Object Oriented Programming

Lecture No. 9

Ing. Lukáš Slánský

Univerzita Pardubice

5. 1. 2009

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Contents

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Design Patterns

Design pattern is a general repeatable solution to a comonly occuring problem in software design.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Design Patterns

Design pattern is a general repeatable solution to a comonly occuring problem in software design.

- It's not a finished design. It is a description for how to solve a problem.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Design Patterns

Design pattern is a general repeatable solution to a comonly occuring problem in software design.

- It's not a finished design. It is a description for how to solve a problem.
- It is like algorithm (that solves computational problem) – but solves *design problem.*

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Design Patterns Classification

Design Patterns are divided into many types:

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Design Patterns Classification

Design Patterns are divided into many types:

- Creational Patterns – deal with object creation mechanisms.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Design Patterns Classification

Design Patterns are divided into many types:

- Creational Patterns – deal with object creation mechanisms.
- Structural Patterns – ease the design by identifying a simple way to realize relations between entities.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Design Patterns Classification

Design Patterns are divided into many types:

- Creational Patterns – deal with object creation mechanisms.
- Structural Patterns – ease the design by identifying a simple way to realize relations between entities.
- Behavioral Patterns – identify common communication patterns between objects.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Factory Method Pattern

Factory Method Pattern

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Factory Method Pattern

Factory Method Pattern is a Creational Pattern that deals with the problem of creating object without specifying the exact class of object that will be created.

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Factory Method Pattern

Factory Method Pattern is a Creational Pattern that deals with the problem of creating object without specifying the exact class of object that will be created.

Factory method is a static method that returns new implemented object.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Factory Method Pattern

Factory Method Pattern is a Creational Pattern that deals with the problem of creating object without specifying the exact class of object that will be created.
Factory method is a static method that returns new implemented object.

```
class Complex {
  public static Complex fromCartesian(double re, double im){
    return new Complex(re, im);
  }
  public static Complex fromPolar(double mod, double ang) {
    return new Complex(mod*cos(ang), mod*sin(ang));
  }
  private Complex(double re, double im) {
    ...
  }
```

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Factory Method Pattern

Factory Method Pattern is a Creational Pattern that deals with the problem of creating object without specifying the exact class of object that will be created.
Factory method is a static method that returns new implemented object.

```
class Complex {
  public static Complex fromCartesian(double re, double im){
    return new Complex(re, im);
  }
  public static Complex fromPolar(double mod, double ang) {
    return new Complex(mod*cos(ang), mod*sin(ang));
  }
  private Complex(double re, double im) {
    ...
  }
  Complex c=Complex.fromPolar(1, pi);
}
```

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Singleton Pattern

Singleton Pattern

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

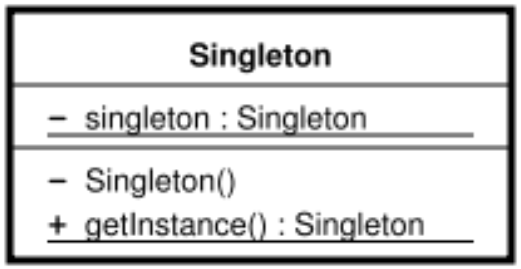
Singleton Pattern

Singleton Pattern is a Creational Pattern thet is used to restrict instatntion of a class to *one* object.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

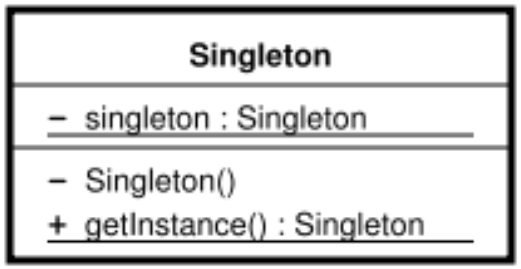
Singleton Pattern

Singleton Pattern is a Creational Pattern thet is used to restrict instatntion of a class to *one* object.



Singleton Pattern

Singleton Pattern is a Creational Pattern thet is used to restrict instatntion of a class to *one* object.




```
class Singleton {
  private Singleton() {} //private constructor!!!
  private final static Singleton INSTANCE;
  public static Singleton getInstance() {
    if (INSTANCE==null) {
      INSTANCE=new Singleton();
    }
    return INSTANCE;
  }
}
```

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Adapter Pattern

Adapter Pattern

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Adapter Pattern

Adapter Pattern is a Structural Pattern that 'adapts' one interface for a class int one that client expects.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Adapter Pattern

Adapter Pattern is a Structural Pattern that 'adapts' one interface for a class int one that client expects.

```
class DList<T> { //Double-Linked List
  public void insertHead(T o) { ... }
  public void insertTail(T o) { ... }
  public T removeHead() { ... }
  public T removeTail() { ... }
  public int getNumItems() { ... }
  ... constructors and other methods ...
}
```

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Adapter Pattern

Adapter Pattern is a Structural Pattern that 'adapts' one interface for a class int one that client expects.

```
class DList<T> { //Double-Linked List
  public void insertHead(T o) { ... }
  public void insertTail(T o) { ... }
  public T removeHead() { ... }
  public T removeTail() { ... }
  public int getNumItems() { ... }
  ... constructors and other methods ...
}

interface Stack<T> {
  void push(T o);
  T pop();
  bool isEmpty();
}
```

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

```
//Double-Linked List implemented by Stack
class DListImpStack<T> implements Stack<T> {
  private DList<T> list=new DList<T>();
  public void push(T o) {
    list.insertTail(o);
  }
  public T pop() {
    return list.removeTail();
  }
  public bool isEmpty() {
    return list.getNumItems()==0;
  }
}
```

Observer Pattern

Observer Pattern

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

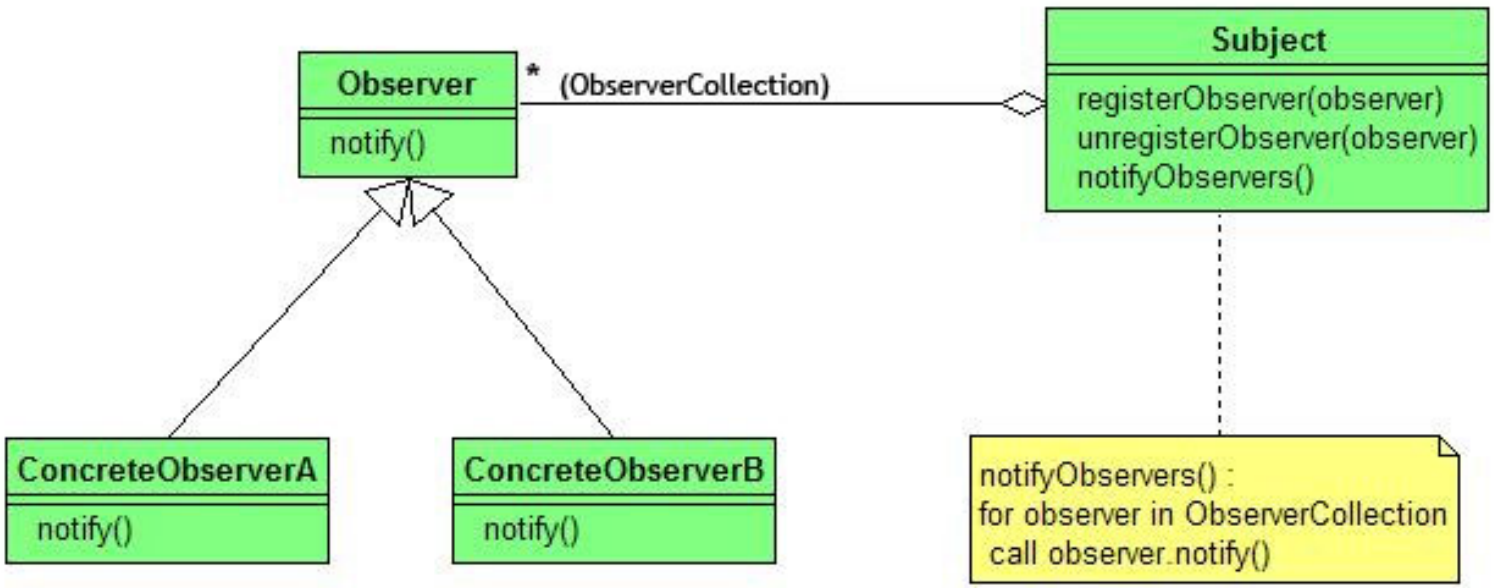
Observer Pattern

Observer Pattern is a Behavioral Pattern that is used to observe the state of an object.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Observer Pattern

Observer Pattern is a Behavioral Pattern that is used to observe the state of an object.



Further Reading on Design Patterns

- `http://en.wikipedia.org/wiki/Design_pattern_(computer_science)`
- `http://objekty.vse.cz/Objekty/Vzory`
- Fowler, Martin. *Patterns of Enterprise Application Architecture*.
- Gamma, Erich. *Design Patterns: Elements of Reusable Object-Oriented Software*.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Distributed Object Computing

- Programs can be written as distributed. Some offer services to others.

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Distributed Object Computing

- Programs can be written as distributed. Some offer services to others.
- Services can be distributed over network.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Distributed Object Computing

- Programs can be written as distributed. Some offer services to others.
- Services can be distributed over network.
- Services and programs can be written in different programming languages.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Common things

- IDL – Interface Definition Language
  - Definition of classes that will be exported by service provider.
  - It's language independent.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Common things

- IDL – Interface Definition Language
  - Definition of classes that will be exported by service provider.
  - It's language independent.
- IDL file is compiled into target language skeleton file.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Common things

- IDL – Interface Definition Language
  - Definition of classes that will be exported by service provider.
  - It's language independent.
- IDL file is compiled into target language skeleton file.
- Service connects to IIOP (Internet Inter-ORB Protocol) server or something similar (ORPC, JRMP).
- Client connects to server with request and server forwards this request to service provider.
- Response and exceptions goes through server as well.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Further Reading on Distributed Object Computing

- http://my.execpc.com/~gopalan/misc/compare.html
- http://en.wikipedia.org/wiki/CORBA
- http://en.wikipedia.org/wiki/Distributed_Component_Object_Model
- http://en.wikipedia.org/wiki/Java_remote_method_invocation

Univerzita Pardubice
Fakulta elektrotechniky a informatiky