

Programação Orientada a Objetos

Módulo 1

Vaux Gomes ¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Campus Jaguaribe

7 de Agosto de 2022

Sumário

Sumário

Variáveis, Métodos, Classes e Blocos Estáticos

- Variáveis Estáticas

- Métodos Estáticos

- Classes Estáticas

- Blocos Estáticos

- Exercício

Variáveis Estáticas

Static: Variáveis, Métodos, Blocos e Classes Estáticas

A palavra reservada **static** pode ser usada para classes, variáveis, métodos e blocos. Um membro estático **pertence à classe em vez de a uma instância específica**, isto significa que é possível acessar o membro estático sem a existência de um objeto (instância de classe).

- ▶ Uma **variável estática** é comum a todas as instâncias (objetos) de uma classe porque pertence ao nível de classe.
 - ▶ Em outras palavras, podemos dizer que **uma única cópia de uma variável estática é criada e compartilhada** entre todas as instâncias de uma classe.
 - ▶ **A alocação de memória desta variável acontece apenas uma vez quando a classe é carregada na memória.**

Variáveis Estáticas

Inicialização de variáveis estáticas

1. São inicializadas quando as classes são carregadas
2. São inicializadas antes de qualquer objeto da classe ser criado
3. São inicializadas antes de qualquer método ou bloco estático da classe ser executado.

► **Variáveis estáticas são úteis:**

- Quando precisamos economizar (gerenciar) memória.
- Quando queremos compartilhar um valor entre todas as instâncias de uma classe (como uma variável global). Isto também pode salvar memória.

► **Sintaxe:**

- `static <tipo_de_dado> <nome_da_variavel>`

► **Exemplo:**

- `static int contador = 0;`

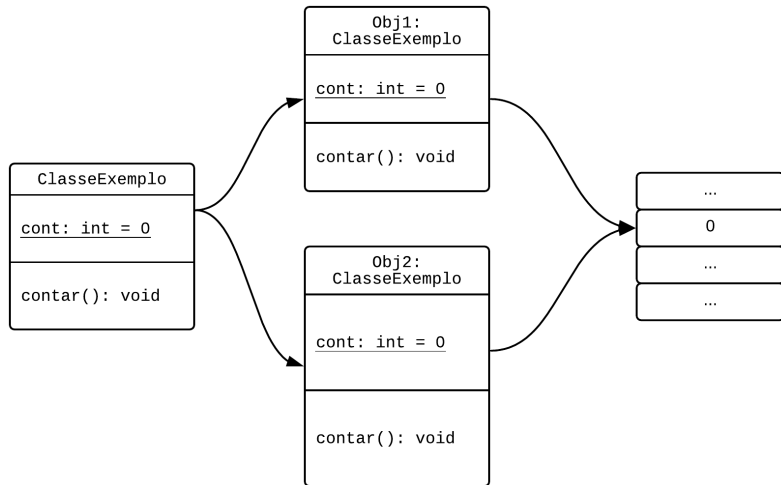
Variáveis Estáticas

Exemplo

```
class ClasseExemplo {  
    static int cont = 0;  
    public void contar() { cont++; }  
  
    public static void main(String args[]) {  
        ClasseExemplo obj1 = new ClasseExemplo();  
        ClasseExemplo obj2 = new ClasseExemplo();  
  
        obj1.contar();  
        obj2.contar();  
  
        System.out.println("Obj1: contador eh = " + obj1.cont);  
        System.out.println("Obj2: contador eh = " + obj2.cont);  
  
        // Obj1: contador eh = 2  
        // Obj2: contador eh = 2  
    }  
}
```

Variáveis Estáticas

Exemplo



```
cont = 0;
//...
obj1.contar();
obj2.contar();
```

Variáveis Estáticas

Exemplo

- ▶ Dependendo do nível de acesso as variáveis estáticas podem ser acessadas através da classe (via nome de classe).

```
ClasseExemplo.cont = 10;
```

- ▶ Variáveis estáticas também podem ser finais

```
public static final int MY_FINAL_VAR = 27;  
public static int MY_VAR = 27;
```

Métodos Estáticos

- ▶ **Métodos estáticos** podem acessar variáveis estáticas sem a necessidade de usar objetos (instanciação)¹.
 - ▶ Métodos estáticos podem ser acessados diretamente em métodos estáticos e não estáticos
- ▶ Sintaxe:
 - ▶ `static <tipo_de_retorno> <nome_do_metodo>()`
- ▶ Exemplo:
 - ▶ `static void contar() { cont++; }`

¹Isto não é possível com variáveis e métodos não estáticos

Métodos Estáticos

Exemplo

```
class ClasseExemplo {  
    static int cont = 0;  
    public static void contar() { cont++; }  
  
    public static void main(String args[]) {  
        ClasseExemplo obj1 = new ClasseExemplo();  
  
        obj1.contar(); // Acessando a partir do objeto  
        ClasseExemplo.contar(); // Acessando a partir da classe  
  
        System.out.println("Objeto: contador eh = " + obj1.cont);  
        System.out.println("Class : contador eh = " + ClasseExemplo.cont);  
  
        // Objeto: contador eh = 2  
        // Class : contador eh = 2  
    }  
}
```

Classes Estáticas

- ▶ Uma classe pode ser estática apenas se esta estiver **aninhada** em outra classe.
 1. Classes estáticas aninhadas não precisam de referência da classe externa.
 2. Uma classe estática não pode acessar membros (métodos, variáveis) não estáticos da classe externa.

Classes Estáticas

Exemplo

```
class ClasseExemplo {
    private static String str = "Java Tem Seus Dramas";

    static class ClasseAninhada {
        public void imprime() {
            // Somente funciona porque str eh uma variavel estatica
            System.out.println(str);
        }
    }

    public static void main(String args[]) {
        // Para criarmos uma instancia de ClasseAninhada nao precisamos ter uma instancia de
        // ClasseExemplo.
        // Classes nao estaticas precisariam

        ClasseExemplo.ClasseAninhada obj;
        obj = new ClasseExemplo.ClasseAninhada();

        obj.imprime();
    }
}
```

Blocos Estáticos

- ▶ **Blocos estáticos** são usados para inicializar variáveis estáticas.
 - ▶ Os blocos são executados **quando a classe é carregada na memória**.
 - ▶ Uma classe pode conter múltiplos blocos estáticos.
 - ▶ Os blocos vão ser executados na sequência em que foram escritos no programa.

Blocos Estáticos

Exemplo

```
class ClasseExemplo {
    static int num;
    static String str;

    // Bloco 1
    static {
        System.out.println("Bloco estatico 1");
        num = 68;
        str = "B1";
    }

    // Bloco 2
    static {
        System.out.println("Static Block 2");
        num = 98;
        str = "B2";
    }

    public static void main(String args[]) {
        System.out.println("Valor de num: " + num);
        System.out.println("Valor de str: " + str);

        // Valor de num: 98
        // Valor de str: B2
    }
}
```

Exercício

1. Adicione o pacote `ifce.jbe.poo.bingo`

2. Implemente a classe **Cartela**

▶ **Atributos**

- ▶ Contendo um atributo array de inteiros (final) chamado `numeros` que é inicializado no construtor da cartela;
- ▶ Adicione outro atributo chamado `marcadas` que conta quantos números já saíram no bingo. Inicialize com valor zero.

▶ **Métodos**

- ▶ Implemente também um método chamado `marcar` que verifica se um número existe na cartela e incrementa o contador `marcadas` se o número existir.
- ▶ Implemente um método chamado `isFim` que retorna verdade se o valor de `marcadas` for maior ou igual à quantidade de números na cartela.

3. Implemente a classe **Bingo**

▶ Atributos estáticos:

- ▶ `TAMANHO_CARTELA` (público, final): Total de números da cartela
- ▶ `MAX_CARTELA` (público, final): Maior número que pode existir numa cartela. Deve ser ao menos 3x maior que `TAMANHO_CARTELA`.

Exercício

- ▶ sorteador (privado, final): Semente = 2
 - ▶ Método estático:
 - ▶ gerarCartela: recebe nada e retorna uma cartela de tamanho = TAMANHO_CARTELA
 - ▶ rodarBingo: recebe um array de cartelas, sorteia vários números até que uma cartela vença. Imprime os números sorteados no final.
4. Inicialize os atributos estáticos em maiúsculo a partir de um bloco estático.
 5. Na classe Main crie 4 cartelas e rode o bingo até uma cartela ganhar a partida.
 6. É possível que a cartela e os números sorteados pelo bingo tenha se repetido.
Tente consertar!