Suporte para GitHub:

Carregando arquivos pelo navegador: https://drive.google.com/open?id=1Klf0HCJcB_4O5B7efMXrs_YYDXGwH89U
Tutorial Basico GitHub com Eclipse e EGit Usando Chave SSH: http://www.youtube.com/watch?v=fFBSazTSGZw
Usando Github com Github Desktop em Projetos Eclipse: http://www.youtube.com/watch?v=EqHIjlYyS4U
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

Vídeo Suporte:
Gerar Bibliotecas Java: https://youtu.be/9x3_c_0i6O0
Fila Dinâmica: https://youtu.be/XJ5NhHRhWss

Gerando link compartilháveis:
https://drive.google.com/file/d/1cyoxa5W67MY5xDM6gCYpklel1GU3QCKa

Para todos os exercícios, quando solicitado teste de mesa, carregar a solução para um drive compartilhado e quando solicitado desenvolvimento, definir o que se pede e aplicar o código em Java e carregar a solução no Github.

1. Considere o vetor a seguir:

0	5	7	-4	3	5	-2	-1	10	4	3	-6	2	-9	1	-5

Faça o teste de mesa do algoritmo:

```
Fila f = new Fila();
Para (int valor : vetor) {
	Se (valor >= 0) {
		f.insert(valor + 100);
	} senao {
		f.insert(valor * 2);
	}
}
Enquanto (!f.isEmpty()) {
	int num = f.remove();
	Se (num >= 0) {
		escreva(num * 3);
	} senao {
		escreva(num);
	}
	escreva(f.size());
}
```

2. Criar um projeto Java que implementa com a biblioteca Fila Genérica para simular um identificador de chamadas telefônicas enquanto o aparelho está fora da rede ou desligado. A aplicação deve ter uma classe no package controller, TelefoneController que tem os seguintes métodos:

- insereLigacao(Fila f, String numeroTelefone): void, que insere números de telefone em uma fila iniciada
- consultaLigacoes(Fila f):void, que desenfilera cada ligação realizada e exibe no console. Exibir uma exceção caso não haja ligação

Deve ter também uma classe Principal no package view com operações usando JOptionPane que permita ao usuário inserir números e consultar as chamadas perdidas. A aplicação deve rodar até uma opção de saída ser selecionada pelo usuário

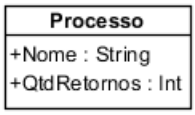
3. Considerando a biblioteca Fila Genérica, já criada, faça:

Criar um projeto Java (EscalonamentoProcessos) e importe a biblioteca FilaObject. Esse novo projeto irá receber Objectos com dados de processos com o formato do objecto abaixo e simular o escalonamento de processos RoundRobin.
A classe EscalonadorController no package controller deve ter 1 método:

- O método escalonador que recebe uma Fila de processos, como parâmetros, verifica se o número que vezes é maior que 1, decrementa esse valor no objeto e, se for maior que 1, insere o objeto novamente na fila;

A classe Principal, no package view, no seu método Main, deve inicializar uma fila Com processos, como no exemplo abaixo

Objeto:



Exemplo de entrada:
{"notepad.exe;14","write.exe;35","chrome.exe;27","acrobat.exe;52","firefox.exe;18","word.exe;25"}

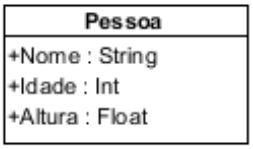
4. Criar um projeto Java que implementa com a biblioteca Fila Genérica para simular um ambiente corporativo, comum, como agências bancárias que tem diversos computadores e 1 impressora central. Nesses casos, a impressora tem uma fila de impressões para que cada documento enviado comece e termine a impressão sem que documentos se misturem.
A aplicação deve ter uma classe no package controller, ImpressoraController que tem os seguintes métodos:

- insereDocumento(Fila f, String documento): void, o documento no formato ID_PC;Nome_Arquivo (Já validado antes do envio) deverá enfileirar os documentos enviados
- imprime(Fila f):void, que desenfilera um documento da fila, por vez, exibe no console [#PC: ID_PC – Arquivo: Nome_Arquivo]. Cada impressão dura de 1 a 2 segundos usar Math.random() ou a classe Random e um Thread.sleep*(tempo) para simular o tempo de impressão. Exibir uma exceção caso não haja documento na fila de impressão.

Deve ter também uma classe Principal no package view com operações usando JOptionPane que permita ao usuário inserir e validar os documentos de entrada e iniciar um procedimento de impressão. A aplicação deve rodar até uma opção de saída ser selecionada pelo usuário
*** Como a classe de ImpressoraController não é uma Thread, para forçar um sleep, deve-se usar uma chamada estática da classe Thread (Thread.sleep(tempo))**

5. No parque de diversões, forma-se uma fila de crianças para ir ao brinquedo denominado Boomerang. Para entrar no brinquedo, a criança deve ter mais de 1.60m e mais de 16 anos. Criar uma aplicação Java, baseada na biblioteca Fila Genérica, que faça:

1) Criar um objeto model com os atributos:



2) Criar uma classe view (Principal.java) que, na main:
    a. Inicialize uma nova fila
    b. Insira 30 clientes com suas características, sendo:
        i. Nome (Pessoa1, Pessoa2, Pessoa3, ...)
        ii. Idade (Número entre 10 e 40)
        iii. Altura (Número entre 1.35 e 2.00)
3) Criar uma classe controller chamada ParqueController, que tenha um método brinquedo(Fila fila): void, que receba a fila populada, como parâmetro, e faça, para cada Cliente:
    a. Verificar se a pessoa está em condições de ir ao brinquedo
    b. Exibir o nome do cliente, se passou para dentro do brinquedo ou não e a razão
4) O método main da classe Principal.java deve chamar o método brinquedo(Fila fila): void.

O método brinquedo(Fila fila): void deve estar baseado nas operações da fila (insert(), remove(), list(), size(), isEmpty())