Riza Satria Perdana, S.T., M.T.

Teknik Informatika - STEI ITB

Nested Class

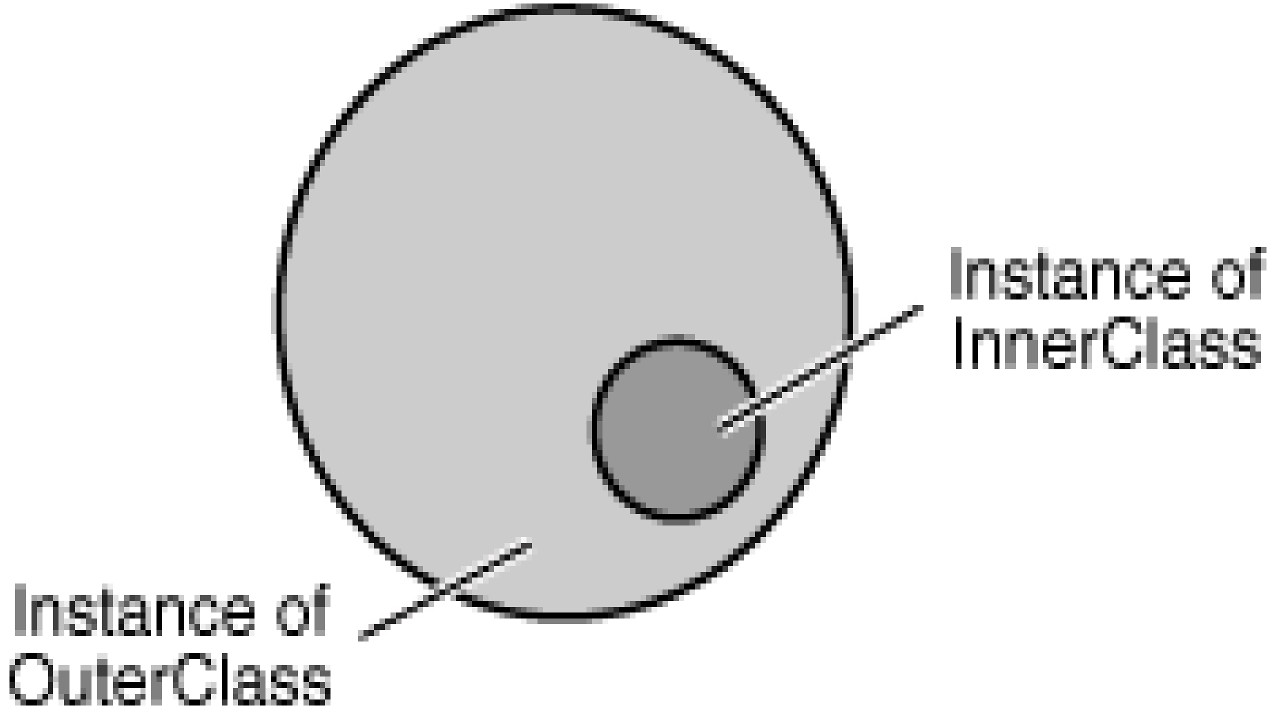
Inner Class

Pemrograman Berorientasi Objek

Inner Class

- Inner class diasosiasikan dengan instans dari outer class-nya
- Inner class mempunyai akses langsung ke atribut atau field outer class-nya
- Tidak boleh memiliki static member
- Instans inner class hanya exist dalam instans outer class-nya

Ilustrasi Inner Class



Instansiasi Nested Class

- Untuk menginstansiasi inner class harus menginstansiasi outer class-nya dulu
- Setelah itu create inner objek dalam outer objek

```
OuterClass.InnerClass innerObject =
outerObject.new InnerClass();
```

Contoh ...

```java
public class DataStructure {
    // Create an array
    private final static int SIZE = 15;
    private int[] arrayOfInts = new int[SIZE];

    public DataStructure() {
        // fill the array with ascending integer values
        for (int i = 0; i < SIZE; i++) {
            arrayOfInts[i] = i;
        }
    }

    public void printEven() {
        // Print out values of even indices of the array
        InnerEvenIterator iterator = this.new InnerEvenIterator();
        while (iterator.hasNext()) {
            System.out.print(iterator.getNext() + " ");
        }
        System.out.println();
    }
```

Contoh ...

```
    // Inner class implements the Iterator pattern,
    private class InnerEvenIterator {
        // Start stepping through the array from the beginning
        private int next = 0;

        public boolean hasNext() {
            // Check if the current element is the last in the array
            return (next <= SIZE - 1);
        }

        public Integer getNext() {
            // Record a value of an even index of the array
            Integer retValue = arrayOfInts[next];
            // Get the next even element
            next += 2;
            return retValue;
        }
    }

    public static void main(String s[]) {
        // Fill the array with integer values and print out only values of even indices
        DataStructure ds = new DataStructure();
        ds.printEven();
    }
}
```

Contoh ...

```java
public class Class1 {
    protected InnerClass1 ic;

    public Class1() {
        ic = new InnerClass1();
    }

    public void displayStrings() {
        System.out.println(ic.getString() + ".");
        System.out.println(ic.getAnotherString() + ".");
    }

    static public void main(String[] args) {
        Class1 c1 = new Class1();
        c1.displayStrings();
    }

    protected class InnerClass1 {
        public String getString() {
            return "InnerClass1: getString invoked";
        }

        public String getAnotherString() {
            return "InnerClass1: getAnotherString invoked";
        }
    }
}
```

Terima Kasih