UNIP
UNIVERSIDADE PAULISTA

UNICLUSTER

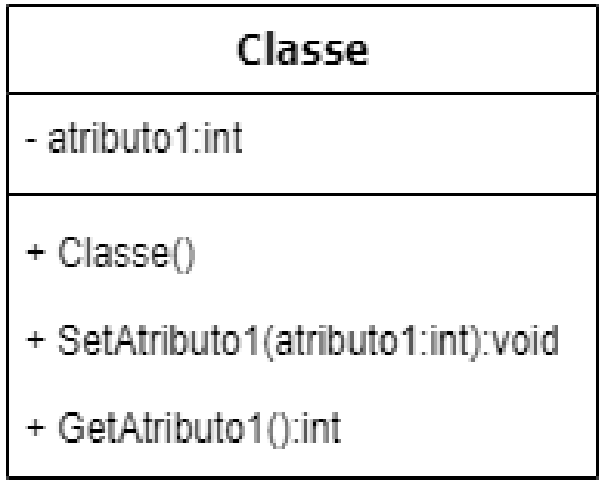
Método Construtor

Carlos Arruda Baltazar
UNIP – Cidade Universitária

Método Construtor

O método construtor determina que ações devem ser executadas e as pré-condições necessárias no momento da criação de um objeto. Em algumas linguagens de programação como Java, C# ou C++, o construtor é definido como um método cujo nome deve ser o mesmo nome da classe e sem indicação do tipo de retorno, bem como a indicação de vazio. O construtor é unicamente invocado no momento da criação do objeto.

Método Construtor




```java
public class Classe
{
    private int atributo1;

    public Classe()
    {
        this.atributo1 = 0;
    }

    public int getAtributo1() {
        return atributo1;
    }

    public void setAtributo1(int atributo1) {
        this.atributo1 = atributo1;
    }
}
```

Método Construtor


```
public class Main
{
    public static void main(String[] args)
    {
        Classe classe1 = new Classe();

        classe1.
    }

}
```

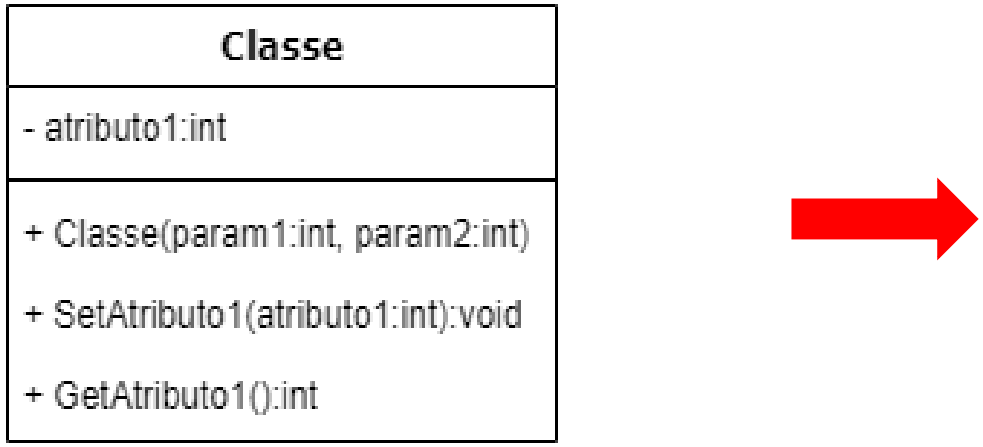
equals(Object obj) : boolean - Object
getAtributo1() : int - Classe
getClass() : Class<?> - Object
hashCode() : int - Object
notify() : void - Object
notifyAll() : void - Object
setAtributo1(int atributo1) : void - Classe
toString() : String - Object
wait() : void - Object
wait(long timeoutMillis) : void - Object
wait(long timeoutMillis, int nanos) : void - Object

Press 'Ctrl+Space' to show Template Proposals

Método Construtor

O retorno do método construtor é uma referência para o objeto recém-criado. O construtor pode receber argumentos, como qualquer método. Usando o mecanismo de sobrecarga, mais de um construtor pode ser definido para uma classe.

Método Construtor




```java
public class Classe
{
    private int atributo1;

    public Classe(int param1, int param2)
    {
        this.atributo1 = param1 + param2;
    }

    public int getAtributo1() {
        return atributo1;
    }

    public void setAtributo1(int atributo1) {
        this.atributo1 = atributo1;
    }
}
```

Método Construtor


```
public class Main
{
    public static void main(String[] args)
    {
        int param1 = 2;
        int param2 = 3;

        Classe classe1 = new Classe(param1, param2);

        classe1.
    }

}
```

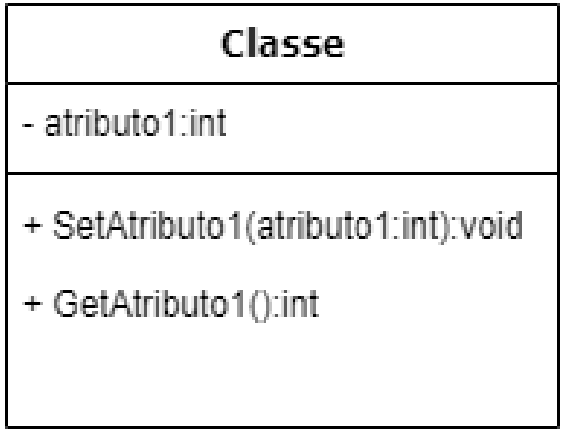
- equals(Object obj) : boolean - Object
- getAtributo1() : int - Classe
- getClass() : Class<?> - Object
- hashCode() : int - Object
- notify() : void - Object
- notifyAll() : void - Object
- setAtributo1(int atributo1) : void - Classe
- toString() : String - Object
- wait() : void - Object
- wait(long timeoutMillis) : void - Object
- wait(long timeoutMillis, int nanos) : void - Object

Press 'Ctrl+Space' to show Template Proposals

Método Construtor

Toda classe tem pelo menos um construtor sempre definido. Se nenhum construtor for explicitamente definido pelo programador da classe, um construtor padrão, que não recebe argumentos, é incluído para a classe pelo compilador. No entanto, se o programador da classe criar pelo menos um método construtor, o construtor padrão não será criado automaticamente.

Método Construtor



```java
public class Classe
{
    private int atributo1;

    public int getAtributo1() {
        return atributo1;
    }

    public void setAtributo1(int atributo1) {
        this.atributo1 = atributo1;
    }
}
```

Método Construtor


```
public class Main
{
    public static void main(String[] args)
    {
        Classe classe1 = new Classe();

        classe1.
    }

}
```

equals(Object obj) : boolean - Object
getAtributo1() : int - Classe
getClass() : Class<?> - Object
hashCode() : int - Object
notify() : void - Object
notifyAll() : void - Object
setAtributo1(int atributo1) : void - Classe
toString() : String - Object
wait() : void - Object
wait(long timeoutMillis) : void - Object
wait(long timeoutMillis, int nanos) : void - Object

Press 'Ctrl+Space' to show Template Proposals

Método Construtor

No momento em que um construtor é invocado, a seguinte sequência de ações é executada para a criação de um objeto:

1. O espaço para o objeto é alocado e seu conteúdo é inicializado com zeros.
2. O construtor da classe base é invocado.
3. Os membros da classe são inicializados para o objeto, seguindo a ordem em que foram declarados na classe.
4. O restante do corpo do construtor é executado.

Método Construtor

Seguir essa sequência é uma necessidade de forma a garantir que, quando o corpo de um construtor esteja sendo executado, o objeto já terá à disposição as funcionalidades mínimas necessárias, quais sejam aquelas definidas por seus ancestrais. O primeiro passo garante que nenhum campo do objeto terá um valor arbitrário, que possa tornar erros de não inicialização difíceis de detectar.

OBRIGADO