POO aula 10: Herança(parte 1)

Pilares da POO(EHP)

**-E**ncapsulamento

**-H**erança

**-P**olimorfismo

Herança: Permite basear uma nova classe na definição de uma outra classe previamente existente, sendo a classe previamente existente chamada de **Classe mãe(ou Pai ou Progenitora ou Superclasse)** e a classe que herda os atributos e métodos da classe mãe, é chamada de **Classe filho(ou Subclasse)**.

Perceba que a herança será aplicada tanto para características(ou seja herda atributos) quanto para comportamento(herdando também métodos).

Exemplo da aula: Estamos dentro de uma escola e temos que definir os dados relacionados à alunos, professores e funcionários. Onde suas características e funções tem em comun os atributos(nome,idade, sexo) e métodos em comum(fazerAniv()), o objetivo é fazer um algoritmo que defina uma classe mãe com todos esses atributos e métodos, herdando eles para as classes filho(aluno, professor e funcionário), onde cada classe vai ter seus atributos e métodos únicos.

Classe Pessoa

privado nome: Caractere

privado idade: Inteiro

privado sexo: Caractere

//Métodos

público método fazerAniv()

idade = (getIdade() + 1)

FimMétodo

//Métodos Especiais, fazer getters e setters de todos os atributos

FimClasse

=========================================================

Classe Aluno estende Pessoa

//Atributos

privado matr: Inteiro

privado curso: Caractere

//Métodos

público método cancelMatr()

\*Fazer questionário

Escreva(“Tem certeza que quer cancelar matrícula?”)

Se (apertado sim) então

matr = nulo

Senão então

Escreva(“Matrícula não cancelada”)

FimSe

FimMétodo

//Métodos Especiais, fazer getters e setters de todos os atributos

FimClasse

=========================================================

Classe Professor estende Pessoa

//Atributos

privado especialidade: Caractere

privado salario: Real

//Métodos

público método receberAum($valor\_aumento)

salario = getSalario() + $valor\_aumento

FimMétodo

//Métodos Especiais, fazer getters e setters de todos os atributos

FimClasse

=========================================================

Classe Funcionario estende Pessoa

//Atributos

privado setor: Caractere

privado trabalhando: Lógico

//Métodos

público método mudarTrabalho()

Se (trabalhando) então

trabalhando = false

Senão então

trabalhando = true

FimSe

FimMétodo

//Métodos Especiais, fazer getters e setters de todos os atributos

FimClasse

=========================================================

\*Programa Principal Index

p1 = new Pessoa()

p2 = new Pessoa()

p3 = new Professor()

p4 = new Funcionario()

faz método construtor por favor(Para todos)

falling behind laufray