

ASSINGMENT NO.	12
TITLE	Template design pattern and exception handling in Java
PROBLEM STATEMENT /DEFINITION	<p>Write a program on template and exception handling in Java: in this assignment multiple templates are to be designed as a pattern and these patterns to be used to take decisions</p> <p>Example:</p> <p>Write a program for implementation of stack ADT in java, Stack is abstract base class with template method consists of push and display. IntergerStack and CharStack are two concrete classes inherited from Stack which give specific implementation of these methods. Also handle stack full and stack empty condition as exceptions</p>
OBJECTIVE	<ul style="list-style-type: none"> • To understand and use the concept of template design pattern • To understand concept and importance of exception handling in java • To learn to use multiple templates as pattern to take decisions
OUTCOME	<ul style="list-style-type: none"> • To be able to implement multiple templates in java • To be able to implement exception handling in java
S/W PACKAGES AND HARDWARE APPARATUS USED	<ul style="list-style-type: none"> • Operating Systems (64-Bit)64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version • Programming Tools (64-Bit) Latest Open source update of Eclipse Programming frame work, GTK+ • Programming language Java. • Editors like gedit, vi editor, etc.
REFERENCES	<ul style="list-style-type: none"> • The complete reference Java seventh edition by Herbert Schildt • Java Programming: 24-Hour Trainer Second Edition by Yakov Fain
STEPS	Refer to steps below
INSTRUCTIONS FOR WRITING JOURNAL	<ol style="list-style-type: none"> 1. Date 2. Assignment no.

	3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis
--	---

Prerequisites:

- Knowledge of object oriented concepts.
- Basic knowledge of java programming.

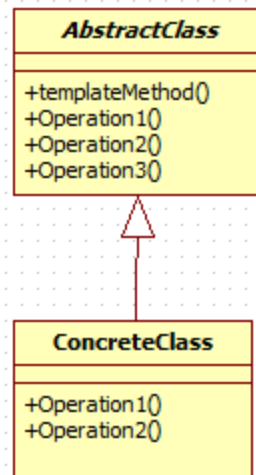
Concepts related Theory:

Template Method Design Pattern:

Design patterns are the best practices used by experienced object-oriented software developers. These design patterns are solutions to general problems that software developers faced during software development. There are total 23 design patterns. All these 23 design patterns are classified into Creational, Structural and Behavioral patterns. Among this behavioral patterns are specifically concerned with communication between objects.

Template method pattern is a behavioral design pattern which provides base method for algorithm, called template method which defers some of its steps to subclasses. So algorithm structure is same but some of its steps can be redefined by subclasses according to context.

Template means preset format like HTML templates which has fixed preset format. Similarly in template method pattern, a preset structure method called template method which consists of steps. These steps can be abstract method which will be implemented by its subclasses. In template method pattern, there is template method which defines set of steps and implementation of steps can be deferred to subclasses. Thus template method defines algorithm but exact steps can be defined in subclasses. We use template when you have a preset format or steps for algorithm but implementation of steps may vary or when you want to avoid code duplication, implementing common code in base class and variation in subclass.



So in above diagram, you can see we have defined template method with three steps i.e. operation1, operation2 and operation3 means,

```
public final void templateMethod(){
    operation1();
    operation2();
    operation3();
}
```

Suppose among them, operation1 and operation2 are abstract steps, so these are implemented by ConcreteClass. As operation3 is common for all so it's defined in base class, so default implementation of that method is in abstract base class.

Components:

Abstract Class: It defines template method defining the structure of algorithm and it also defines abstract operations that will be implemented by subclasses to define steps of algorithm.

Concrete Class: It implements abstract operation of super class to carry out subclass specific steps of the algorithm and also overrides operation if default behavior is not required.

Exception Handling in java:

- An *exception* is an error condition that changes the normal flow of control in a program
- When something unexpected occurs
 - Ensure program detects the problem

- Then program must do something about it
- Exceptions are runtime errors
- Exception handling gives us another opportunity to recover from the abnormality.
- It separates business logic and error handling code

In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime. The exception handling in java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained. Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IO, SQL, Remote etc. The core advantage of exception handling is to maintain the normal flow of the application.

An exception (or exceptional event) is a problem that arises during the execution of a program. When an **Exception** occurs the normal flow of the program is disrupted and the program/ Application terminate abnormally, which is not recommended, therefore, these exceptions are to be handled.

An exception can occur for many different reasons. Following are some scenarios where an exception occurs.

- A user has entered an invalid data.
- A file that needs to be opened cannot be found.
- A network connection has been lost in the middle of communications or the JVM has run out of memory.

There are mainly three types of exceptions:

1. Checked Exception

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.

2. Unchecked Exception

The classes that extend RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time rather they are checked at runtime.

3. Error

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

There are 5 keywords used in java exception handling.

1. try
2. catch

3. finally
4. throw
5. throws

Algorithm:

Template Method design pattern

Step 1: Define abstract class with template method consists of abstract methods and common methods

Step 2: Common implementations of individual steps are defined in the base class

Step 3: Override or implement specific steps to sub class

Step 4: Template method in super class should not be overridden so make it final

Example:

```
abstract class Generalization {
    public final void findSolution() {
        stepOne();
        stepTwo();
        stepThr();
        stepFor();
    }
    protected void stepOne() { //Generalization stepOne }
    abstract protected void stepTwo();
    abstract protected void stepThr();
    protected void stepFor() { //Generalization stepFor }
}

class Specialization extends Generalization {
    protected void stepThr() {
        // Specific implementation of method for class Specialization
    }

    protected void stepTwo() { // Specific implementation of method for class
        Specialization}
}

class Specialization1 extends Generalization {
    protected void stepThr() {
        // Specific implementation of method for class Specialization1
    }

    protected void stepTwo() { // Specific implementation of method for class
        Specialization1 }
}

class TemplateMethodDemo {
    public static void main( String[] args ) {
        Generalization algorithm = new Specialization ();
        algorithm.findSolution();
        algorithm=new Specialization1()
        algorithm.findSolution();
    } }
```

Write a program for implementation of stack ADT in java , Stack is abstract base class with template method consists of push and display. IntergerStack and CharStack are two concrete classes inherited from Stack which give specific implementation of these methods. Also handle stack full and stack empty condition as exceptions

Conclusion:

Student will able unable to understand and implement template design pattern and exception handling in Java.

Review Questions:

1. What is the exception handling in java?
2. What is mean by template design pattern in java?
3. What are the types of exception?
4. What are the components of template?
5. Which are the keywords used in java exception handling?
6. What is use of design patterns?
7. What is template method?