**POO – Laboratório Classe Abstrata e Interface (parte 2)**

**NOME:** Lucca Magnino

**MATRÍCULA:** 202401805

**Parte teórica:**

1. Sobre o Diagrama de Classe LabClasseAbstrataInterface.uxf responda as seguintes questões?
2. Quais são as classes concretas, abstratas e interface do Diagrama de Classe?

As classes concretas são:   
- LabClasseInterface::ProfessorDaEducacaoBasica

- LabClasseInterface::Aluno

- LabClasseInterface::ProfessorUniversitario  
- LabClasseInterface::TestaSolucao  
- LabClasseInterface::Prova  
  
As classes abstratas são:  
- LabClasseInterface::Professor  
  
As interfaces são:  
- LabClasseInterface::FuncionarioAssalariado  
- LabClasseInterface::Estudante

1. Descreva as partes do Diagrama de Classe da interface FuncionarioAssalariado?

- **<<interface>>** = o estereótipo indica que é uma interface e não uma classe.

- **LabClasseInterface::**FuncionarioAssalariado = LabClasseInterface agrupa todas as classes e interfaces, já FuncionarioAssalariado é o nome da interface.

- **~receberSalario(int nTurmas): void** = o ~ indica que é acessado somente pelo pacote LabClasseInterface, o receberSalario(int nTurmas): é o método ou função que a interface irá realizar, já o “void” indica o tipo de retorno.

1. Descreva as partes do Diagrama de Classe da classe Aluno?

- **LabClasseInterface::Aluno** = o LabClasseInterface é o pacote que a classe de nome Aluno pertence.

- **-nome: String** = Atributo privado do tipo String que armazena o nome do Aluno.

- **-nMatricula: int** = Atributo privado do tipo int que armazena a matricula.

- **+Aluno(String nome, int nMatricula): ctor** = método público que recebe nome e matrícula como parâmetros.

- **+fazerProva(Prova prova): void** = método público que recebe um objeto do tipo Prova como parâmetro e é do tipo void.

- **+getNome(): String** = método público que retorna o valor da variável nome.

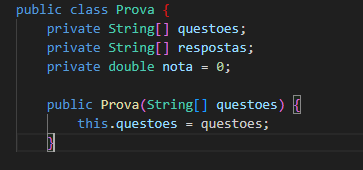
- **+setNome(String nome): void** = método público que modifica o valor da variável nome.

- **+getnMatricula(): int** = método público que retorna o valor da variável matrícula.

- **+setnMatricula(int nMatricula): void** = método público que modifica o valor da variável matrícula.

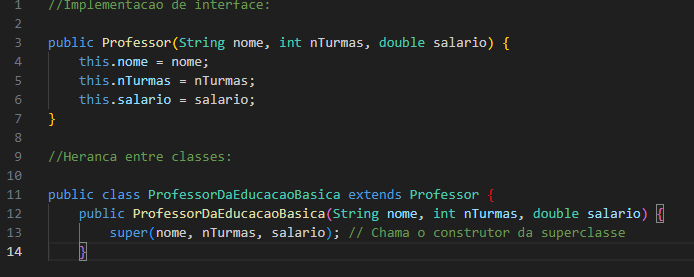
- **+estudar(): void** = método público que implementa a interface Estudante.

1. Descreva resumidamente o código do dialeto XML da Classe Prova e onde se pode encontrá-lo no Diagrama?

O código do dialeto XML é encontrado nos “retângulos” do diagrama, contendo o tipo, o nome, os atributos e os métodos de cada parte de código. Um exemplo de como esse código seria disposto é o seguinte:  


1. Descreva resumidamente o código do dialeto XML da seta que une uma classe a interface e a seta que determina uma relação de herança entre as classes?

A implementação de interfaces e a relação de herança entre classes pode ter o código representado da seguinte forma:



1. Sobre o código-fonte da interface Estudante responda as seguintes questões?
2. Qual é o nome do método e quais são os parâmetros de entrada e de saída deste método?

**Nome do método:** estudar.

**Parâmetros de entrada:** é uma função sem argumentos, não tem “entrada”

**Parâmetros de saída:** por ser do tipo “void” não tem saída, apenas realiza uma ação.

1. Porque o método da interface termina necessariamente com um ponto-e-vírgula (;)?

Seguindo as regras de sintaxe da linguagem Java, o ponto e vírgula encerra a declaração, indicando que não há uma implementação do método.

1. Quem deve implementar o método desta interface?

A classe concreta Aluno.

1. Sobre o código-fonte da Classe Professor responda as seguintes questões?
2. Que tipo de classe é a Classe Professor e o que é necessário e suficiente para uma classe ser deste tipo?

A classe professor é abstrata, sendo considerada desse tipo por conter um método ou mais sem implementação

1. Porque o método da interface termina necessariamente com um pontoe-vírgula (;)?

Assim como já foi respondido, seguindo as regras de sintaxe da linguagem Java, o ponto e vírgula encerra a declaração, indicando que não há uma implementação do método.

1. Quem deve implementar o método desta interface?

A classe concreta ProfessorUniversitario.

1. A classe apresenta uma anotação @Override para que serve esta anotação?

A anotação serve para sobrescrever outros métodos já implementados, dando “prioridade” a ação descrita pelo método com @Override.

1. Sobre o código-fonte ProfessorDaEducacaoBasica responda as seguintes questões?

**A**) Existe um método com o mesmo nome da classe. O que faz este tipo de método? Explique cada linha do código-fonte deste método.

O método:  
*public* ProfessorDaEducacaoBasica(String nome, int nTurmas, double salario) { super(nome, nTurmas, salario);

}

Esse método inicializa os atributos de uma “SuperClasse” criada anteriormente. A primeira linha serve para nomear o método e implementar os parâmetros (nome, nTurmas e salário). Já a segunda linha chama a “SuperClasse” e passa os parâmetros criados anteriormente na classe Professor.

B**)** Que tipo de método é o método elaborarProvas()? A implementação dele é a mesma em todo as classes do projeto onde ele existe? Explique cada linha do código-fonte deste método.

O método elaborarProvas() é abstrato. A sua implementação não é igual a das outras, visto que as subclasses que herda devem fornecer suas próprias implementações.

O código:  
public Prova[] elaborarProvas() {

Prova[] provas = new Prova[4];

provas[0] = new Prova(new String[]{ "P1Q1", "P1Q2", "P1Q3", "P1Q4", "P1Q5" }); provas[1] = new Prova(new String[]{ "P2Q1", "P2Q2", "P2Q3", "P2Q4", "P2Q5" }); provas[2] = new Prova(new String[]{ "P3Q1", "P3Q2", "P3Q3", "P3Q4", "P3Q5" }); provas[3] = new Prova(new String[]{ "P4Q1", "P4Q2", "P4Q3", "P4Q4", "P4Q5" }); return provas;

}

A primeira linha define o método elaborarProvas do tipo Prova[]. A segunda linha cria um vetor com 4 espaços na memória. Das linhas 3 até a 6 os espaços de cada vetor, do 0 ao 3, criando uma nova instância Prova com um conjunto de questões de acordo com cada prova. Por último, retorna o vetor provas, criado na segunda linha.

1. Sobre o código-fonte TestaSolucao responda as seguintes questões?
2. Quais objetos são instanciados no código e de quais são suas classes de origem?

Seguem os objetos e suas classe de origem, respectivamente:

-maria: ProfessorUniversitario

-jose: ProfessorDaEducacaoBasica

-PrimeiroNomedoEstudante: Aluno

-pessoasQueEstudam: Criado no próprio método TestaSolucao

1. Quais os métodos usados nesta classe e em qual classe ele foi implementado?

Seguem os métodos e sua classe de implementação, respectivamente:

-estudar(): Estudante

-elaborarProvas(): Professor

-fazerProvas(): Aluno

-corrigirProva(Prova prova): Professor

-getNota(): TestaSolucao

**Parte prática:**

Fiz os exercícios e os enviarei em outros arquivos dentro do zip, usei as informações no apêndice da tarefa.