

CLASSES INTERNAS

# CLASSES INTERNAS

- Uma classe interna é uma classe que é definida dentro de outra classe.
- Pode ser declarada em qualquer nível ou escopo dentro da classe externa.
- Podem ser declaradas publicas, protegidas ou privadas.
- Pode implementar uma ou mais interfaces.
- Sua visibilidade é limitada ao escopo em que se encontra.
- Colabora com a estruturação do código.
- É um recurso da linguagem, mas não da máquina virtual.

# CLASSES INTERNAS

- As classes internas são convertidas pelo compilador em classes normais, sendo o seu nome composto de:
  - `NomedaClasse Externa+$+NomedaClasseInterna`
- A máquina virtual não tem nenhum conhecimento especial a respeito delas.
- Exemplo:

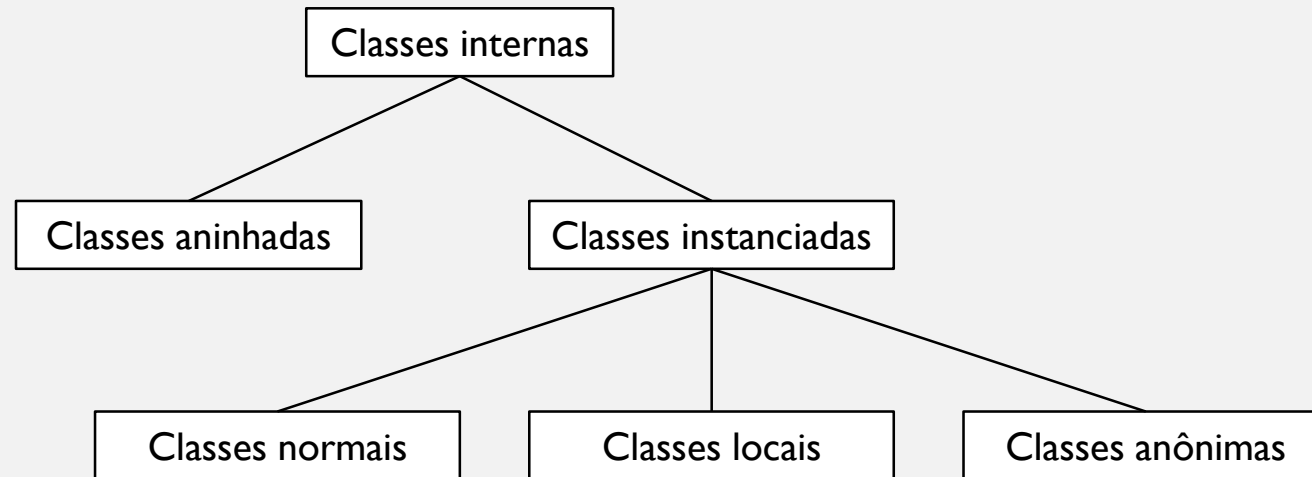
```
class Executivo {  
    class Prefeitura {  
        ...  
    }  
}
```

# CLASSES INTERNAS

- Aplicação
  - Um objeto auxiliar precisa controlar a implementação privada de uma classe, mas sem conceder acesso a outras classes do mesmo pacote.
  - Necessidade de gerar objetos para rotinas auxiliares.
  - Gerar e armazenar objetos auxiliares com seus próprios estados.

# CLASSES INTERNAS

- Tipos de classes internas



# CLASSES INTERNAS

- Tipos de classes internas
  - Aninhadas
    - Tipo mais simples de classe interna.
    - Classe definida dentro de outra, mas funciona como classe de nível superior.
    - Permite definir acesso privado ou protegido e agrupa classes logicamente relacionadas.
    - Definida como se fosse um membro static.
    - Referência via Externa.Interna, como se fosse um atributo estático.

# CLASSES INTERNAS

- Tipos de classes internas
  - Exemplo 1 de classes aninhadas:

```
class Teste {  
    private Um um;  
    private Dois dois;  
    public Teste(Um um, Dois dois) { this.um = um; this.dois = dois;}  
    static class Um {  
        private String nome;  
        public Um(String nome) { this.nome = nome;}  
    }  
    protected static class Dois {  
        private int dois;  
        public Dois(int dois) { this.dois = dois;}  
    }  
}
```

# CLASSES INTERNAS

- Tipos de classes internas
  - Exemplo 1 de classes aninhadas (continuação):

```
public class Principal {  
    public static void main(String[] args) {  
        Teste.Um um = new Teste.Um("Texto Inicial");  
        Teste.Dois dois = new Teste.Dois(5);  
        Teste teste = new Teste(um, dois);  
    }  
}
```



# CLASSES INTERNAS

- Tipos de classes internas
  - Exemplo 2 de classes aninhadas

```
public class exemplo {  
    private static String nome;  
    public void imprime(){  
        System.out.println(nome);  
    }  
    public static class classeI {  
        classeI(){  
            ....//método construtor  
        }  
        public static void calculasalario(){  
            .....//processa as informações  
        }  
    }  
    public static void main(String [] args){  
        exemplo ciI = new exemplo();  
        ciI.imprime();  
        exemplo.classeI.calculasalario();  
    }  
}
```

# CLASSES INTERNAS

- Tipos de classes internas
  - Instanciadas
    - Classe interna normal: definida dentro de outras classes.
    - Classe interna local: definida dentro de um métodos
      - Oculta do mundo exterior.
      - Não admite especificador de visibilidade.
    - Classe interna anônima: definida como valor de retorno de um método.
      - Não possui a definição de um nome para a classe.
      - Não pode ter construtores.
      - Normalmente implementa interface ou classe abstrata.

# CLASSES INTERNAS

- Exemplo de classe interna anônima

```
public class exemplo {  
    protected void IniciaTransacao(){  
        new Thread(){  
            public void run(){ processa(); }  
        }.start();  
    }  
    private void processa(){  
        //processamento de informações...  
    }  
}
```

# CLASSES INTERNAS

- Considerações finais
  - Para acessar atributos e métodos da classe externa é usada a seguinte sintaxe:
    - `nomedaclasse.this.nomedoatributo/nomedometodo()`
  - As classes internas tem acesso aos seus próprios atributos e métodos e os atributos e métodos da classe externa que o declara, isto acontece porque a classe interna sempre
  - Ao compilar uma classe que contém classes internas, são gerados arquivos `.class` para cada uma das classes internas e um arquivo `.class` para a classe externa.
  - Classes internas com nomes têm o nome do arquivo
    - `NomedaClasseExterna$NomedaClasseInterna.class`.
  - Classes internas anônimas têm o nome do arquivo `NomedaClasseExterna$#class.`, onde `#` inicia em `1` e é incrementado para cada classe interna anônima encontrada durante a compilação. obtém uma referência implícita ao objeto da classe que a declarou.

# CLASSES INTERNAS

- Considerações finais
  - As classes internas com nomes de classe podem ser definidas como pública, protegida ou privadas e estão sujeitas às mesmas restrições de uso que os outros membros de uma classe.
  - Para acessar a classe externa, é utilizada a referência: `NomedaClasseExterna.this`.
  - A classe externa é responsável por criar objetos de suas classes internas. Para criar um objeto da classe interna, primeiro deve-se criar um objeto da classe externa e atribuí-lo a uma referência (vamos chamá-lo ref). Em seguida, use uma instrução da seguinte forma para criar um objeto da classe interna:
    - `ClasseExterna.ClasseInterna innerRef = ref.new ClasseInterna();`

# CLASSES INTERNAS

- Considerações finais
  - Uma classe interna pode ser declarada como estática.
  - Uma classe interna estática não exige a definição de um objeto de sua classe externa (ao passo que uma classe não estática exige).
  - Uma classe interna estática não tem acesso aos membros não-estáticos da classe externa
- Estudo complementar:
  - <https://www.devmedia.com.br/classes-anonimas-e-aninhadas-em-java/31167>

## REFERÊNCIAS

- DEITEL, P.J. Java - Como Programar. Porto Alegre: Bookman, 2001.
- NIEMEYER, Patrick. Aprendendo java 2 SDK. Rio de Janeiro: Campus, 2000.
- MORGAN, Michael. Java 2 para Programadores Profissionais. Rio de Janeiro: Ciência Moderna, 2000.
- HORSTMANN, Cay, S. e CORNELL, Gary. Core Java 2. São Paulo: Makron Books, 2001 v.1. e v.2.