

Aula 4

Sumário

Classes e Objetos

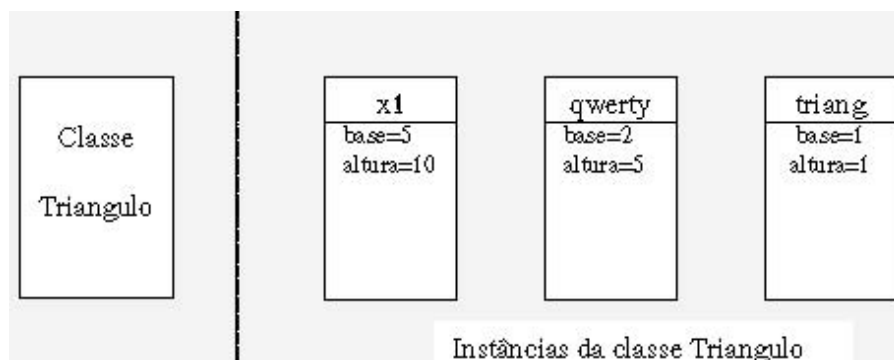
Variáveis de instância, construtores e métodos

Programas de teste

Classes e Objetos

Uma classe é um objeto que serve de molde modelo ou padrão para a construção de objetos semelhantes designados por instâncias da classe. Esses objetos possuem todos a mesma estrutura interna e respondem ao mesmo tipo de mensagens.

Considere-se como exemplo a classe *Triangulo* cujas variáveis de instância (atributos) são capazes de guardar a base e a altura de um triângulo.



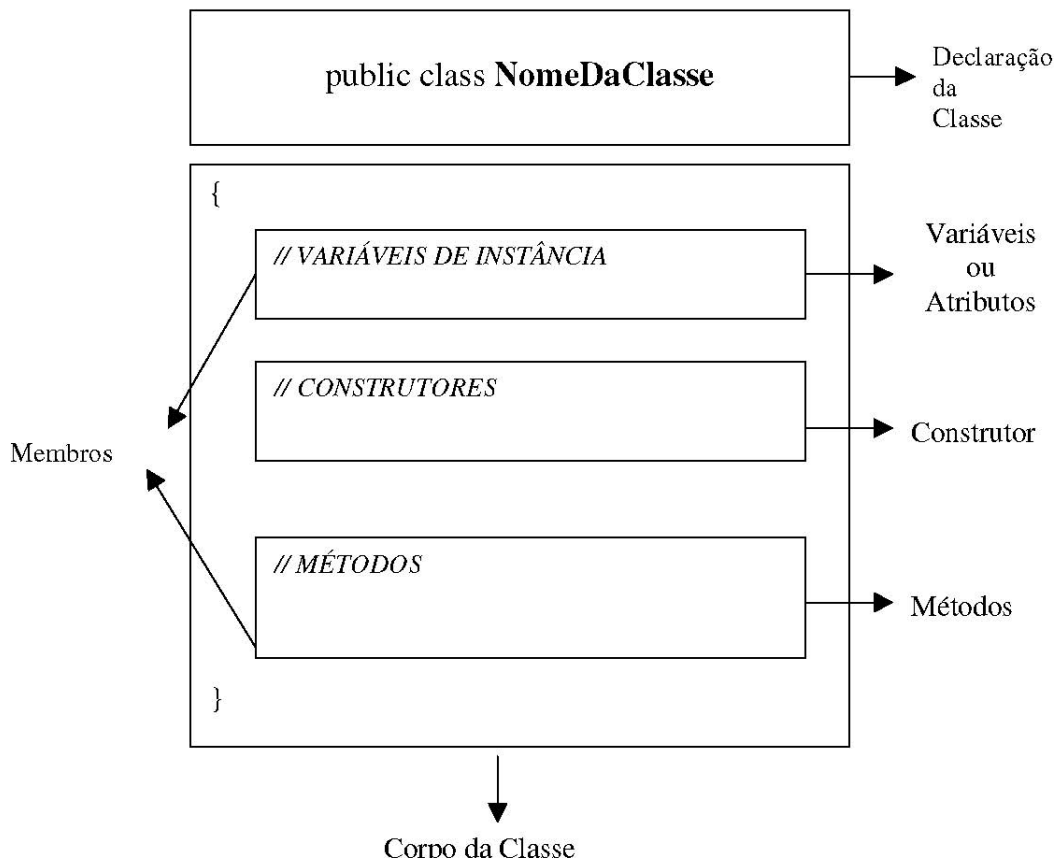
Os objetos *x1*, *qwerty* e *triang* são instâncias da classe *Triangulo*. O objeto *x1*, por exemplo, pode ser criado mediante uma instrução do tipo:

Triangulo x1 = new Triangulo(5,10);

O que fica armazenado em *qwerty* não é o objeto em si, mas sim uma *referência* indicando a posição de memória onde o mesmo está guardado. Antes de criar instâncias de uma dada classe é necessário proceder à implementação da mesma.

Estrutura de uma classe

Estruturalmente uma classe é composta pela respectiva declaração e corpo. Na declaração da classe é indicado o respectivo nome (igual ao nome do arquivo.java onde será guardada a classe). O corpo da classe é composto pelos construtores e membros. Os membros comportam as variáveis de instância (atributos) e os métodos.



Variáveis de instância (Atributos)

Conjunto de identificadores que guardam os valores dos atributos de um determinado objeto. No caso da classe *Triangulo* temos como variáveis de instância a base e a altura:

```
//VARIABLES DE INSTANCIA
private float base;
private float altura;
```

Construtores

Os Construtores caracterizam-se por possuírem a mesma designação que a classe (mesmo nome). São “chamados” quando se pretende criar uma nova instância da classe. Por exemplo, quando se faz:

```
Triangulo x1 = new Triangulo(5,10);
```

estamos chamando um *construtor* da classe *Triangulo* que irá criar uma nova instância da classe inicializando as variáveis de instância, neste caso, com os valores 5 e 10. Os construtores não são membros (não pertencem aos objetos, mas sim à própria classe).

Uma classe poderá possuir mais do que um construtor. No caso da classe *Triangulo* podemos ter um construtor que recebe a base e a altura e cria um novo objeto com esses dados, mas também podemos pensar, por exemplo, num construtor que não recebe nada como argumentos criando, nesse caso, um triângulo de base=altura=1.

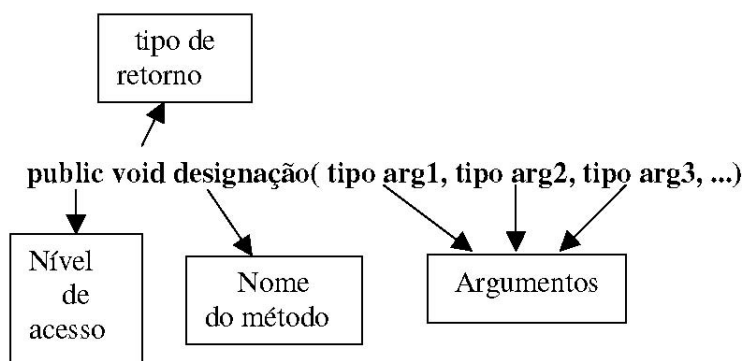
```
//CONSTRUTORES
public Triangulo(){
base=1;
altura=1;
}

public Triangulo(float x,float y){
    base=x; altura=y;
}
```

Métodos

Os métodos definem um conjunto de operações que caracterizam o comportamento das instâncias da classe. É por intermédio dos métodos que podemos comunicar com as instâncias de uma determinada classe.

Declaração de um método:



Nível de acesso

public - o método pode ser evocado pelos métodos da própria classe e por métodos de outras classes.

private - só pode ser evocado no corpo dos métodos da própria classe.

Tipo de Retorno

Tipo dos valores que são retornados pelo método (através da instrução **return**)

void : indica que o método não retorna qualquer valor.

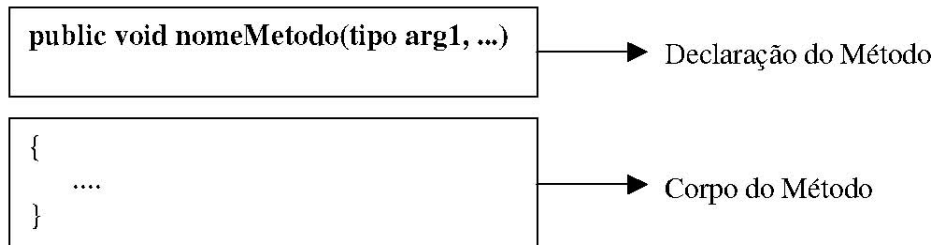
Nome do método

Por convenção deve começar por minúscula e, sendo um identificador, deve respeitar as respectivas regras de escrita.

Argumentos

Um método poderá ter entre zero ou mais argumentos. Os vários argumentos são indicados no formato: *tipo identificador*. Se existirem dois ou mais argumentos estes devem ser separados por vírgulas.

Estrutura de um Método



Métodos Seletores

Permitem acessar aos valores guardados nas variáveis de instância. Têm argumento vazio e limitam-se a devolver o valor de um dado atributo. São habitualmente designados pela palavra *get* seguida da designação do atributo.

Exemplo:

```
//METODOS SELECTORES  
public float getAltura(){ return altura; }  
public float getBase(){ return base; }
```

Métodos Modificadores

Permitem alterar os valores das variáveis de instância. Têm como argumento o novo valor a atribuir a essa variável de instância. Não devolvem qualquer valor. São habitualmente designados pela palavra *set* seguida da designação do atributo:

```
//METODOS MODIFICADORES  
public void setAltura(float x){ altura=x; }  
public void setBase(float x){ base=x; }
```

Outros métodos

Além dos métodos modificadores e seletores podem definir-se outros métodos. Por exemplo, na classe *Triangulo* poderíamos pensar num método *area* definido como se segue:

```
public float area(){ float a=base*altura/2; return a; }
```

Também poderíamos pensar num método de nome *mostra*, capaz de visualizar os atributos referentes a uma dada instância da classe:

```
public void mostra(){ System.out.println("Triangulo:");  
System.out.println("Base:"+base);  
System.out.println("Altura:"+altura); }
```

Este método *mostra* limita-se a apresentar a informação no console. Tem argumento vazio e também não devolve nada. É muito comum considerar um método, designado por *toString*, cuja função é devolver uma *String* com a informação dos valores contidos nos atributos que poderá ser visualizada, ou utilizada de qualquer outra forma, a partir de um ponto exterior à própria classe. Poderíamos ter, por exemplo:

```
public String toString(){  
    String s="Triangulo:\nBase:"+base+"\nAltura:"+altura+"\n";  
    return s;  
}
```

A classe Triangulo completa:

```
public class Triangulo{

    //VARIABLES DE INSTANCIA
    private float base;
    private float altura;

    //CONSTRUTORES
    public Triangulo(){
        base=1;
        altura=1;
    }
    public Triangulo(float x,float y){
        base=x;
        altura=y;
    }

    //METODOS SELECTORES
    public float getAltura(){
        return altura;
    }
    public float getBase(){
        return base;
    }

    //METODOS MODIFICADORES
    public void setAltura(float x){
        altura=x;
    }
    public void setBase(float x){
        base=x;
    }

    //OUTROS METODOS
    public float area(){
        float a=base*altura/2;
        return a;
    }
    public void mostra(){
        System.out.println("Triangulo:");
        System.out.println("Base:"+base);
        System.out.println("Altura:"+altura);
    }
    public String toString(){
        String
        s="Triangulo:\nBase:"+base+"\nAltura:"+altura+"\n";
        return s;
    }
}
```

Programas de teste:

Depois de implementar uma determinada classe (compilar sem erros) é aconselhável proceder a uma série de testes sobre o funcionamento da mesma. Para isso escreve-se um *Programa de Teste*. Este programa deverá consistir na criação (instanciação) de vários objetos da classe e no teste (exaustivo) dos vários métodos implementados. Só depois de garantida a boa implementação da classe poderemos utilizá-la com segurança para o fim a que foi criada (ou mesmo disponibilizá-la para que outros programadores também a possam utilizar).

Um possível programa de teste (incompleto) para a classe *Triangulo*:

```
public class TesteTriangulo{
    public static void main(String args[]){ // exemplo de
alguns testes...

        Triangulo x1;
        Triangulo x2;

        x1=new Triangulo();
        x2=new Triangulo(2,5);

        System.out.println(x1.getBase());
        System.out.println(x2.getBase());

        x1.mostra();
        x2.mostra();

        float a=x1.area();
        System.out.println("Area de x1="+a);

        x1.setAltura(50);
        x1.mostra();

// .... }
}
```