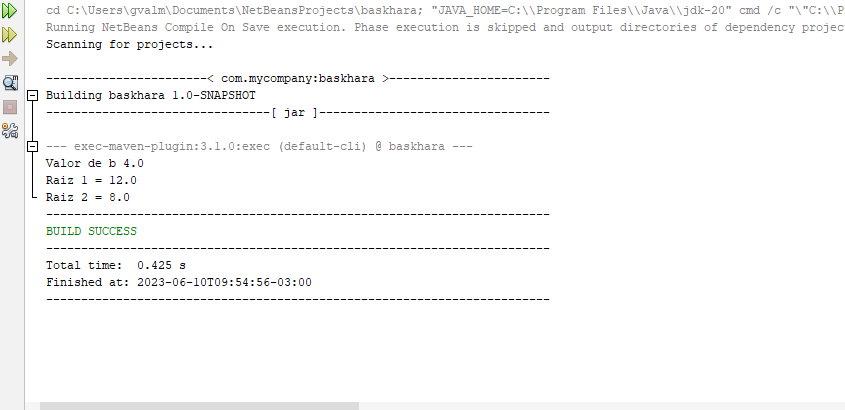
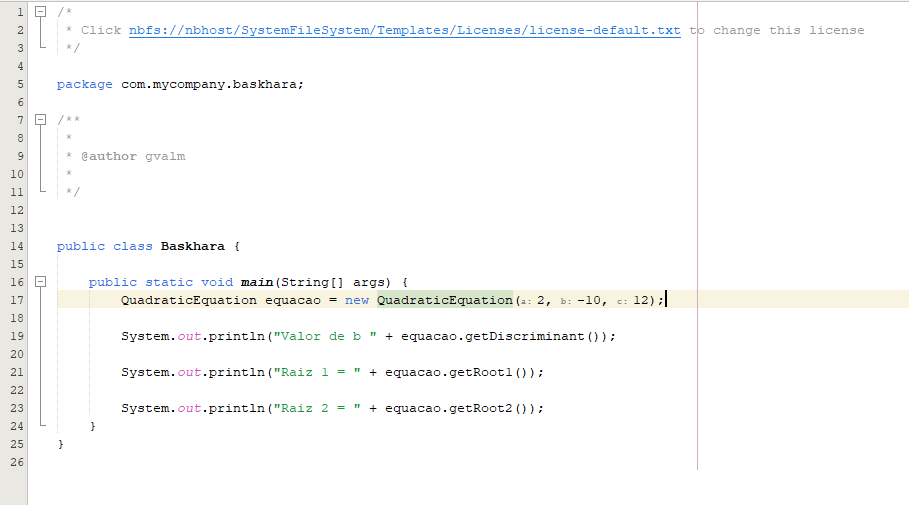
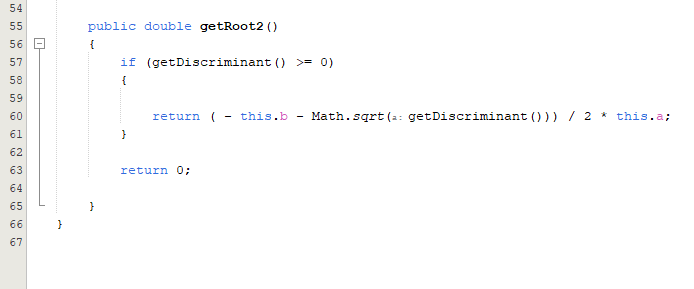
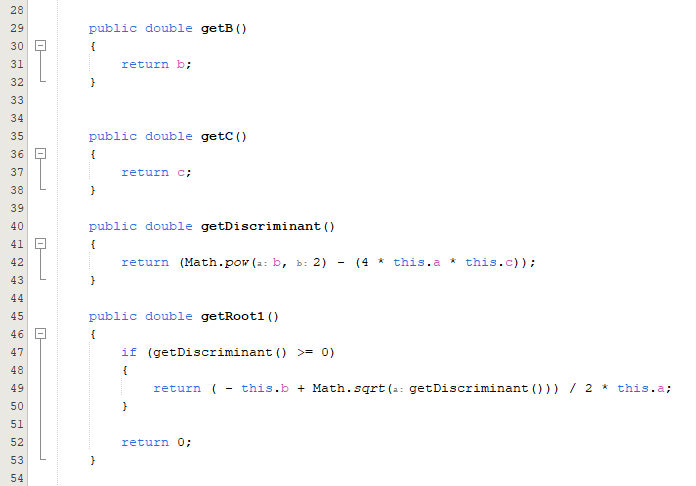
1. **(02 pontos)** Resolver a Questão **9.10** e **9.11** (pág.363) do livro “Introduction to JAVA Programming”, Y. Daniel Liang, 2014. Não precisa desenhar os diagramas UML. “Implemente a classe” = “Faça um programa em Java”

**O professor**



9.11

Baskhara:



Equação linear:

