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

Teknik Informatika - STEI ITB

Design Pattern

Pemrograman Berorientasi Objek

Design Pattern

Design patterns represent the best practices used by experienced object-oriented software developers.

Design patterns are solutions to general problems that software developers faced during software development.

These solutions were obtained by trial and error by numerous software developers over quite a substantial period of time.

Usage of Design Pattern

Common platform for developers

- Design patterns provide a standard terminology and are specific to particular scenario

Best Practices

- Design patterns have been evolved over a long period of time and they provide best solutions to certain problems faced during software development

Types of Design Patterns

Creational Patterns

- provide a way to create objects while hiding the creation logic

Structural Patterns

- concern class and object composition

Behavioral Patterns

- specifically concerned with communication between objects

J2EE Patterns

- specifically concerned with the presentation tier

Singleton Pattern

Type: creational pattern

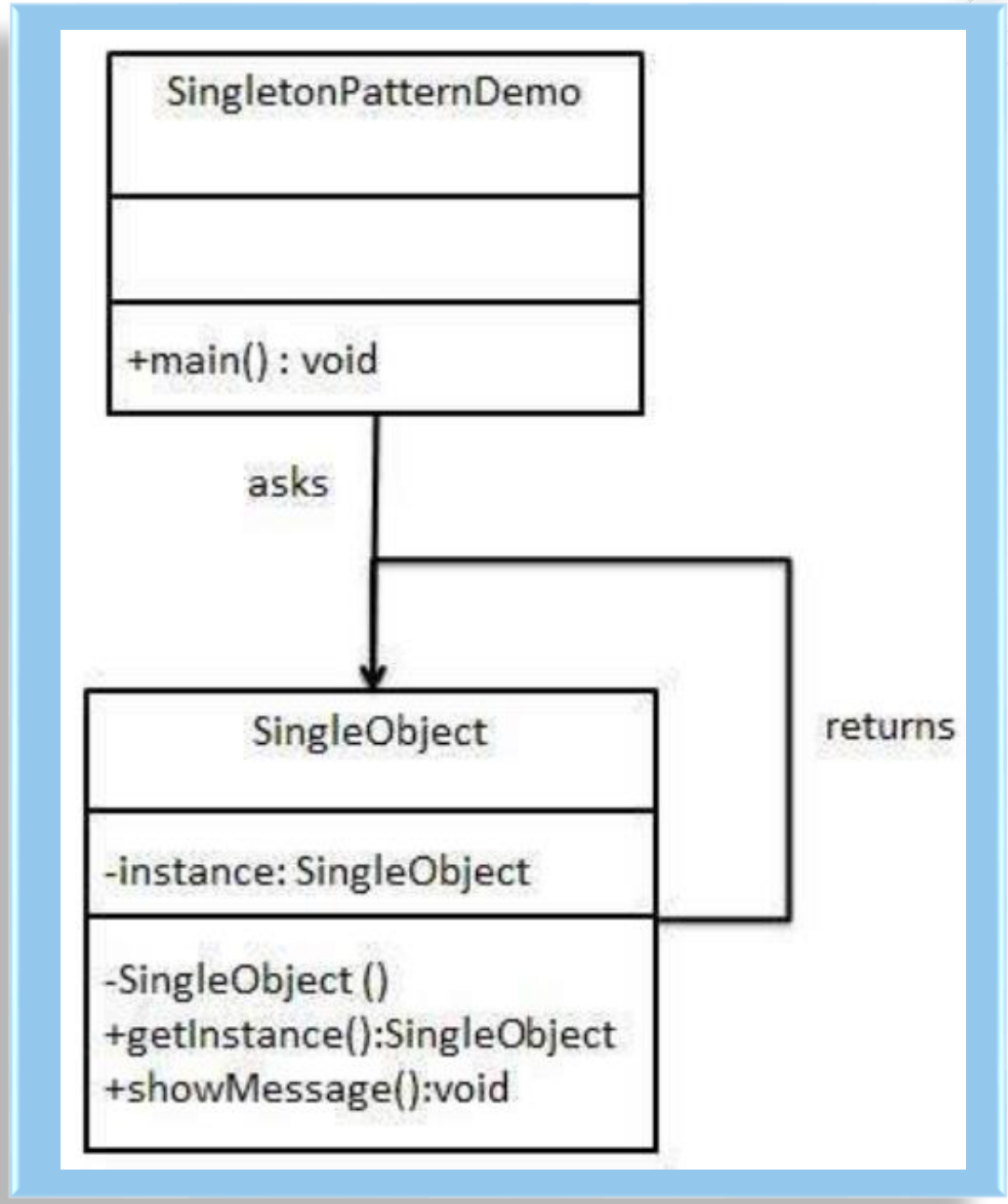
- provides one of the best ways to create an object.
- a single class which is responsible to create an object while making sure that only single object gets created.
- provides a way to access object without need to instantiate the object of the class

Implementation

Implementation

- SingleObject class have its constructor as private and have a static instance of itself
- SingleObject class provides a static method to get its static instance to outside world

Class Diagram



SingleObject.java

```
public class SingleObject {

   //create an object of SingleObject
   private static SingleObject instance = new SingleObject();

   //make the constructor private so that this class cannot be
   //instantiated
   private SingleObject(){}

   //Get the only object available
   public static SingleObject getInstance(){
      return instance;
   }

   public void showMessage(){
      System.out.println("Hello World!");
   }
}
```

SingletonPatternDemo.java

```
public class SingletonPatternDemo {
   public static void main(String[] args) {

      //illegal construct
      //Compile Time Error: The constructor SingleObject() is not visible
      //SingleObject object = new SingleObject();

      //Get the only object available
      SingleObject object = SingleObject.getInstance();

      //show the message
      object.showMessage();
   }
}
```

Terima Kasih