

Modificadores de acesso e atributos de classe

Capítulo V

Modificadores de Acesso

Faz parte de um dos pilares da OO o Encapsulamento. Serve para limitar o acesso aos dados, aumentando o nível de segurança e organização do sistema.



Modificadores de Acesso Java

MODIFICADOR	CLASSE	MESMO PACOTE	PACOTE DIFERENTE (SUBCLASSE)	PACOTE DIFERENTE(GLOBAL)
Public				
Protected				
Default				
Private				

Modificadores de Atributos

- **final**: atributo é uma constante.
- **public**: atributo pode ser acessado por outras classes.
- **private**: atributo só pode ser acessado pela própria classe.
- **protected**: atributo pode ser acessado por classes dentro do mesmo pacote ou pelas subclasses.
- **static**: atributo compartilhado por todos os objetos da classe.

Atributos estáticos ou Atributos de Classe

A palavra **static** é um modificador que faz com que um atributo seja comum para todos os objetos da mesma classe.

É utilizado em quando se deseja ter controle sobre os objetos ou quando se deseja compartilhar informações.

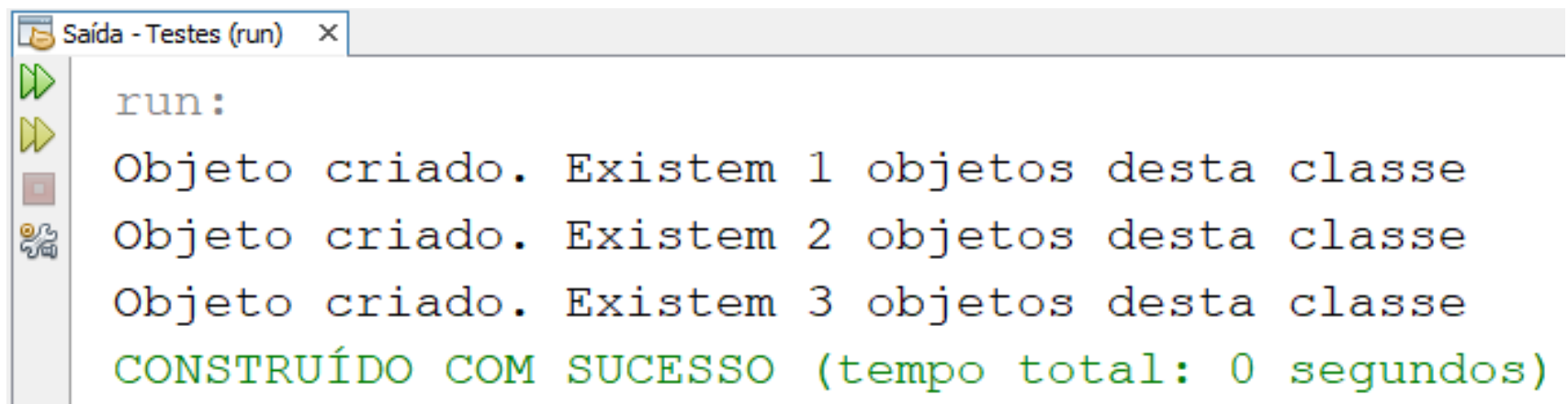
Também são conhecidos como atributos de classe por estarem vinculados a classe e não aos objetos.

Exemplo Controle de Objetos Criados

```
public class Carro {  
    public static int totalCarros = 0;  
  
    public Carro() {  
        totalCarros++;  
        System.out.println("Objeto criado.");  
        System.out.println("Existem " + totalCarros + " objetos desta classe");  
    }  
}
```

Exemplo Controle de Objetos Criados

```
public class Testes {  
  
    public static void main(String[] args) {  
  
        Carro fusca = new Carro();  
        Carro ferrari = new Carro();  
        Carro uno = new Carro();  
  
    }  
}
```



```
run:  
Objeto criado. Existem 1 objetos desta classe  
Objeto criado. Existem 2 objetos desta classe  
Objeto criado. Existem 3 objetos desta classe  
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

Precisamos contar o número de bandidos de uma cidade

```
public class Bandidos {  
    int qtdBandidos;  
  
    int recompensa;  
    String nome;  
    String apelido;  
    Quadrilha cartel;  
  
    public Bandidos() {  
        qtdBandidos++;  
    }  
}
```

Onde está o erro?

O construtor pode nos ajudar, ele é executado toda vez que um objeto é criado

Precisamos contar o número de bandidos de uma cidade

```
public class Bandidos {  
    static int qtdBandidos;  
  
    int recompensa;  
    String nome;  
    String apelido;  
    Quadrilha cartel;  
  
    public Bandidos() {  
        qtdBandidos++;  
    }  
}
```

O modificador `static` permite o compartilhamento de um atributo entre todos os objetos da classe, neste caso *Bandidos*

Onde está o erro?

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Quantidade total de Bandidos cadastrados até hoje é: " + Bandidos.qtdBandidos);  
    }  
}
```

Precisamos contar o número de bandidos de uma cidade

Um desenvolvedor novo no projeto poderia muito bem alterar o valor da variável *qtdBandidos* inadvertidamente

Como nos proteger?

```
public class Bandidos {  
    private static int qtdBandidos;  
  
    private int recompensa;  
    private String nome;  
    private String apelido;  
    private Quadrilha cartel;  
  
    public Bandidos() {  
        qtdBandidos++;  
    }  
}
```

Encapsulamento de dados

E agora?! Como o atributo *qtdBandidos* possui o modificador *private* não podemos acessar diretamente através do seletor

```
public class Main {  
  
    public static void main(String[] args) {  
  
        System.out.println("Quantidade total de Bandidos cadastrados até hoje é: " + Bandidos.qtdBandidos);  
  
    }  
}
```

qtdBandidos has private access in Bandidos

(Alt-Enter mostra dicas)

Então vamos acessar através de métodos!

```
public static int getQtdBandidos() {  
    return qtdBandidos;  
}  
  
public static void main(String[] args) {  
  
    System.out.println("Quantidade total de Bandidos cadastrados até hoje é: " + Bandidos.getQtdBandidos());  
}
```

Encapsulamento de dados com Setter e Getter

```
private int recompensa;
```

```
public int getRecompensa() {  
    return recompensa;  
}
```

Recupera o valor de um atributo

```
public void setRecompensa(int recompensa) {  
    this.recompensa = recompensa;  
}
```

Altera o valor de um atributo

Palavra reservada this

Notou algo de diferente?! A palavra reservada **this** é utilizada para referenciarmos o objeto em questão

```
public Bandidos() {  
    qtdBandidos++;  
}  
  
public Bandidos(String nome, int recompensa) {  
    this();  
    this.nome = nome;  
    this.recompensa = recompensa;  
}
```

Opa!!! Ao utilizarmos **this()** estamos chamando o construtor padrão, desta forma nunca vamos nos esquecer de incrementar a **qtdBandidos**

Vamos falar do projeto?

1. Trabalho até três pessoas
2. Tema “livre” - CRUD
 - i. Lista de sugestão
 - ii. Sem temas duplicados
3. Alunos que fazem BD na mesma equipe
4. Código e Apresentação
5. Apresentação
 - i. Proposta
 - ii. Utilização dos conceitos
 - iii. Questionamentos
6. Avaliação do código



Vamos falar do projeto?

1. Arquivo no Github
2. Sugestão inicial
 - i. Cadastro de clientes e fornecedores
 - ii. Agenda de consulta médica
 - iii. Sistema de controle de estoque (almoxarifado)
 - iv. Sistema de controle de biblioteca
 - v. Sistema de controle das despesas (pagamento de contas)



Exercícios

1. Implemente uma calculadora que realize as seguintes operações entre 2 números: soma, subtrair, dividir e multiplicar através de métodos estáticos.
2. Implemente um sistema gerenciador de cartas de Yu-Gi-Oh! O usuário poderá inserir e listar as cartas em um *deck*. Um deck possui no máximo 50 cartas e cada carta possui uma quantidade de ATK, DEF e um tipo. Crie um menu de opções onde o usuário possa realizar as operações acima e também uma opção para mostrar quantas cartas já foram criadas ao todo.
3. Uma loja de instrumentos musicais está precisando de um software para cadastrar diversos instrumentos e você foi selecionado para esta tarefa. Sabe-se que o número de cordas de determinado instrumento é fixo (por exemplo, violão possui 6 cordas e um violino possui 4, e assim por diante). Você deve criar um sistema que mostre quantos instrumentos existem na loja no total e o número de cada instrumento (quantos instrumentos foram instanciados). Crie um método para mostrar os atributos para cada instrumento.

Obrigado!