

Programação Orientada a Objetos

Contato:
anderson.elias@estacio.br



Estácio

1

Construtores

- Um Construtor especifica como se deve inicializar um objeto;
- Os construtores contém instruções para inicializar objetos;
- O nome do construtor será sempre o igual ao nome da classe;
- O operador **new** invoca o construtor.

2

Construtores

```
public class Cumprimentar {
    private String nome;

    public Cumprimentar(String aNome){
        nome = aNome;
    }

    public String digaOla(){
        String msg = "Olá, " + nome + "!";
        return msg;
    }
}
```

- Os Construtores são geralmente declarados como **public** para permitir que qualquer código em um programa consiga construir novos objetos da classe.
- Diferente dos métodos, os construtores não tem um tipo de retorno.
- Pode usar o construtor apenas com o operador **new**.

3

Construtores

```
public class CumprimentarTeste {
    public static void main(String[] args){
        Cumprimentar mundoCumprimentar = new Cumprimentar("Mundo");
        System.out.println(mundoCumprimentar.digaOla());

        Cumprimentar JoseCumprimentar = new Cumprimentar("José");
        System.out.println(JoseCumprimentar.digaOla());
    }
}
```

- O Propósito é definir o comportamento de um construtor, que utilizado para inicializar as variáveis de instância dos objetos recém-criados.

4

Construtores

```
public class Cumprimentar {
    private String nome;

    public Cumprimentar(String aNome){
        nome = aNome;
    }

    public String digaOla(){
        String msg = "Olá, " + nome + "!";
        return msg;
    }
}
```

O objetos sendo criados mais de uma vez através de uma mesma classe que tem propósitos diferentes

```
public class CumprimentarTeste {
    public static void main(String[] args){
        Cumprimentar mundoCumprimentar = new Cumprimentar("Mundo");
        System.out.println(mundoCumprimentar.digaOla());

        Cumprimentar JoseCumprimentar = new Cumprimentar("José");
        System.out.println(JoseCumprimentar.digaOla());
    }
}
```

5

Construtores

```
public class Aluno {
    private String nome;
    private int idade;
    private String cpf;
}
```

- Toda classe tem implicitamente um construtor que aparece, não recebe parâmetros e não é declarado. Então quando um objeto desta classe é criado ele invoca o construtor desta classe.

- Este construtor implícito tem as características como mostrado na mesma classe ao lado, que é oculto. Mas ele pode ser declarado.

```
public class Aluno {
    private String nome;
    private int idade;
    private String cpf;

    public Aluno() {
    }
}
```

6

Construtores

- Também é possível criar vários construtores para a mesma classe onde cada um tem uma assinatura de método diferente.

```
public class Aluno {
    private String nome;
    private int idade;
    private String cpf;

    public Aluno() {
    }
    public Aluno(String cpf) {
        this.cpf = cpf;
    }
    public Aluno(String nome, int idade, String cpf) {
        this.nome = nome;
        this.idade = idade;
        this.cpf = cpf;
    }
}
```

7

Construtores

Quando uma classe tem mais de um construtor, nos dá opções de invocar o construtor que desejamos.

Então o construtor de uma classe estabelece uma regra a ser seguida para os objetos que forem criados.

```
public class UsarAluno {
    public static void main(String[] args) {
        //Atende ao Construtor sem parâmetro
        Aluno aluno_1 = new Aluno();

        //Atende ao Construtor com um parâmetro
        Aluno aluno_2 = new Aluno("12345678900");

        //Atende ao Construtor com três parâmetros
        Aluno aluno_3 = new Aluno("José", 15, "12345678900");
    }
}
```

8

Dúvidas?



Estácio

