Software Design Patterns

Design Patterns based on the GoF

Boniface Kabaso

March 12, 2016

Outline

Outline

Outline
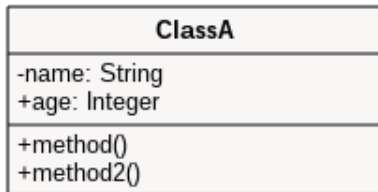
Software Design Patterns
Unified Modelling Language (UML)

1 Class

2 Association

3 Dependency

4 Aggregation

5 Composition

6 Inheritance-Generalisation

7 Inheritance-Realisation

Outline

Unified Modelling Language (UML
Class Definition

ClassA

-name: String
+age: Integer

+method()
+method2()

Code

```
1   public class ClassA {
2       private String name;
3       public Integer age;
4       public void method(){}
5       public void method2(){}
6   }
```

Outline

Class Relationships
Bi-direction Association

ClassA				ClassB
	1		1	

Code

```
1   public class ClassA {
2       private ClassB b;
3   }
```

Code

```
1   public class ClassB {
2       private ClassA a;
3   }
```

Class Relationships
Bi-direction Association One to Many

ClassA		ClassB

1   0..*

Code

```
1  public class ClassA {
2      private List<ClassB> b;
3  }
```

Code

```
1  public class ClassB {
2      private ClassA a;
3  }
```

Class Relationships
Bi-direction Association One to Many

ClassA		ClassB

1   *

Code

```
1  public class ClassA {
2      private List<ClassB> b; // > 1 b
3  }
```

Code

```
1  public class ClassB {
2      private ClassA a;
3  }
```

Class Relationships
Uni-direction Association One to Many

ClassA

1   *

ClassB

Code

```
1   public class ClassA {
2        private List<ClassB> b; //  > 1 b
3   }
```

Code

```
1   public class ClassB {
2
3   }
```

Class Relationships
Dependency

Code

```
1   public class ClassA {
2   }
```

Code

```
1   public class ClassB {
2       public void methodInB(ClassA a) {
3       }
4   }
```

Class Relationships
Aggregation

Code

```
1  public class ClassA {
2    private ClassB b;
3    public void methodInA(ClassB value) {
4     b=value;
5    }
6  }
```

Class Relationships
composition

Code

```
1   public class ClassA {
2     private ClassB b = new ClassB();
3   }
```

NB: Constructor Initialisation of a declared variable is also composition

Class Relationships
Inheritance-Generalisation

Code

```
1   public class ClassA {
2   }
```

Code

```
1   public class ClassB   extends ClassA{
2
3   }
```

Class Relationships
Inheritance-Realisation

Code

```
1   public class ClassA implements ClassB{
2   }
```

Code

```
1   public interface ClassB{
2
3   }
```

Questions ?

Boniface Kabaso

kabasoB AT cput DOT ac DOT za