Inicialização de um Objeto de uma Subclasse

Quando se cria um objeto de uma subclasse (classe derivada) ele contém dentro dele um sub-objeto da classe base. Este sub-objeto contém o mesmo conteúdo que teria se se criasse um objeto da classe base.

Para que o sub-objeto da classe base seja inicializado corretamente, Java automaticamente insere uma chamada a um construtor da classe base na sublasse (um construtor da classe base tem o conhecimento e privilégios apropriados para efetuar essa inicialização).

Se o construtor da classe base não tem uma invocação explícita do construtor da superclasse, Java automaticamente chama o construtor por omissão (sem argumentos) da superclasse.

Em qualquer dos casos o sub-objeto da classe base é inicializado (com a execução de um ou mais construtores da classe base) antes dos construtores da subclasse acederem ao objeto.

Exemplo:

```
class A {
    B b1 = new B(1);
    A() { System.out.println("Construtor A()"); }
}

class B {
    B(int i) { System.out.println("Construtor B("+i+")"); }
}

class SubA extends A {
    B b2 = new B(2);
    SubA() { System.out.println("Construtor SubA()"); }
}

public class Sub1 {

    public static void main(String args[]) {
        SubA sa = new SubA();
    }
}
```

- Saída produzida pelo programa:

```
Construtor B(1)
Construtor A()
Construtor B(2)
Construtor SubA()
```

Exemplo:

```
class A {
    A(int i) { System.out.println("Construtor A("+i+")"); }
}

class B {
    B(int i){ System.out.println("Construtor B("+i+")"); }
}

class SubA extends A {
    B b1, b2 = new B(2);
    SubA(int i) { super(i); b1 = new B(i);     }
}

public class Sub2 {
    public static void main(String args[]) {
        SubA sa = new SubA(1);
    }
}
```

- Saída produzida pelo programa:

```
Construtor A(1)
Construtor B(2)
Construtor B(1)
```

Exemplo:

```
class Pao { Pao(){ System.out.println("Pao()"); } }

class Queijo { Queijo(){ System.out.println("Queijo()");} }

class Alface { Alface(){ System.out.println("Alface()"); } }

class Refeicao {
    Refeicao() { System.out.println("Refeicao()"); }
}
class Almoco extends Refeicao {
    Almoco() { System.out.println("Almoco()"); }
}
class Sandwich extends Almoco {
    Pao p = new Pao();
    Queijo q = new Queijo();
    Alface a = new Alface();
    Sandwich() { System.out.println("Sandwich()"); }

    public static void main(String args [])
                              throws java.io.IOException  {
        new Sandwich();
        System.in.read();
    }
}
```

- Saída produzida pelo programa:

```
Refeicao()
Almoco()
Pao()
Queijo()
Alface()
Sandwich()
```

Sumariando o processo de inicialização:

- Carregamento da classe:
  - Inicialização estática da classe base;
  - Inicialização estática da classe derivada.
- Se um objeto da subclasse é criado:
  - Todos os campos de dados (da classe base e da subclasse) são colocados nos valores por omissão (0's binários);
  - Inicializações dos campos de dados da classe base;
  - Chamada do construtor da classe base (o construtor da classe base invocado explicitamente na subclasse ou o construtor por omissão);
  - Inicializações dos campos de dados da classe derivada;
  - Execução do resto do construtor da classe derivada.

Exemplo:

```
class Flor {
    Flor(int i) { System.out.println("Flor(" + i + ")"); }
    void cor(int i){ System.out.println("cor(" + i + ")"); }
}

class Jarra {
    Flor f1 = new Flor(1);
    static Flor f2 = new Flor(2);
    Jarra() {
        System.out.println("Jarra()");
        f3.cor(3);
    }
    static Flor f3 = new Flor(3);
}

public class InicializacaoEstatica {
    public static void main(String[] args) {
        System.out.println("No main");
        new Jarra();                          //(2)
        Jarra.f2.cor(2);                      //(2)
    }
    static Jarra j = new Jarra();             //(1)
}
```

- Saída produzida pelo programa:

```
| Flor(2)          | Sem (1):         | Sem
| Flor(3)          |                  |  (1)
| Flor(1)          |                  | e (2)
| Jarra()          | Flor(2)          |
| cor(3)           | Flor(3)          |
| No main          | No main          |
| Flor(1)          | Flor(1)          |
| Jarra()          | Jarra()          |
| cor(3)           | cor(3)           | No
| cor(2)           | cor(2)           | main
```