

FIAP GRADUAÇÃO

Agenda

- Revisão UML
- Classe
- Atributo
- Método

UML

UML é uma forma de representar programas em forma gráfica.
Vamos utilizar o diagrama de classes para o seguinte caso:

A FIAP vai abrir um banco e os clientes serão os alunos. Neste banco será possível sacar, depositar e imprimir os dados da conta. É importante saber que o cadastro do aluno neste banco será diferente do cadastro da faculdade, então devemos cadastrar o nome, a data de nascimento (menores de idade não podem ser clientes), o cpf e a senha de acesso.

Classes

- As classes são a representação (a forma) dos objetos no nosso sistema.
- Cada classe poderá ter múltiplas instâncias (objetos)

Classes

A classe é definida por um arquivo com a extensão “.java”. O nome do arquivo deve ser igual ao nome da classe.

Vamos utilizar o UpperCamelCase para nomes de classes.

```
[modificador de visibilidade] class [NomeDaClasse] {  
    [atributos]  
    [construtores]  
    [métodos]  
}
```

Classes

```
public class Pessoa {  
  
    private String nome;  
    private int idade;  
  
    public Pessoa(){  
  
    }  
  
    public void apresentacao(){  
        System.out.println("Olá. Meu nome é "+nome);  
        System.out.println("e eu tenho "+idade+" anos");  
    }  
}
```

→ Declaração dos atributos

→ Declaração dos Construtores

↓ Declaração dos atributos

Atributos

- São as características de cada classe
- Cada atributo deve ter um tipo que pode ser:
 - Primitivo (int, float, double, boolean etc)
 - Referência (Pessoa, String, Carro, Conta)

Atributos

Os atributos são os primeiros a serem declarados em uma classe.

Vamos utilizar o lowerCamelCase para nomes de atributos.

[modificador de visibilidade] [tipo] [nomeDoAtributo]

Atributos

```
public class Pessoa {  
    private String nome;  
}
```

↓ ↓ ↓

visibilidade tipo nomeDoAtributo

Construtores

- O construtor é a forma que temos de criar um objeto (uma instância) de uma classe
- Toda classe tem que ter um construtor. Caso nenhum construtor seja declarado, o Java define um construtor “default” que não recebe nenhum parâmetro

Construtor

Cada classe pode ou não ter construtores declarados. Se declararmos um construtor, o construtor default não irá mais funcionar.

Para criar um construtor temos que utilizar o mesmo nome da classe:

```
[modificador de visibilidade] [NomeDaClasse] (  
    [parâmetros]  
) {  
    [corpo do construtor]  
}
```

Construtores

```
private int idade; ➡ Algum atributo
```

```
public Pessoa(int idade) {  
    this.idade = idade; ➡ Construtor que recebe  
    alguma idade como  
    parâmetro  
}
```

Métodos

- São as ações ou comportamentos de cada classe
- Define o que a classe faz
- O método só é executado quando invocado através do objeto

Métodos

Os atributos são os primeiros a serem declarados em uma classe.

Vamos utilizar o lowerCamelCase para nomes de atributos.

```
[modificador de visibilidade] [tipoDeRetorno] [nomeDoAtributo] (  
    [parâmetros]  
) {  
    [corpo do método]  
}
```

Métodos

visibilidade retorno nomeDoMetodo parâmetros

```
public void apresentacaoParaOutraPessoa(Pessoa pessoa){  
    System.out.println("Olá "+pessoa.getNome()+". Meu nome é "+nome);  
    System.out.println("e eu tenho "+idade+" anos");  
}
```

Corpo do método

Métodos

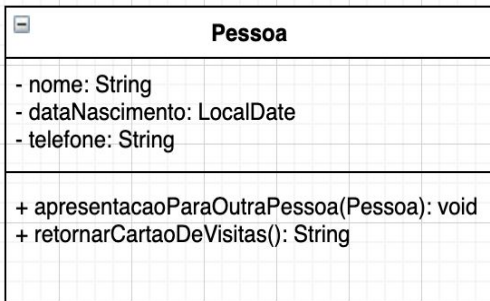
Todo método deve ter o tipo de retorno. Caso o método não tenha nenhum retorno vamos utilizar a palavra reservada “void”.

Se um método tem o tipo de retorno declarado, é obrigatório utilizar a palavra reservada “return” quando quisermos retornar o valor. Este valor precisa ser do tipo declarado e nenhuma linha código será executada no método após o return.

Métodos

```
public String retornarCartaoDeVisitas(){  
    String dadosCartao = "Nome: "+nome+". Telefone: "+telefone;  
    return dadosCartao;  $\Rightarrow$  Retorno do tipo String  
    System.out.println("Dados enviados ");  $\Rightarrow$  Erro de compilação:  
                                           UnreachableCode  
}
```

UML => Class



```
public class Pessoa {

    private String nome;
    private LocalDate dataNascimento;
    private String telefone;

    public void apresentacaoParaOutraPessoa(Pessoa pessoa){
        System.out.println("Olá "+pessoa.getNome()+" Meu nome é "+nome);
    }

    public String retornarCartaoDeVisitas(){
        String dadosCartao = "Nome: "+nome+" Telefone: "+telefone;
        return dadosCartao;
    }

    private String getNome() {
        return this.nome;
    }

}
```

OBRIGADO

FIAP

Copyright © 2020 | Professor Fabio Tadashi Miyasato
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.