PET SI
SISTEMAS DE INFORMAÇÃO

DATE
21/11/2024

OFICINA

DIA 2 - ORIENTAÇÃO A OBJETOS COM JAVA

TÓPICOS ABORDADOS

O QUE É ORIENTAÇÃO A OBJETOS?

- Quando iniciamos os estudos em programação, geralmente aprendemos o que são e como manipular variáveis;

```
String nome = "Nathália";
int codCurso = 314;
double saldoRU = 5.00;
boolean formando = false;
```

- Conforme os programas se tornaram mais complexos e o hardware mais potente, surgiu a necessidade de tornar os códigos mais legíveis e as informações, mais organizadas;

O QUE É ORIENTAÇÃO A OBJETOS?

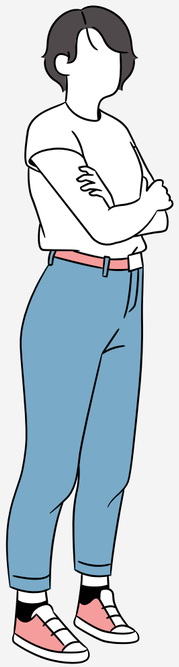
Um objeto da realidade

```
String nome = "Nathália";
int codCurso = 314;
double saldoRU = 5.00;
boolean formando = false;
```

Conjunto de informações sobre o mesmo objeto

As variáveis que se referem a um objeto são denominados **atributos** do objeto.

O QUE É ORIENTAÇÃO A OBJETOS?



String nome = “Nathália”;
int codCurso = 314;
double saldoRU = 5.00;
boolean formando = false;

void insereSaldoRU(double valor);
double consultaSaldoRU();
void almocaRU();
void efetuaMatricula(int codDisciplina);

Funções realizadas pelo objeto

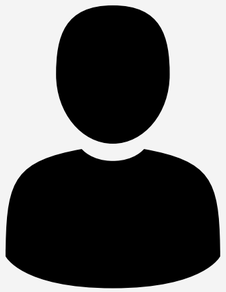
As funções que o objeto executa são denominados **métodos** do objeto.

O QUE É ORIENTAÇÃO A OBJETOS?

String nome = “Nathália”;
int codCurso = 314;
double saldoRU = 5.00;
boolean formando = false;

String nome = “Leonardo”;
int codCurso = 314;
double saldoRU = 15.00;
boolean formando = false;

String nome = “Amanda”;
int codCurso = 314;
double saldoRU = 0.00;
boolean formando = true;



Classe Aluno

String nome;
int codCurso;
double saldoRU;
boolean formando;

void insereSaldoRU();
double consultaSaldoRU();
void almocaRU();
void efetuaMatricula();

Uma classe é o formato genérico de um objeto: todos os objetos terão os atributos e os métodos da classe à qual pertence;

CRIANDO UMA CLASSE

Em linhas gerais, uma classe deverá conter, no mínimo:

- Atributos;

```java
public class Aluno {

    private String nome;
    private int codCurso;
    private double saldoRU;
    private boolean formando;
```

As palavras reservadas **public, private** e **protected** indicam o que pode (ou não) ser acessado por outras classes.

É uma boa prática manter os atributos da classe privados ou protegidos, sendo acessados somente através de métodos chamados Getters.

CRIANDO UMA CLASSE

Em linhas gerais, uma classe deverá conter, no mínimo:

- Construtor: um método “especial” que instancia (cria) um objeto da classe;

```java
public Aluno(String nome, int codCurso, double saldoRU, boolean formando) {
    this.nome = nome;
    this.codCurso = codCurso;
    this.saldoRU = saldoRU;
    this.formando = formando;
}
```

Você pode criar vários métodos construtores, inclusive um construtor vazio (sem o corpo da função).

CRIANDO UMA CLASSE

Em linhas gerais, uma classe deverá conter, no mínimo:

- Getters: métodos que outras classes utilizam para acessar os atributos da classe;

```java
public String getNome() {
    return nome;
}

public int getCodCurso() {
    return codCurso;
}

public double getSaldoRU() {
    return saldoRU;
}

public boolean getFormando() {
    return formando;
}
```

CRIANDO UMA CLASSE

Em linhas gerais, uma classe deverá conter, no mínimo:

- Setters: métodos que outras classes utilizam para modificar os atributos da classe;

```java
public void setNome(String nome) {
    this.nome = nome;
}

public void setCodCurso(int codCurso) {
    this.codCurso = codCurso;
}

public void setSaldoRU(double saldoRU) {
    this.saldoRU = saldoRU;
}

public void setFormando(boolean formando) {
    this.formando = formando;
}
```

PILARES DA ORIENTAÇÃO A OBJETOS

HERANÇA

- Você pode criar uma classe que herdará os atributos e métodos de sua classe-mãe;
- Organiza o código e evita repetições desnecessárias.

```java
public class Bolsista extends Aluno {

    int numBeneficio;
    double valorBolsa;


    public Bolsista(String nome, int codCurso, int numBeneficio, double valorBolsa) {
        super(nome, codCurso);
        this.numBeneficio = numBeneficio;
        this.valorBolsa = valorBolsa;
    }

}
```

PILARES DA ORIENTAÇÃO A OBJETOS

POLIMORFISMO - SOBRESCRITA

- Um método herdado da classe-mãe é sobrescrito pela classe-filha, ou seja, a mesma função terá um comportamento diferente quando executado pela classe-filha;

```
public void almocaRU() {  no usages
    saldoRU = saldoRU - 2.5;
}
```

Na classe Aluno

```
@Override  no usages  new *
public void almocaRU() {
    saldoRU = saldoRU - 2.75;
}
```

Na classe Bolsista

PILARES DA ORIENTAÇÃO A OBJETOS

POLIMORFISMO - SOBRECARGA

- Quando você coloca o mesmo nome para um método, mudando sua assinatura (os parâmetros passados);
- Pode ser implementado dentro da mesma classe.

```java
public void colocaSaldoRU(double valor) {  no usages  new *
    saldoRU = saldoRU + valor;
}

public void colocaSaldoRU(Aluno aluno, double valor) {
    aluno.saldoRU = aluno.saldoRU + valor;
    this.saldoRU = this.saldoRU - valor;
}
```

Exemplo de sobrecarga
na classe Aluno

APÊNDICE: CLASSES/IMPLEMENTAÇÕES ÚTEIS EM JAVA

- Estruturas de dados: Queue, ArrayList, Set - (ler mais sobre Collections);
- Random, para gerar valores aleatórios;
- Timer, para "agendar" e repetir trechos de código de tempos em tempos.

EXERCÍCIO PRÁTICO

NÍVEL 1: LÓGICA DE PROGRAMAÇÃO

- Altere o método fazUmaJogada() para que ele não permita ao usuário inserir dados inválidos (diferentes de 0, 1 ou 2);

NÍVEL 2: USO DE CLASSES

- Crie uma classe Maquina que herde de Jogador, e sobrescreva a função fazUmaJogada() para criar jogadas aleatórias.

PET SI
SISTEMAS DE INFORMAÇÃO

Obrigada pela participação!

Não perca as atividades promovidas pelo PET-SI! Acompanhe as redes sociais:

@petsi_ufsm

Link para o Discord