

Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina (IFSC) - Câmpus Canoinhas

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Programação Orientada a Objetos

Aluno: Ruan Carlos Binder da Silva

Classe executável

- Deve possuir o método main;
- Um projeto pode possuir mais de uma classe executável;
- Toda as classes executáveis devem possuir a linha `public static void main(String[] args)`.
- Exemplo:

```
1 public class Test {  
2  
3 static void main(String[] args){  
4  
5     System.out.println("Hello World");  
6  
7 }  
8 }
```

Classe

- As classes de programação são projetos de um objeto;
- Possuem características e comportamentos;
- Geralmente representa algo que seja abstrato.
- Exemplo:

```
1 public class Gato {  
2  
3 }
```

Atributos

- Propriedades de um objeto;
- Definem as características de um objeto.
- Exemplo:

```
1 public class Cachorro {  
2  
3     public String nome;  
4     public int peso;  
5     public String corOlhos;  
6     public int quantPatas;  
7 }
```

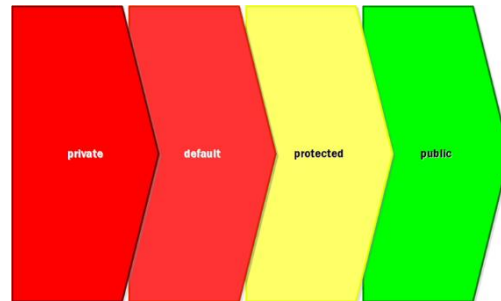
Métodos

- Ações ou procedimentos, que podem interagir e se comunicar com outros objetos;
- Todos os métodos devem sempre ser definidos dentro de uma classe.
- Exemplo:

```
1 class gato{  
2     int tamanho;  
3     String nome;  
4  
5  
6     void miar(){  
7         if(tamanho > 30)  
8             System.out.println("Fink, Fink");  
9         else if(tamanho > 14)  
10            System.out.println("Fiona, Fiona");  
11        else  
12            System.out.println("Flock, Flock");  
13    }  
14 }
```

Encapsulamento

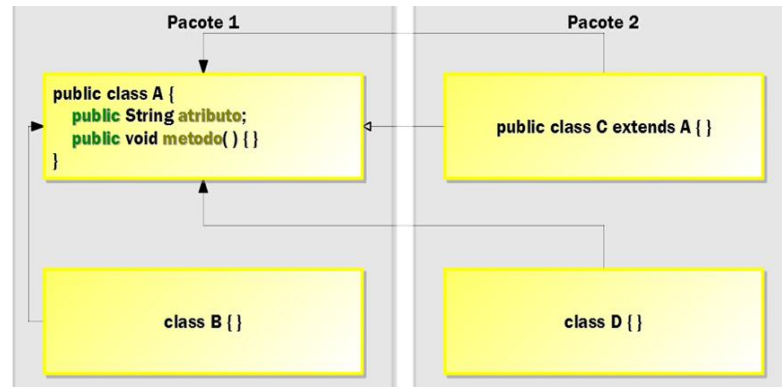
- Responsável pela definição de acessos as classes e seus métodos e atributo;
- Consiste em ocultar a implementação dos métodos e restringir o acesso aos atributos somente através de métodos internos;
- Forma eficiente de proteger os dados;
- O encapsulamento é dividido dois níveis e quatro tipos.



Encapsulamento

Public (público)

- Para que um método seja visível em outras classes, é necessário que ele seja do tipo Public
- O modificador public deixará visível a classe ou membro para todas as outras classes, subclasses e pacotes do projeto Java.
- Exemplo:



Encapsulamento

Private (privado)

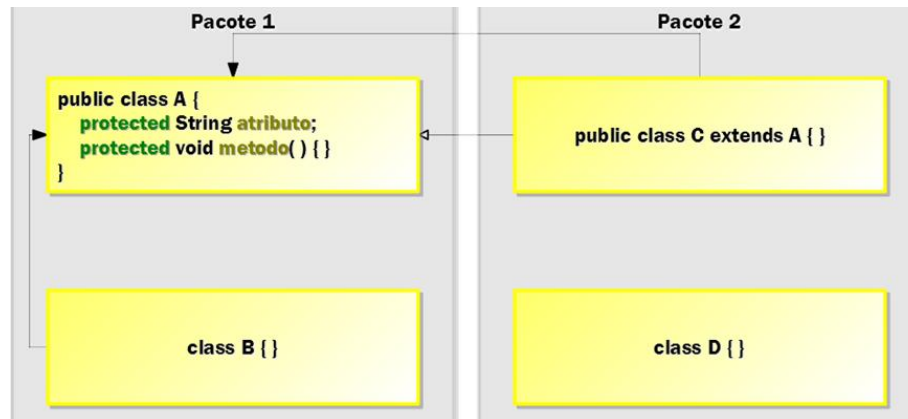
- O modificador `private` que faz com que ninguém consiga modificar, nem mesmo ler, o atributo em questão;
- Um atributo privado só pode ser acessado ou alterado por código escrito dentro da classe na qual ele foi definido;
- Exemplo:

```
1 class Funcionario {  
2     private double salario ;  
3     void aumentaSalario ( double aumento ) {  
4     }  
5 }
```


Encapsulamento

Protected (protegido)

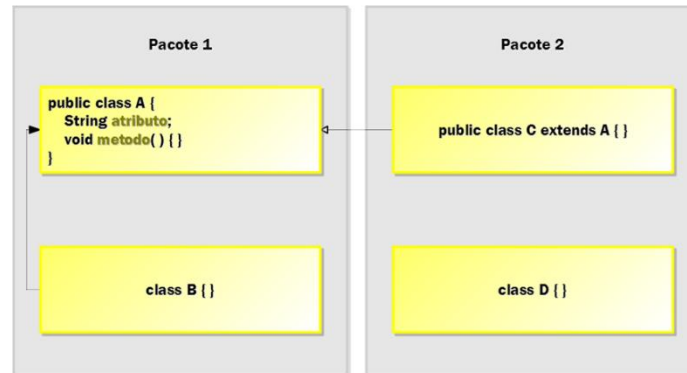
- Visível dentro de atributos e métodos do mesmo pacote ou subclasses;
- O Protected é mistura entre Public e Private;
- Exemplo:



Encapsulamento

Default (padrão)

- Todos os atributos e métodos sem modificador de acesso recebem esse controle;
- Visível dentro do próprio pacote.
- Exemplo:



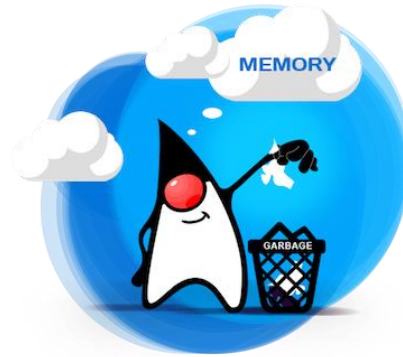
Método construtor

- Inicializa seus atributos toda vez que é instanciado;
 - Determina que ações devem ser executadas;
 - Chamado no momento da criação do objeto através do operador new.
-
- Exemplo:

```
1 public class Employee {  
2     public Employee()  
3     {  
4     }  
5 }  
6 public static void main(String[] args) {  
7     Employee e1 = new Employee();  
8 }  
9 }
```

Método destrutor

- Responsável por liberar os recursos usados pelos objetos durante a execução do programa;
- É executado quando um objeto deixa de existir;
- Realizado de forma implícita.



Criação/destruição de objeto

Criação:

- A criação do objeto é feita através da aplicação do operador `new`.
- Exemplo 1:

```
1 new NomeDaClasse ();
```

Destruição:

- Podemos especificar operações antes da destruição de qualquer objeto de uma classe.
- Exemplo 2:

```
1 void finalize() {  
2 }
```

Referência bibliografia

- CARDOSO, C. Orientação a objetos na prática: aprendendo orientação a objetos com Java. Rio de Janeiro: Ciência Moderna, 2006.
- PINHEIRO, F.A. C. Fundamentos de computação e orientação a objetos usando Java. Rio de Janeiro: LTC, 2006.
- MODIFICADORES DE ACESSO JAVA. Marcus Vinicius de Sousa Vieira, 2013. Disponível em: <<https://marcussousavieira.wordpress.com/2013/04/11/modificadores-de-acesso-java/>>. Acesso em: 21, abr. 2022.
- JAVA: MODIFICADORES DE ACESSO. Gran Cursos, 2021. Disponível em: <<https://blog.grancursosonline.com.br/java-modificadores-de-acesso-parte-i-teoria/>>. Acesso em: 21, abr. 2022.
- O que é uma classe de driver em Java. Delftstack, 2021. Disponível em: <<https://www.delftstack.com/pt/howto/java/driver-class-in-java/>>. Acesso em: 21, abr. 2022.
- Construtores. UNICAMP, 2002. Disponível em: <<https://www.dca.fee.unicamp.br/cursos/PooJava/metodos/construtor.html>>. Acesso em: 21, abr. 2022.

Referência bibliografia

- Declaração de Construtores em Java. Mballem, 2011. Disponível em: <<https://mballem.files.wordpress.com/2011/01/construtoresjava.pdf>>. Acesso em: 21, abr. 2022.
- Objetos e classes em Java. Javatpoint Services. Disponível em: <<https://www.javatpoint.com/pt/objetos-e-classes-em-java>>. Acesso em: 21, abr. 2022.
- Construtor em Java. TutorialCup. Disponível em: <<https://pt.tutorialcup.com/java/constructor-in-java.htm>>. Acesso em: 21, abr. 2022.