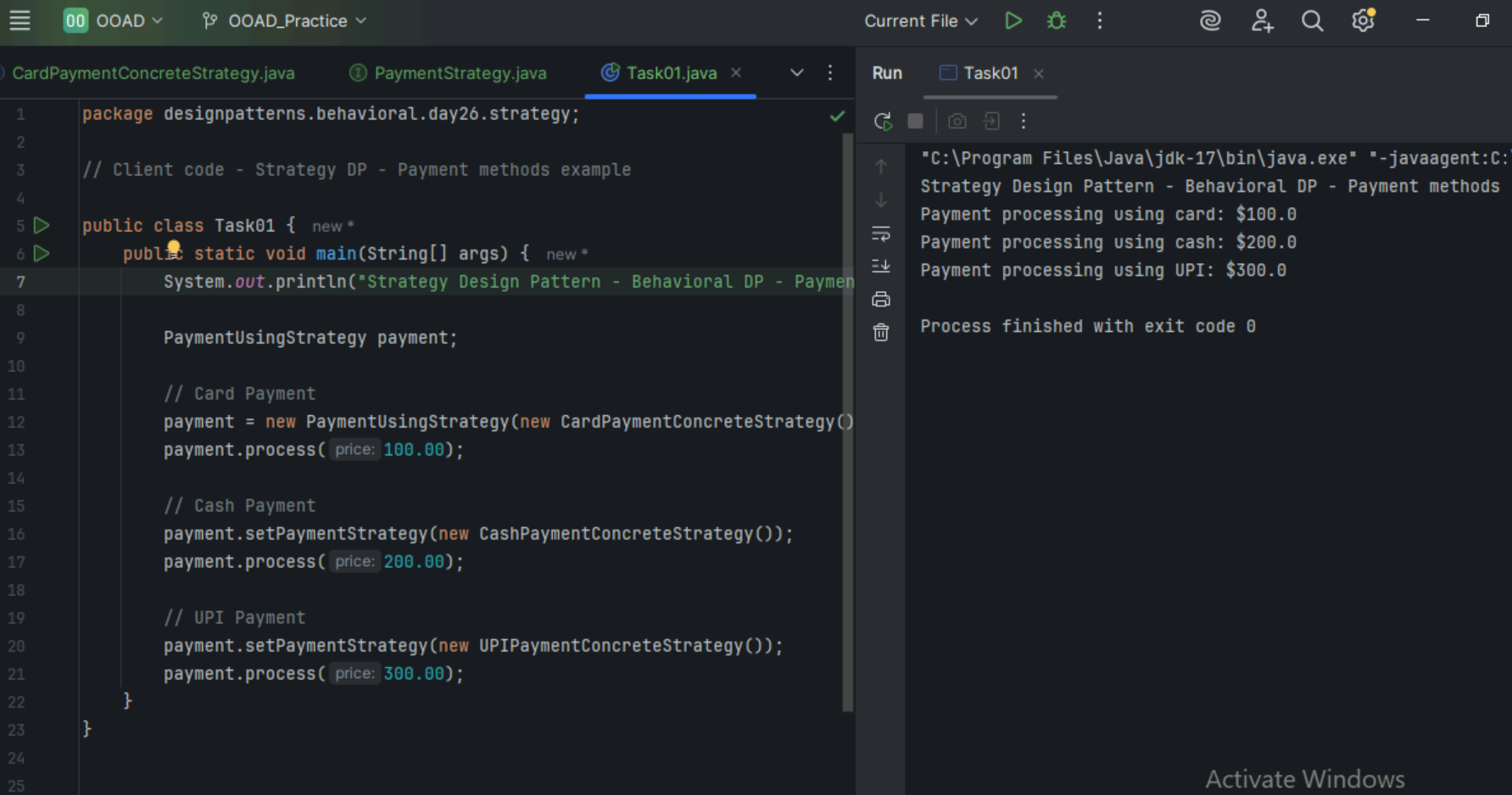
Day 26 - 104608492 - Shirisha Perapagu

Design Patterns

* Behavioral Design Patterns
  1. Strategy
  2. Pub-Sub

Task 01

Strategy Design Pattern – Behavioral DP – Payment methods example



Task 02

Design Workflow

Design Workflow is about understanding the problem, plan ideas, create the designs, get feedback and improve until its done.

Example:

Designing a website for bakery

1. **Understanding the problem** - know what the client need.

So here client wants a bakery website to show the menu, location and order online options.

1. **Plan ideas** – what pages or features it requires

So here we can plan or have a rough sketch of website like Home Page, Menu Page, Order Online etc.

1. **Create designs** – design the website

Design webpages by using UI tools use colors, photos, buttons etc.

1. **Get feedback** – show the design to client and ask for feedback

if client wants some changes in the page do accordingly.

1. **Improve –** make changes and finalize the design and handover to developer

Task 03

Persistent objects are objects that stay in memory even after a program stops running so we can use them later.

Example

1. A user profile saved in database
2. Game level is saved even if we close the game.
3. A shopping cart saved even after we close the browser.

Task 04:

Which of the following components is not typically part of the Command pattern?

a) Invoker

b) Receiver

c) Abstract Factory

d) Command (interface/abstract class)

Task 05:

What role does the Invoker play in the Command pattern?

a) It knows how to perform the operations associated with a request.

b) It encapsulates the request as an object.

c) It asks the command to carry out the request.

d) It defines the interface for executing an operation.

Task 06:

A key benefit of using the Command pattern is its ability to support:

a) Lazy initialization

b) Undo/Redo functionality

c) Singleton instance creation

d) Compile-time polymorphism

Task 07:

In the Strategy pattern, what role does the "Context" play?

A. It defines the interface for the algorithms.

B. It implements a specific algorithm.

C. It maintains a reference to a Strategy object and delegates the task to it.

D. It creates the Concrete Strategy objects.

Task 8:

1. In which of the following mechanisms, types of all variables and expressions are fixed at compilation time.

a) Strong Typing

b) Weak Typing

c) Static Binding/ early binding

d) Dynamic Binding/ late binding

Task 9:

In which pattern does a class represent the functionality of another class, providing a simplified interface to a complex subsystem?

a) Decorator Pattern

b) Facade Pattern

c) Proxy Pattern

d) Composite Pattern

Task 10:

 Which of the following statements about Persistence is correct?

a) It is the enforcement of the class of an object, such that objects of different types may not be interchanged, or at the most they may be interchanged only in very restricted ways.

b) It is the property of an object through which its existence transcends time and/or space.

c) It is the property that distinguishes an active object from one that is not active.

d) All of the mentioned

Task 11:

 What is that concept in type theory in which a single name may denote objects of many different classes that are related by some common super class referred to \_\_\_\_\_\_

a) Monomorphism

b) Type Checking

c) Polymorphism

d) Generalization

Task 12:

Which of the following patterns is used to create a single instance of a class and provide a global point of access to it?

a) Factory Pattern

b) Singleton Pattern

c) Builder Pattern

d) Prototype Pattern

Task 13:

 The Adapter pattern is a type of \_\_\_\_\_\_ pattern.

a) Creational

b) Structural

c) Behavioral

d) Concurrency

Task 14:

Which design pattern defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically?

a) Strategy Pattern

b) Command Pattern

c) Observer Pattern

d) Mediator Pattern

Task 15:

The Model-View-Controller (MVC) is an example of a \_\_\_\_\_\_ pattern.

a) Creational

b) Structural

c) Behavioral

d) Architectural

Task 16:

Clas name DManager

Add , remove , retrieve..  Methods (list)

→ Declare private static variable ..- type Dmanager - make sure it holds singleton instance

→ create private constructor to prevent direct instantiation of class

→ create instance..

→ throw an illegalStaticException , create **singleton** instance if not created .. use getInstance()

→ public static synchronized method – getInstance()

 → return existing instance..

→ create a new instance if doesnot exist..

In the thread safe list management:

→ declare a private variable

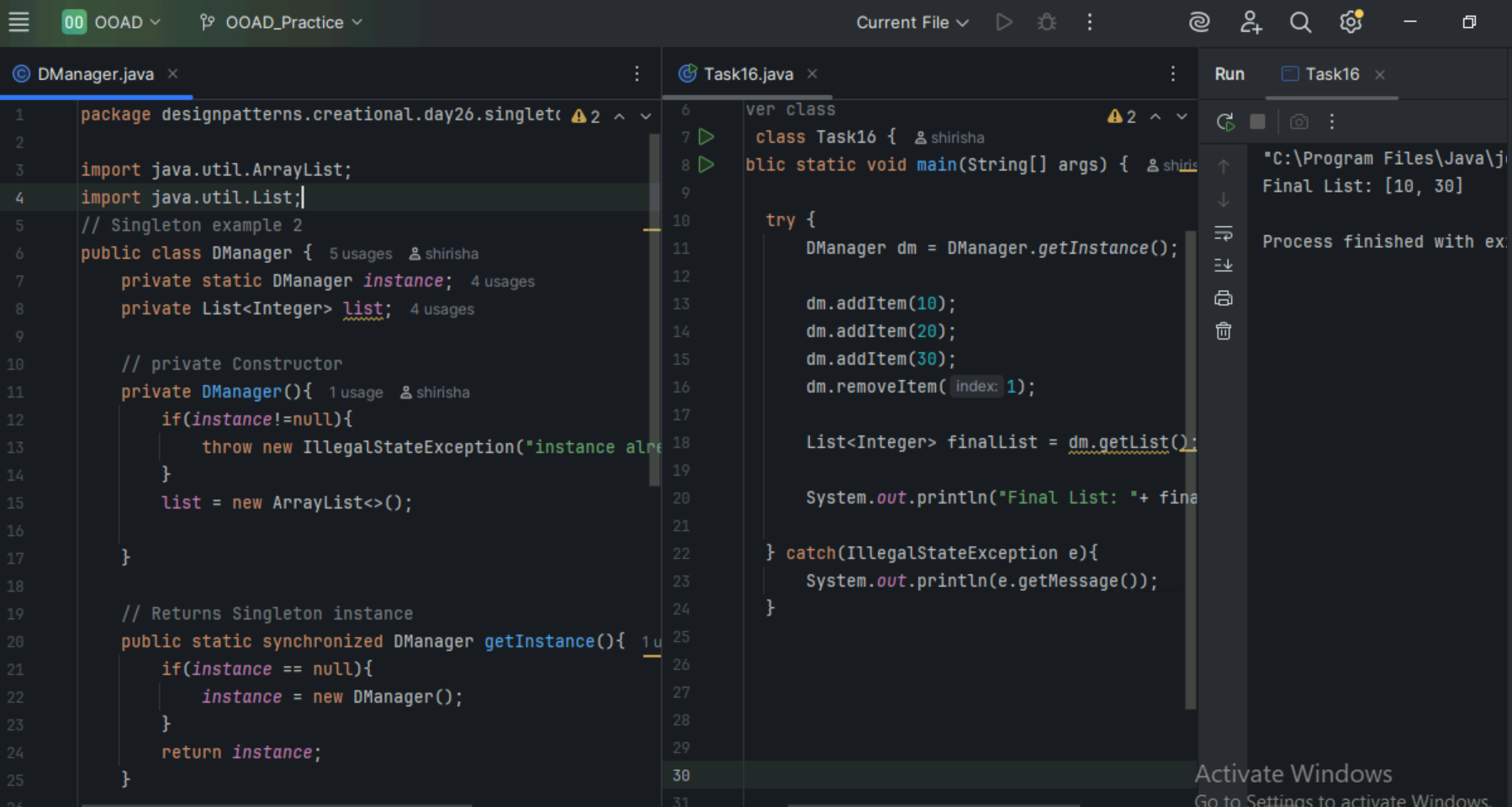
→ initialize the list inside the constructor using new..

→ also implement public sync methods

→ addItem

→ removeitem

→ list



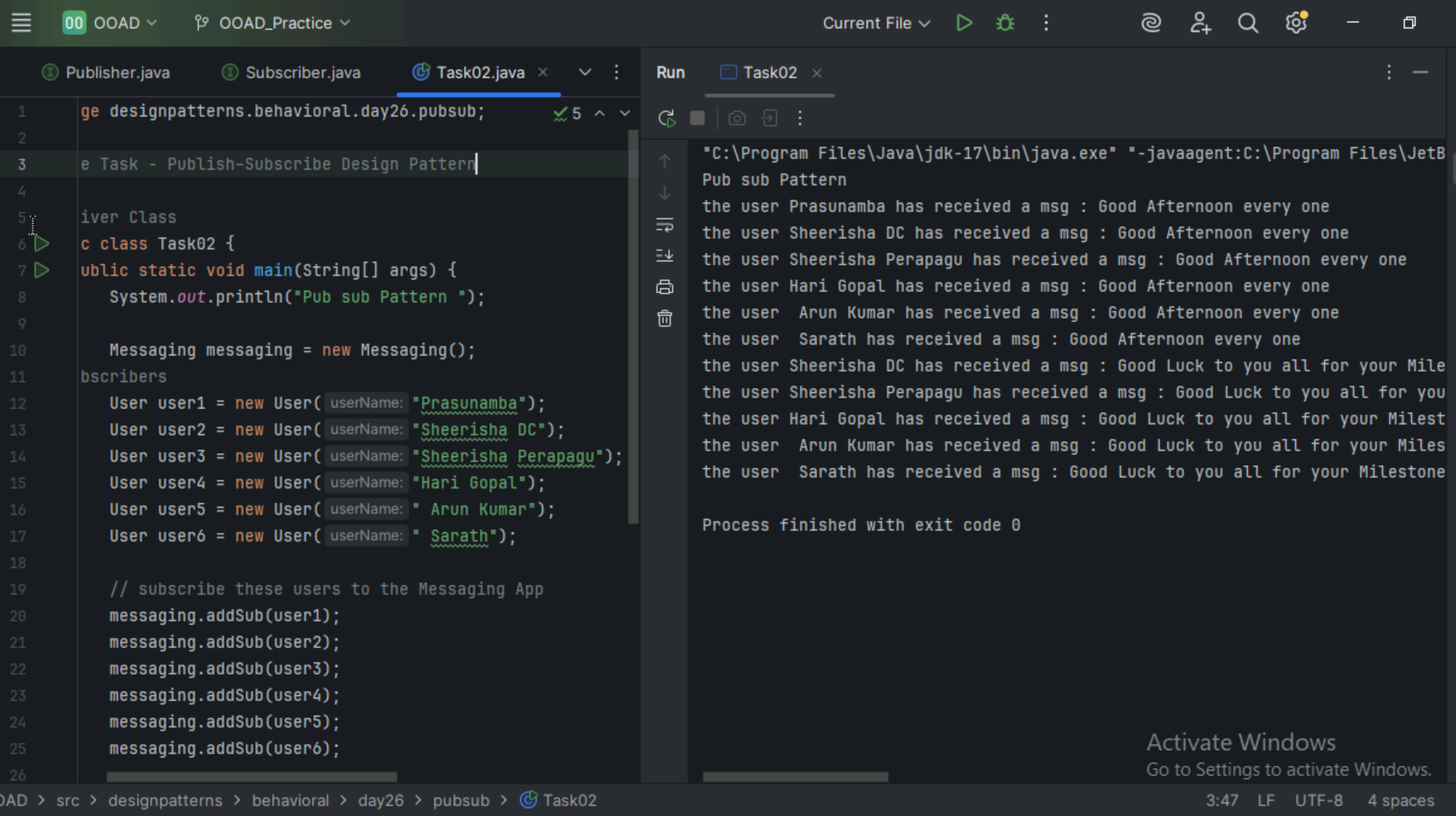
===================================================================

Home Tasks

===================================================================

Task 17

Pub-Sub code



Task 18

Strategy DP – Travel modes example

