

Classes Internas

Classes Internas
Estáticas

Classes Internas de
Instância

Classes Locais

Referências

Classes Internas

prof. Fábio Luiz Usberti

MC322 - Programação Orientada a Objetos

Instituto de Computação - UNICAMP - 2014



Classes Internas

Classes Internas
Estáticas

Classes Internas de
Instância

Classes Locais

Referências

- 1 Classes Internas
- 2 Classes Internas Estáticas
- 3 Classes Internas de Instância
- 4 Classes Locais
- 5 Referências

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Classes dentro de classes

- Em geral, classes são declaradas em um nível superior (**top-level**), o que significa que elas não se encontram declaradas dentro de nenhuma classe.
- As exceções às classes top-level são as classes internas e elas podem ser de quatro tipos:
 - 1 Classes internas estáticas
 - 2 Classes internas de instância
 - 3 Classes locais
 - 4 Classes anônimas (serão revistas após os conceitos de herança e interfaces)

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Motivações

Três razões para o uso de classes internas:

- É uma maneira de **agrupar logicamente** classes que são usadas em um único lugar: se uma classe B é útil somente para outra classe A, então faz sentido mantê-las juntas, embutindo a classe B dentro da classe A.
- Favorece o **encapsulamento**: considere duas classes top-level, A e B, onde B precisa acessar membros de A que seriam, de outro modo, declarados privados. Ao esconder a classe B dentro da classe A, os membros de A podem ser declarados privados enquanto B ainda teria acesso a eles. Além disso, a própria classe B ficaria encapsulada dentro de A.
- O código se torna mais **legível e fácil de manter**: Classes internas dentro de classes top-level posiciona o código mais próximo de onde ele será efetivamente utilizado.

Classes Internas Estáticas

Classes Internas

Classes Internas Estáticas

Classes Internas de Instância

Classes Locais

Referências

Classes pertencentes a classes

- Uma **classe interna estática** é considerada, assim como métodos e atributos estáticos, um membro estático da classe externa que a contém.
- Assim como um método estático é denominado como um método de classe, uma classe interna estática também é conhecida como uma **classe de classe**.
- Uma classe interna estática possui o mesmo comportamento de uma classe top-level, porém com a conveniência de estar encapsulada dentro de outra classe.

```
// classe externa OuterClass
public class OuterClass {
    /* corpo da classe OuterClass */

    // classe interna estática InnerClass
    public static class StaticInnerClass {
        /* corpo da classe InnerClass */
    }
}
```

Classes Internas Estáticas

Classes pertencentes a classes

- Por ser estática, esse tipo de classe interna **não está associada a nenhum objeto** da classe externa.
- A visibilidade entre as classes interna e externa é completa, ou seja, a classe externa tem acesso a todos os membros (estáticos) da classe interna, mesmo os privados, e vice-versa.
- A classe interna estática, por estar no escopo da classe externa, possui **acesso direto** a todos os membros estáticos da classe externa. Acesso direto significa que não é necessário identificar a classe externa.
- Qualquer modificador de acesso (`public`, `protected`, `package-private`, `private`) pode ser utilizado para classes internas estáticas.

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Classes Internas Estáticas

Classes pertencentes a classes

- As classes internas estáticas podem ser acessadas utilizando o nome da classe externa:

```
OuterClass.StaticInnerClass
```

- Para criar um objeto de uma classe interna estática, podemos utilizar a seguinte sintaxe:

```
OuterClass.StaticInnerClass innerObject = new OuterClass.StaticInnerClass();
```

Classes Internas de Instância

Classes pertencentes a objetos

- Uma **classe interna de instância** é considerada um membro não-estático da classe externa que a contém.
- Esse tipo de classe interna é análogo a um método ou atributo de instância, ou seja, essa classe está **associada a um objeto da classe externa**.
- A classe interna de instância tem acesso a todos os atributos e métodos da classe externa, sejam eles privados, estáticos ou não-estáticos.
- Qualquer modificador de acesso (`public`, `protected`, `package-private`, `private`) pode ser utilizado para classes internas de instância.

```
// classe externa OuterClass
public class OuterClass {
    /* corpo da classe OuterClass */

    // classe interna de instância InnerClass
    public class InnerClass {
        /* corpo da classe InnerClass */
    }
}
```


Classes Internas de Instância

Classes pertencentes a objetos

- Classes internas de instância **não podem conter nenhum membro estático**, exceto atributos primitivos constantes (`static final`).
- Para referenciar explicitamente (com a palavra-chave `this`) um atributo de um objeto que contém a classe interna, deve-se utilizar a seguinte sintaxe:

```
<nome classe externa>.this.<nome atributo>
```

- Para instanciar um objeto da classe interna é necessário uma referência do objeto da classe externa (ainda que essa referência seja implícita `this`).
- A sintaxe para instanciar um objeto de classe interna é fornecida a seguir:

```
OuterClass outerObject = new OuterClass(); // instancia um objeto da classe externa  
InnerClass innerObject = outerObject.new InnerClass(); // instancia um objeto da classe interna
```

Classes Internas de Instância

Classes Internas

Classes Internas
Estáticas

Classes Internas de
Instância

Classes Locais

Referências

```
// OuterClass.java
// Classe que exemplifica o sombreamento de variáveis em classes internas
public class OuterClass {

    public int x = 3;

    class InnerClass {

        public int x = 2;

        void methodInner(int x) {
            System.out.println("x = " + x);
            System.out.println("this.x = " + this.x);
            System.out.println("OuterClass.this.x = " + OuterClass.this.x);
        }
    } // fim classe InnerClass

    public static void main(String[] args) {
        OuterClass outerObject = new OuterClass();
        OuterClass.InnerClass innerObject = outerObject.new InnerClass();
        innerObject.methodInner(1);
    } // fim main
}
```

Classes Locais

Classes pertencentes a métodos

- Uma **classe local**, em analogia a variáveis locais, é uma classe definida dentro de um método.
- Apesar das classes locais não serem consideradas membros de uma classe externa, ainda assim são consideradas classes internas.
- Uma classe local tem acesso a todos os atributos e métodos da classe externa (mesmo os privados).
- Uma classe local também tem acesso às variáveis locais e parâmetros do método onde ela está declarada, contanto que tenham o modificador **final**.

```
// classe externa OuterClass
public class OuterClass {
    /* corpo da classe OuterClass */

    // método da classe OuterClass
    public void myMethod() {
        /* corpo do método myMethod */

        // classe local InnerClass
        class InnerClass {
            /* corpo da classe InnerClass */
        }
    }
}
```

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Classes Locais

Classes pertencentes a métodos

- A principal característica de uma classe local é de que seu **escopo se restringe ao método** em que foi declarada, assim como em uma variável local.
- Desse modo, se uma classe interna de instância é utilizada somente dentro de um método da classe externa então essa classe deveria ser declarada de forma local.
- Uma vantagem de classes locais é que elas, em geral, são **declaradas próximas de onde serão efetivamente utilizadas**, o que aumenta a clareza do código.
- Os **modificadores de acesso não são aplicáveis** às classes locais (análogo às variáveis locais).
- Assim como classes internas de instância, as classes locais não podem conter nenhum membro estático.

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Classes Internas

```
// OuterClass.java
// A classe OuterClassA exemplifica o uso de classes internas (estáticas, de instância e locais).
public class OuterClassA {

    private int valA;

    public OuterClassA() {
        valA = 1;
        System.out.println("valA = " + valA);
    }

    // classe interna estática
    public static class InnerStaticClassB {

        private int valB;

        public InnerStaticClassB() {
            valB = 2;
            System.out.println("valB = " + valB);
        }

    } // fim classe InnerStaticClassB

    // classe interna de instância
    public class InnerInstanceClassC {

        private int valC;

        public InnerInstanceClassC() {
            valC = 3;
            System.out.println("valC = " + valC);
        }

    } // fim classe InnerInstanceClassC
    /* continua na próxima página... */
}
```

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Classes Internas

```

/* ... continua da página anterior */
// método que possui uma classe interna local
public void methodA() {
    final int localVal = 3;

    // classe local ao método methodA
    class LocalClassD {
        private int valD;

        public LocalClassD() {
            valD = localVal + valA;
            System.out.println("valD = " + valD);
        }
    } // fim classe LocalClassD

    LocalClassD myObjectD = new LocalClassD();
} // fim método methodA

public static void main(String args[]) {

    // instanciando um objeto da classe top—level OuterClassA
    OuterClassA myObjectA = new OuterClassA();

    // instanciando um objeto da classe interna estática InnerStaticClassB
    OuterClassA.InnerStaticClassB myObjectB = new OuterClassA.InnerStaticClassB();

    // instanciando um objeto da classe interna de instância
    // InnerInstanceClassC
    InnerInstanceClassC myObjectC = myObjectA.new InnerInstanceClassC();

    // chamando o método methodA() do objeto myClassA da classe top—level
    myObjectA.methodA();
} // fim método main
} // fim classe OuterClassA

```

Classes Internas

Classes Internas
EstáticasClasses Internas de
Instância

Classes Locais

Referências

Classes Internas

Exemplo: impressão no terminal

```
valA = 1  
valB = 2  
valC = 3  
valD = 4
```

Classes Internas

Classes Internas

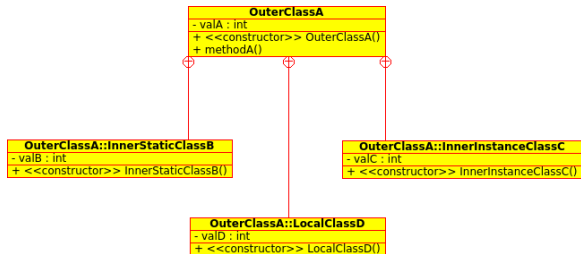
Classes Internas Estáticas

Classes Internas de Instância

Classes Locais

Referências

Diagrama de classes com exemplo de classes internas



Classe interna: classe estática **InnerStaticClassB** interna à classe **OuterClassA**.

Classe interna: classe de instância **InnerInstanceClassC** interna à classe **OuterClassA**.

Classe interna: classe local **LocalClassD** interna à classe **OuterClassA**.

Classes Internas

Classes Internas
Estáticas

Classes Internas de
Instância

Classes Locais

Referências

Referências

- 1 Java: Como Programar, Paul Deitel & Heivey Deitel; Pearson; 7a. Ed. (no. chamada IMECC – 05.133 D368j)
- 2 Data Structures and Algorithms with Object Oriented Design Patterns in Java, Bruno Preiss;
(<http://www.brpreiss.com/books/opus6/>)
- 3 The Java Tutorials (Oracle)
(<http://docs.oracle.com/javase/tutorial/>)
- 4 Guia do Usuário UML, Grady Booch et. al.; Campus(1999)
- 5 Java Pocket Guide - Robert Liguori & Patricia Liguori; O'Reilley, 2008.