

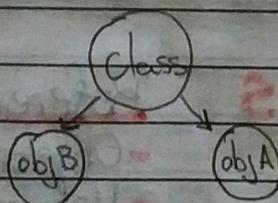
PROGRAMAÇÃO

P1

Programação de Alto Nível forma de modelar o mundo real

Propriedades e comportamentos são armazenados em objetos

- Propriedade atributo dos objetos
- Comportamento ações operações que o objeto realiza
- Classe modelo abstrato que define tipo do objeto da classe
- Objeto instância de classe



Tipos de Dados primitivos

- byte
- short
- int
- long
- float
- double
- boolean
- char

Package organizar classe em um projeto, evitar que classe com mesmo nome conflitem entre si (devem ser criados na ordem inversa do domínio na net)

Import br.com.google.MinhaClasse;

Variáveis

- De instância: atributos não estáticos
- Da classe: estáticos
- Locais dentro dos métodos
- Parâmetros: entrada para os métodos

tipo variável = valor;

Métodos

Similares as funções em C, porém pode ocorrer a sobrecarga de métodos (mesmo nome ≠ quant de parâmetro)

Passagem por Valor VS

- tipos primitivos

Referência

- Objetos
- Vetores
- Matrizes

EM JAVA TODA CLASSE HERDA DE OUTRA CLASSE

Se não explicitado herda java.lang.Object

Construtor mesmo nome da classe não possui retorno
Inicializa as variáveis de instância

Método to String

System.out.println(objeto),
System.out.println(objeto.toString())

Modificadores De Acesso

Meio de ocultar atributos e métodos de uma classe para as outras classes

Nível Superior (Na classe)

Public - publica a todos as outras

package private (sem modificador) visivel apenas no seu package

Nível Membro

Public na classe e fora

package private dentro do package

protected classe que herdam dessa classe + mesmo package

Private apenas na classe (classe) get (acessor) set (mutador)

	Classe	Package	Subclasse	Fóra do Package
Public	✓	✓	✓	✓
protected	✓	✓	✓	✗
sem modificador	✓	✓	✗	✗
Private	✓	✗	✗	✗

STATIC

O tipo em que o variável e o método existe na classe independente da criação de objetos

(Classe. variável),

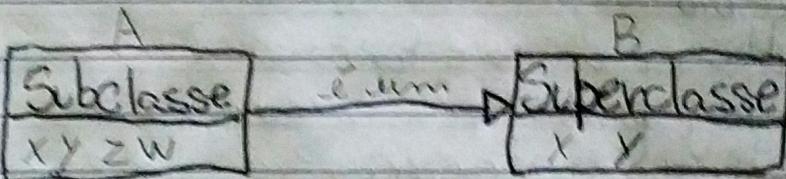
(Classe. método);

HERANÇA é um

Novas classes de objetos aproveitam características de classes já existentes

Subclasse herda da Superclasse

Composição: tem um Relacionamento



O que tem na subclasse não tem na superclasse
já se contrário é válido

public class A extends B {

3

A: Subclasse

B: Superclasse

CONSTRUTORES NUNCA SÃO HERDADOS

Se chamar excessivas na hierarquia para inicializar os variáveis de instância de cada classe

Se houver 2 construtores usar super(), para saber qual pai quer chamar ele vai saber qual usar pela quantidade de parâmetros

todas as linhas tem um exemplo de classe object

public class Estudante { } Classe for e igual

Estudante e = new Estudante(); instanciar objeto

NovoClasse() { } Construtor notes = new float[TAM]; vetor estatico

super(param); no chamar construtor específico da Superclasse

public Class A extend B { } Herança

package br; import br.com.site.MinhaClasse; import package quando classe
import br.com.site.MinhaClasse; não pertence ao mesmo diretório

System.out.println(" "); print java

import java.util.Scanner;

Scanner obj = new Scanner(System.in); Pegar um dado
int valor1 = obj.nextInt();
String s = obj.nextLine();

import java.util.ArrayList;
private ArrayList<Tipo> lista;

ArrayList lista = new ArrayList();

Construtor...

lista = new ArrayList<>(); usar array list
lista.add(item)

DIAGRAMA UML

NAME
ATTRIBUTO
METODOS

no nome : tipo || name : tipo = valor
 no metodo (param) : tipo - retorno

- + Pública
- # Protegido
- Privado
- ~ Pacote
- / Derivado

→ Dependência (classe usa informação da outra)
 existem uma sem outra

→ Associação (faz referência a outra)

- (o classe A está dentro da outra B)
 - ↳ Unidirecional
 - ↳ bidirecional

→ Agregação (classe é coleção de outra classe)
 (quem agrupa)

→ Composição (quando uma classe é extinta a outra também é)

Superclasse Subclasse
 ▶ Generalização / Especialização (Herança)