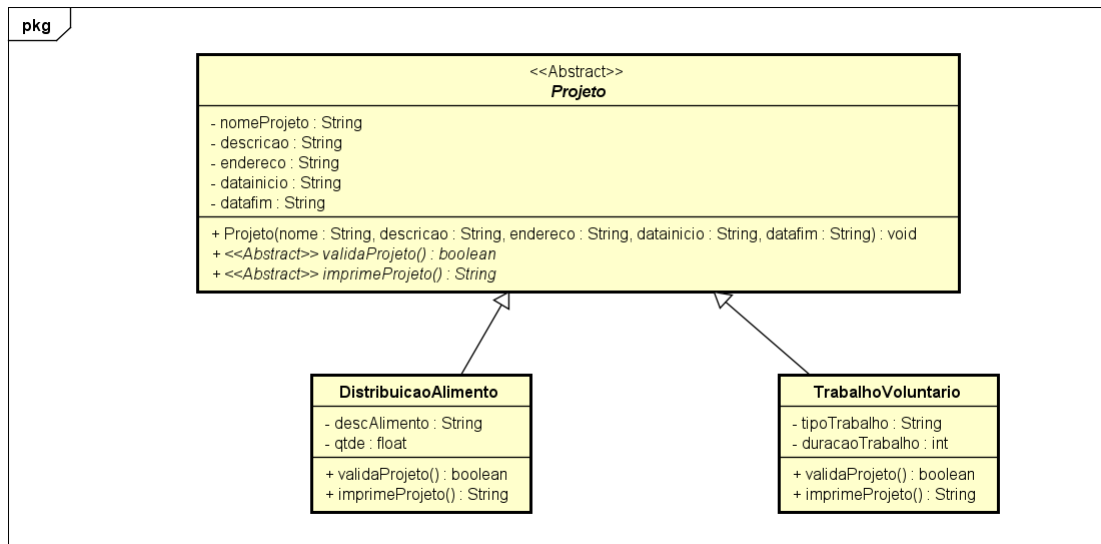


Prática 6

Exercício 1)



1. Implemente as classes em Java, conforme o diagrama. Crie os construtores para cada classe, mesmo que não esteja nos Diagramas, utilize o comando **super()**.
2. Crie os métodos *Setters* e *Getters* de todos os atributos.
3. Crie o método **validaProjeto** na classe **DistribuicaoAlimento**. Este método deverá retornar **true** se a **datafim** do Projeto estiver vazia, caso contrário retornará **false**.
4. Crie o método **imprimeProjeto** na classe **DistribuicaoAlimento**. Este método deverá retornar uma String com os seguintes atributos: *nomProjeto*, *descricao*, *datainicio*, *datafim*, *descAlimento* e *qtde*.
5. Crie o método **validaProjeto** na classe **TrabalhoVoluntario**. Este método deverá retornar **true** se a **duracao** for superior a 2, caso contrário retornará **false**.
6. Crie o método **imprimeProjeto** na classe **TrabalhoVoluntario**. Este método deverá retornar uma String com os seguintes atributos: *nomProjeto*, *descricao*, *datainicio*, *datafim*, *tipoTrabalho* e *duracaoTrabalho*.
7. Crie a classe **main()** que deve possuir um menu para o usuário escolher a opção que deseja se voluntariar:

<1> Cadastrar Projeto Distribuir Alimentos

<2> Cadastrar Projeto Trabalho Voluntário

<3> Sair

O Sistema deverá solicitar a entrada de dados. Use a classe Scanner ou JOptionPane

A opção <1> deverá criar um objeto do tipo **DistribuicaoAlimento**, executar o método **validaProjeto()** e o **imprimeProjeto()**

A opção <2> deverá criar um objeto do tipo **TrabalhoVoluntario**, executar o método **validaProjeto()** e o **imprimeProjeto()**

O programa deverá ficar em loop e só será encerrado se o usuário digitar o número 3.

Exercício 2)

1. Implemente a classe Pessoa, contendo os atributos “nome”, “email” e “telefone”, pelo menos.
2. Crie uma classe para um “Funcionario”. Ela deve herdar de Pessoa os seus atributos, além de adicionar: o departamento onde trabalha, o salário dele (double), a data de entrada no banco (string), o RG dele (string) e um valor booleano que indique se o funcionário está na empresa no momento ou se já foi embora.
3. Crie uma classe Empresa que possua “nome”, “cnpj”, “qtde_de_funcionario” e um array de objetos da classe Funcionario (o array pode armazenar até 100 funcionários).
4. Crie os seguintes métodos para a classe Funcionario, criada anteriormente:
 - a) método “bonificar”, que aumenta o salário do funcionário de acordo com o parâmetro passado como argumento;
 - b) método “demitir”, que não recebe parâmetro algum, apenas modifica o valor booleano indicando que o funcionário não trabalha mais na empresa;
 - c) método “mostrarDados”, que simplesmente imprime todos os atributos de um funcionário.

5. Crie 10 objetos da classe Funcionário e insira-os em um array de objetos dessa classe. Utilize um for para imprimir todos os atributos dos objetos do array.
6. Crie um objeto Empresa e adicione os 10 Funcionários criado.

Exercício 3)

1. Criar uma Classe Artigo com os seguintes atributos:

```
private String titulo;  
private String resumo;  
private String autores;
```

Fazer os métodos setters e getters e o construtor

2. Crie uma classe denominada Edição que contenha os seguintes atributos:

```
private int numero;  
private int volume;  
private String dataEdicao;  
private int tiragem;  
  
private ArrayList<Artigo> artigos;
```

Fazer os métodos setters e getters e o construtor.

Criar o método que adiciona Artigos em uma Edição.

3. Crie uma classe denominada RevistaCientifica que contenha os seguintes atributos:

```
private String titulo;  
private long issn;  
private String peridiocidade;  
  
private Edicao edicoes;
```

Fazer os métodos setters e getters e o construtor

Fazer um método que adiciona uma **Edição** na RevistaCientífica

4. Criar um main para manipular os objetos:

Adicione um ArrayList de 10 objetos da classe Artigos.

Crie um objeto do tipo Edição, adicione todos os artigos.

Crie uma Revista Científica e adicione a Edição.