

Nested Classes

Intro

Problem

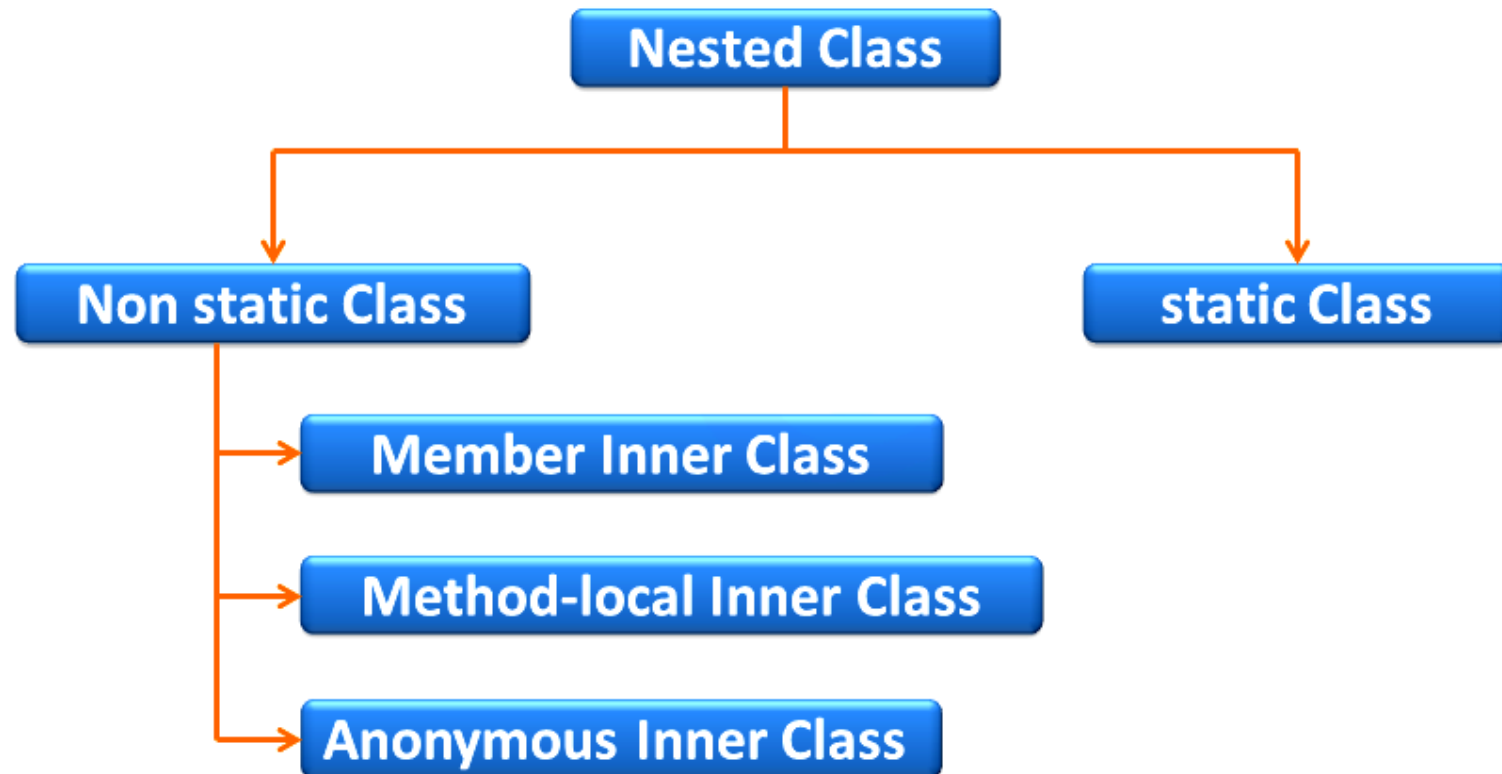
- Каждый раз, когда нам нужна новая функциональность, то создавать объект?
- Каждый раз, когда нам нужен контейнер для данных, то создавать объект?
- Каждый раз, когда нам нужно реализовать интерфейс, то нужно создавать класс?

Solution

- **Nested Class**

Nested Classes

Nested Classes



Nested Classes

- **Внутренним классом** называют **класс**, который является членом другого класса.
- Существует четыре базовых типа внутренних классов в Java:
 - **Static Nested classes or Member of outer class** (статические вложенные классы)
 - **Nested Inner classes** (вложенные внутренние классы)
 - **Method Local Inner classes** (внутренние классы в локальном методе)
 - **Anonymous Inner classes** (анонимные классы)

Static Nested classes

```
class Math {  
    public static class Factorial {  
        private int result;  
        private int key;  
  
        public Factorial(int number, int x) {  
            this.result = number;  
            this.key = x;  
        }  
  
        public int getResult() {  
            return result;  
        }  
  
        public int getKey() {  
            return key;  
        }  
    }  
}  
  
public static Factorial getFactorial(int x) {
```

Static Nested classes

```
public class Program {  
    public static void main(String[] args) {  
        Math.Factorial fact = Math.getFactorial(6);  
        System.out.printf("Факториал числа %d равен %d \n",  
            fact.getKey(), fact.getResult());  
    }  
}
```

Nested Inner classes

```
class Person {  
    private String name;  
    Account account;  
  
    Person(String name, String password) {  
        this.name = name;  
        account = new Account(password);  
    }  
  
    public void displayPerson() {  
        System.out.printf("Person \t Name: %s \t Password: %s  
                           name, account.password);  
    }  
  
    public class Account {  
        private String password;  
  
        Account(String password) {  
            this.password = password;  
        }  
    }  
}
```

Nested Inner classes

```
public class Program {  
    public static void main(String[] args) {  
        Person tom = new Person("Tom", "qwerty");  
        tom.displayPerson();  
        tom.account.displayAccount();  
    }  
}
```

Method Local Inner Classes

```
class Person {  
    private String name;  
  
    Person(String name) {  
        this.name = name;  
    }  
  
    public void setAccount(String password) {  
        class Account {  
            void display() {  
                System.out.printf("Account Login: %s \t Password: %s\n",  
                                   name, password);  
            }  
        }  
        Account account = new Account();  
        account.display();  
    }  
}
```

Method Local Inner Classes

```
public class Program {  
    public static void main(String[] args) {  
        Person tom = new Person("Tom");  
        tom.setAccount("qwerty");  
    }  
}
```

Anonymous Inner classes

Extend a Class:

```
new Book("Design Patterns") {  
    @Override  
    public String description() {  
        return "Famous GoF book.";  
    }  
}
```

Anonymous Inner classes

Implement an Interface:

```
new Runnable() {  
    @Override  
    public void run() {  
        // code  
    }  
}
```