

DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

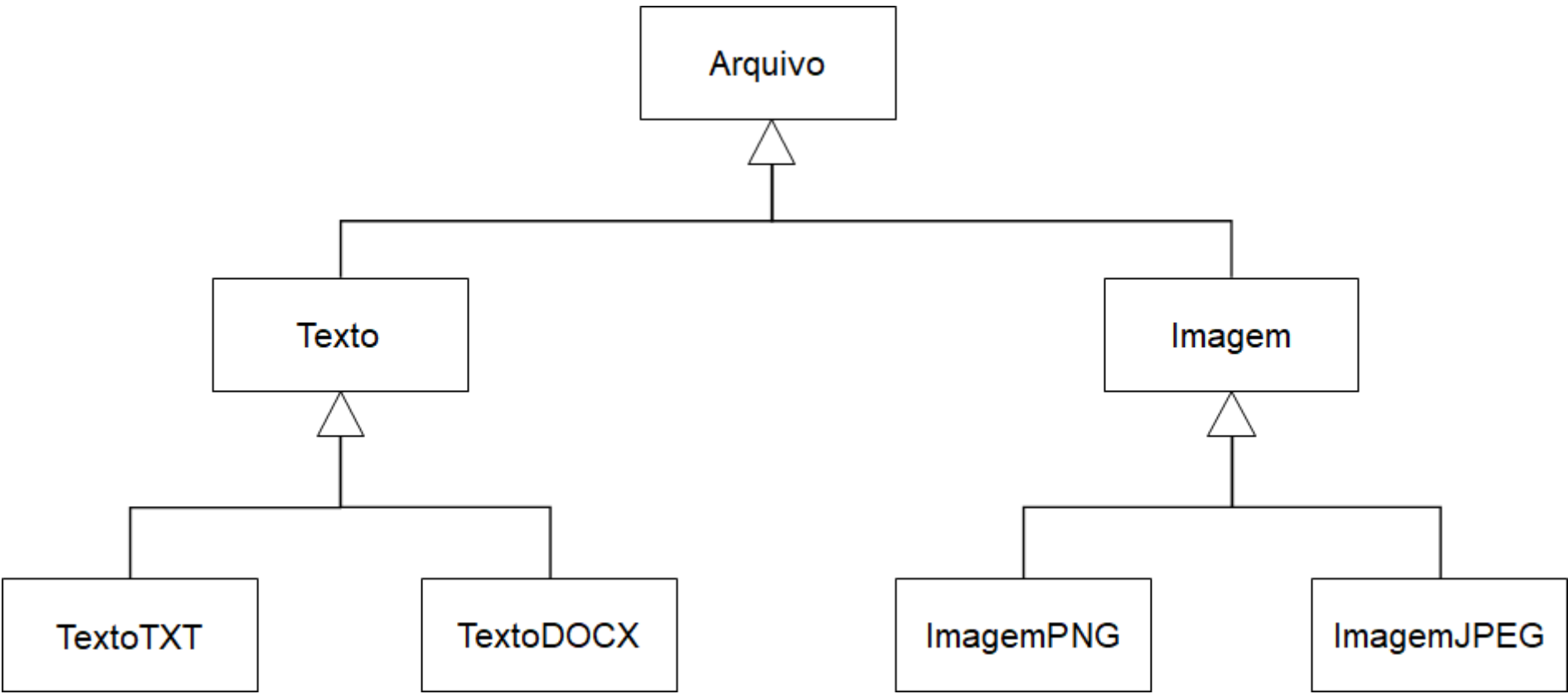
[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Generics](#) / [\[EP\] Pasta compactada](#)

Descrição [Visualizar envios](#)

[EP] Pasta compactada

Data de entrega: segunda, 2 Ago 2021, 23:59
Arquivos requeridos: CompactaArquivos.java ([Baixar](#))
Tipo de trabalho: Trabalho individual

Considere a seguinte hierarquia de classes de arquivos (essas classes já estão implementadas no sistema de correção automática dentro do pacote arquivos). Essas classes podem ser instanciadas com o construtor sem parâmetros.



Descrição das classes no diagrama:

- Classe Arquivo;
- Classes Texto e Imagem, que são subclasses de Arquivo;
- Classes TextoTXT e TextoDOCX, que são subclasses de Texto;
- Classes ImagemPNG e ImagemJPEG, que são subclasses de Imagem.

Escreva um programa (pacote arquivos) para guardar arquivos de texto e de imagem em pastas compactadas. O programa deverá guardar duas pastas compactadas, uma para guardar apenas arquivos de texto e outra para guardar apenas arquivos de imagem.

Uma pasta compactada é representada por uma instância da classe PastaCompactada<T>. A classe PastaCompactada<T> já está implementada e possui um parâmetro de tipo T. Portanto, para instanciar uma pasta compactada que armazenará apenas arquivos de Texto:

```
PastaCompactada<Texto> pastaTextos = new PastaCompactada<Texto>();
```

Os métodos da classe são:

- **adicionar**(arquivo: T) - adicionar um arquivo (tipo T) na pasta compactada.
- **imprimir**() - imprime os arquivos na pasta compactada.

O programa deve:

1. Instanciar duas pastas compactadas: uma para arquivos de texto e outra para arquivos de imagem;
2. Ler um número (quantidade de arquivos a serem adicionados);

3. Ler uma sequência de tipos de arquivos (cada tipo é uma String que corresponde ao nome da classe do arquivo). Ao ler um tipo, instancie a classe respectiva. Se o arquivo instanciado for uma imagem, ele deve ser adicionado na pasta de imagens e, se for um texto, na pasta de textos.
4. Após terminar de adicionar todos os arquivos, o programa deve chamar o método imprimir da pasta de textos e depois da pasta de imagens.

Importante: Submeta apenas o programa principal, que deverá estar na classe pública **CompactaArquivos** (o programa deve estar no **pacote arquivos**). O programa não pode realizar impressão de dados. A impressão de dados é permitida apenas com o método imprimir das pastas. Também não é permitido utilizar java.util ou import. Apenas o import java.util.Scanner é permitido.

Casos de teste

Formato dos casos de teste (que aparecem ao avaliar as classes no sistema de correção automática):

Entrada:

- quantidade de arquivos a serem lidos (int)
- lista de tipos de arquivos (sequência de String)
 - o tipo corresponde ao nome da classe que deve ser instanciada, conforme hierarquia de classes apresentada no enunciado

Saída:

- saída do método imprimir para a pasta compactada com arquivos de texto
- saída do método imprimir para a pasta compactada com arquivos de imagem

Arquivos requeridos

CompactaArquivos.java

```
1 package arquivos;
2
3 import java.util.Scanner;
4
5 public class CompactaArquivos {
6     public static void main(String[] args) {
7
8     }
9 }
```

[VPL](#)



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

- Conheça a UFABC
- Conheça o NTI
- Conheça o Netel

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos

