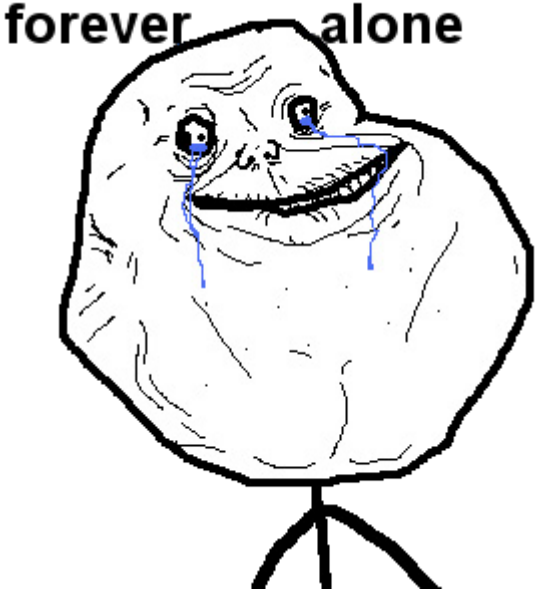
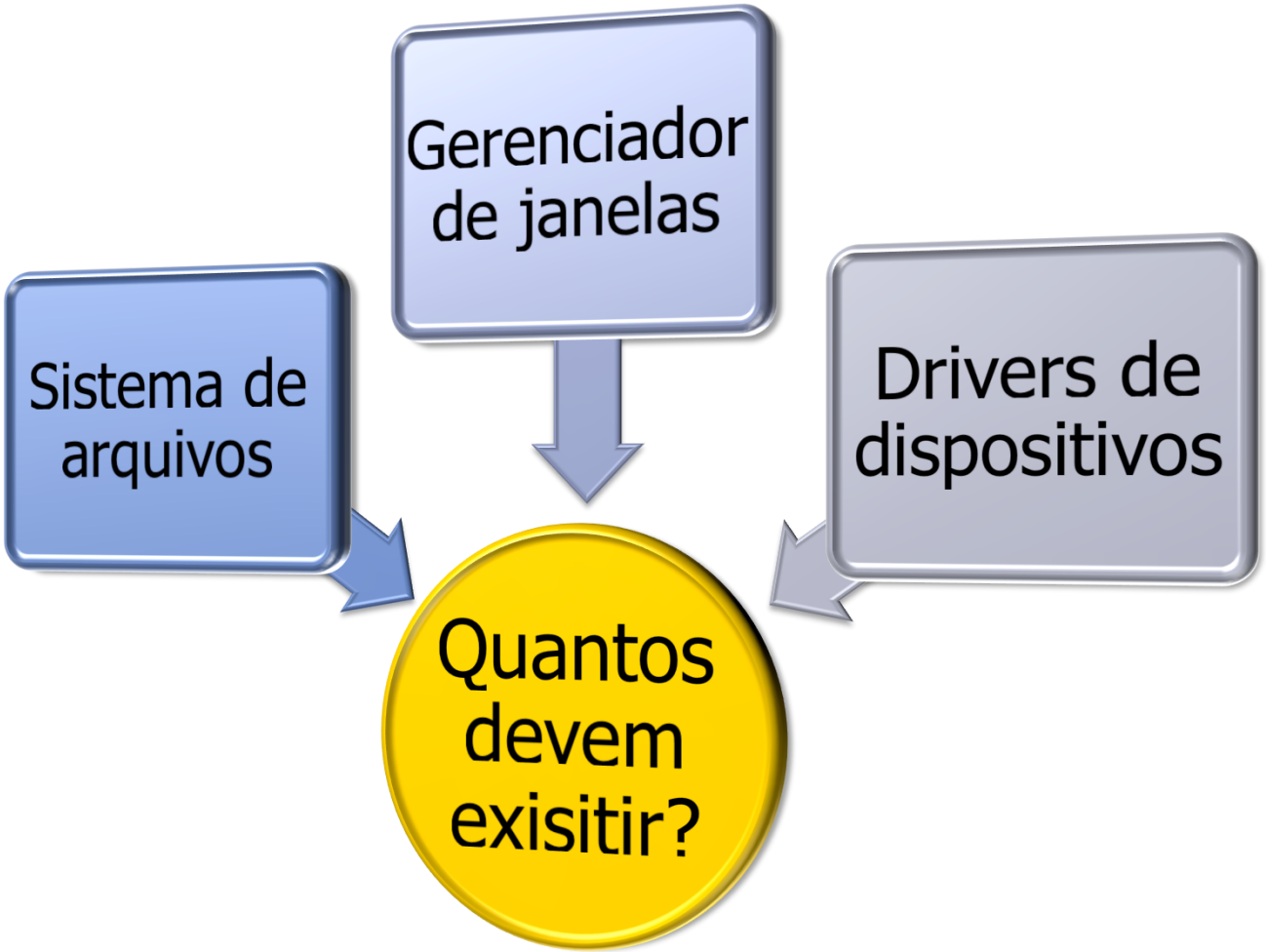
Programação Orientada a Objetos 2
GBC055

Prof. Henrique Fernandes

Padrão Singleton



Motivação


Gerenciador de janelas
Sistema de arquivos
Drivers de dispositivos
Quantos devem exisitir?

Singleton
Motivação

- Contextos em que só deva existir um único objeto de uma classe

- Mais de uma instância de certos objetos podem causar problemas na execução de um programa
- Importante quando um determinado objeto possui muitos recursos

Singleton
Motivação

Global

Acesso único

Uma maneira de garantir que haja somente um único objeto de uma determinada classe

Criando um objeto

- Como fazer para criar um único objeto?
  - new MeuObjeto();
- É possível criar outro?
  - Sim
- Podemos fazer o que se segue?

Singleton

```
public class MeuObjeto {

   private MeuObjeto() {}


}
```

- O que significa isso?
- Existe algum código que possa usar este construtor?

Singleton

```
public class MeuObjeto {

   public static MeuObjeto getInstance() {}


}
```

- O que isto significa?

```
MeuObjeto.getInstance();
```

Singleton

- Juntando as 2 coisas

```
public class MeuObjeto {

   private MeuObjeto(){}

   public static MeuObjeto getInstance() {
      return new MeuObjeto();
   }

}
```

Singleton

- Uma outra forma de instanciar um objeto

```
MeuObjeto.getInstance();
```

Exercício - 10min

- Altere a classe abaixo de forma que ela só possa criar uma única instância da classe

```
public class MeuObjeto {

   private MeuObjeto(){}

   public static MeuObjeto getInstance() {
      return new MeuObjeto();
   }

}
```

Singleton
Implementação clássica

```
public class Singleton {
  private static Singleton instanciaUnica;

  private Singleton() {}

  public static Singleton getInstance() {
    if (instanciaUnica == null) {
      instanciaUnica = new Singleton();
    }
    return instanciaUnica;
  }
}
```

Singleton
Implementação clássica

```
public class Singleton {
  private static Singleton instanciaUnica;

  private Singleton() {}

  public static Singleton getInstance() {
    if (instanciaUnica == null) {
      instanciaUnica = new Singleton();
    }
    return instanciaUnica;
  }
}
```

- Uma variável estática para garantir uma única instância para toda a classe
- Um construtor privado que somente a própria classe tem acesso
- O método getInstance instancia o objeto único e o retorna
- Pode haver outros métodos?
  - Sim

+1 Padrão SINGLETON

O **Padrão Singleton** garante que uma classe possua apenas uma **única instância** e fornece um **ponto global** de acesso a ela.

Singleton
Resumindo

- A classe Singleton gerencia sua única instância
  - Nenhuma outra classe pode criar uma nova instância de uma classe Singleton
  - É preciso fazer uso da própria classe para criá-la
- Ponto de acesso global
  - A própria classe

Diagrama de classes

Aplicabilidade

- Quando for necessário existir apenas uma **única instância** de uma classe, e essa instância deve dar **acesso** aos clientes através de **um ponto bem conhecido**

Participantes

- **Singleton**
  - Define uma operação **getInstance()** que permite aos clientes acessarem sua única instância. getInstance é uma **operação de classe,** ou seja **estática**

Colaborações

- Os clientes acessam uma instância Singleton unicamente pela operação getInstance do Singleton

Exemplo

... Veja o código no ambiente de programação!