ITP4507 Contemporary Topics in Software Engineering

Assignment Report

|  |  |
| --- | --- |
| Name: | Tse Man Fung |
| Student ID: | 230026964 |
| Class: | SE-2A |

# Table of Contents

[Table of Contents 2](#_Toc182171023)

[Assumption 3](#_Toc182171024)

[Java version 3](#_Toc182171025)

[Assumption in design 3](#_Toc182171026)

[Class diagram 5](#_Toc182171027)

[Design pattern 6](#_Toc182171028)

[Factory Pattern 6](#_Toc182171029)

[Command Pattern 8](#_Toc182171030)

[Memento Pattern 9](#_Toc182171031)

[Adapter Pattern 10](#_Toc182171032)

[Test Plan and Test Cases 11](#_Toc182171033)

[Test Case 1 – simple test 11](#_Toc182171034)

[Test Case 2 – call hero skill in redo 30](#_Toc182171035)

[Test Case 3 – Error test 37](#_Toc182171036)

[Source Code 40](#_Toc182171037)

[Run Batch 40](#_Toc182171038)

[Main 40](#_Toc182171039)

[package FantasticWorld 43](#_Toc182171040)

# Assumption

## Java version

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

## Assumption in design

In program design, the following assumptions are made:

1. The Hero class will be used solely for inheritance purposes, and specific hero classes should be newly created. Any newly created specific hero class should inherit from the Hero class to inherit its methods while also implementing the Hero interface to support new methods, such as callSkill().
2. The Factory Pattern is used to create instances of Player, Hero, and all Command classes.
3. The Memento Pattern is employed in ChangePlayerNameCommand and CallHeroSkillCommand, with these two commands acting as caretakers, instructing the mementos when to call save() and restore().
4. The RefCurrentPlayerAdapter interface and its concrete class RefCurrentPlayerAdapter have been introduced. This class is responsible for managing the current player's settings and checks. It also handles displaying the current player's information in the user interface (e.g., "The current player is...").
5. When the caller is a class that creates a new player and executes a redo operation, only a null value should be passed to the setCurrentPlayer() method in RefCurrentPlayerAdapter. Passing a null value will trigger the if statement in the method to set the current player to the last created player, since the current player has been removed by the undo operation.
6. All command factories are stored in a HashMap within the Main class and are invoked using the corresponding option strings (e.g., c = create player) based on user input.
7. The Scanner in the Main class is set as static to receive user input across all classes.

# Class diagram

You can download the file from [Class diagram](https://vtcmca-my.sharepoint.com/:f:/g/personal/230026964_stu_vtc_edu_hk/Eg2Wrejz6H9Injh1O8-dDdwBNj_y1zD6BPqahOIjPUtlQA?e=lVP4Hr)

一張含有 圖表, 文字, 方案, 寫生 的圖片

自動產生的描述

# Design pattern

## Factory Pattern

In this case, the Factory Pattern is used to create a family of related objects without specifying their concrete classes. Two factory interfaces are defined: Factory and CommandFactory. Concrete factories implement these interfaces to initialize specific classes. Class instances are declared as abstract instance types of these classes and returned to the caller. Here are the details of the related classes:

1. **Abstract Factories**: Factory, CommandFactory
2. **Concrete Factories**: PlayerFactory, HeroFactory, AddHeroCommandFactory, CallHeroSkillCommandFactory, ChangePlayerNameCommandFactory, CreatePlayerCommandFactory, DeleteHeroCommandFactory, DisplayAllPlayersCommandFactory, ExitCommandFactory, ListUndoRedoCommandFactory, RedoCommandFactory, SetCurrentPlayerCommandFactory, ShowPlayerDetailCommandFactory, UndoCommandFactory
3. **Abstract Product**: Command
4. **Concrete Products**: AddHeroCommand, CallHeroSkillCommand, ChangePlayerNameCommand, CreatePlayerCommand, DeleteHeroCommand, DisplayAllPlayersCommand, ExitCommand, ListUndoRedoCommand, RedoCommand, SetCurrentPlayerCommand, ShowPlayerDetailCommand, UndoCommand

In the Main class, upon receiving user input, the corresponding command factory is retrieved from a HashMap. The create() method is then called to declare an abstract instance type object of Command, initializing a concrete command instance returned to the caller. PlayerFactory and HeroFactory are called by AddHeroCommand and CreatePlayerCommand to create and return Player and Hero instances. This design delegates the instantiation responsibility to subclasses, allowing the client to handle any concrete product and factory through the use of abstract interface classes. For example, the constructor of AddHeroCommand can accept an object declared as HeroFactory. When a new type of hero is added, the user can easily extend behavior by adding a new concrete HeroFactory class and a new command factory, which will pass the new concrete HeroFactory to AddHeroCommand, thereby handling and utilizing the new factory without modification. This approach implements the Open-Closed Principle.

## Command Pattern

The Command Pattern encapsulates requests into objects. In this case, it includes statements for executing, redoing, and undoing commands. This behavioral design pattern is employed, where all commands are created as independent objects. They perform various functions, such as creating players, adding heroes, and setting the current player. Details are as follows:

1. **Invoker**: Main
2. **Command**: Command
3. **Concrete Commands**: AddHeroCommand, CallHeroSkillCommand, ChangePlayerNameCommand, CreatePlayerCommand, DeleteHeroCommand, DisplayAllPlayersCommand, ExitCommand, ListUndoRedoCommand, RedoCommand, SetCurrentPlayerCommand, ShowPlayerDetailCommand, UndoCommand
4. **Receiver**: Player, Hero, RefCurrentPlayerAdapter

The Main class acts as the invoker, holding all command instances and executing them based on user input. Concrete commands are implementations of the Command interface. The main features of the command classes are the execute(), undo(), and redo() methods. The receivers are Player, Hero, RefCurrentPlayerAdapter, and parts of specific command classes. The command classes interact with these concrete classes by invoking their methods. By adopting the Command Pattern, all request information is encapsulated within these command instances, making the structure of executing scripts more organized and allowing for undo and redo operations on executed commands. An ArrayList is created in Main to log executed commands. UndoCommand and RedoCommand will use this list to retrieve commands and invoke their undo and redo methods.

## Memento Pattern

The Memento Pattern aims to maintain the encapsulation of an object while saving its initial state for recovery. Details are as follows:

1. **Caretakers**: ChangePlayerNameCommand, CallHeroSkillCommand
2. **Mementos**: Memento, PlayerMemento, HeroMemento, WarriorMemento, WarlockMemento
3. **Originators**: Player, Hero, Warrior, Warlock

ChangePlayerNameCommand and CallHeroSkillCommand act as caretakers, responsible for creating, containing mementos, and deciding when to use mementos to save and restore an object's state. Memento is an interface, while PlayerMemento and HeroMemento are implementations. They accept a Player and Hero object, saving their states and executing recovery upon invocation. This design makes it easier to save and restore the states of Player and Hero objects. Through these two caretakers, the encapsulation of Player and Hero is better protected, as access is entirely managed by the memento instances.

## Adapter Pattern

The Adapter Pattern is used to convert one interface into another interface that the client expects. In this case, the RefCurrentPlayerAdapter class is used to transform the Player class interface into the interface expected by the client. Details are as follows:

1. **Target Interface**: Player
2. **Adapter**: RefCurrentPlayerAdapter
3. **Adaptee**: Player

The RefCurrentPlayerAdapter class holds a reference to a Player object and implements the methods of the Player interface. These methods call the corresponding methods of the adaptee Player object. In this way, the client can use RefCurrentPlayerAdapter to access the functionalities of the Player object without directly interacting with the Player object.

# Test Plan and Test Cases

## Test Case 1 – simple test

|  |
| --- |
| Input |
| c  P001  Thomas Yiu  a  H001, peter pang  1  a  H002, john wick  2  s  c  P002  Stan Lee  p  a  H003, scarlet witch  1  a  H004, tony stark  1  s  g  P001  m  H001  s  d  H002  t  Russo Brothers  s  l  u  u  u  s  u  u  g  P002  s  u  g  P002  l  r  r  r  s  r  g  P001  s  l  x |
| Output |
| Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player ID:-  Player Name:-  Player Thomas Yiu is created.  Current player is changed to P001.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero information (id, name):-  Hero Type (1 = Warrior | 2 = Warlock):-  Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero information (id, name):-  Hero Type (1 = Warrior | 2 = Warlock):-  Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Thomas Yiu (P001)  Heroes:  H001, peter pang, Warrior, Hp: 500, Damage: 0, Defence Point: 500  H002, john wick, Warlock, Hp: 100, Damage: 200, Mp: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player ID:-  Player Name:-  Player Stan Lee is created.  Current player is changed to P002.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Thomas Yiu (P001)  Player Stan Lee (P002)  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero information (id, name):-  Hero Type (1 = Warrior | 2 = Warlock):-  Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero information (id, name):-  Hero Type (1 = Warrior | 2 = Warlock):-  Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Stan Lee (P002)  Heroes:  H003, scarlet witch, Warrior, Hp: 500, Damage: 0, Defence Point: 500  H004, tony stark, Warrior, Hp: 500, Damage: 0, Defence Point: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input player ID:-  Changed current player to P001.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero ID:-  H001 peter pang's attributes are changed to:  H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence Point: 400  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Thomas Yiu (P001)  Heroes:  H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence Point: 400  H002, john wick, Warlock, Hp: 100, Damage: 200, Mp: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero ID:-  H002 john wick is deleted.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input new name of the current player:-  Player's name is updated.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Russo Brothers  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Russo Brothers (P001)  Heroes:  H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence Point: 400  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Russo Brothers  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Undo List:  Change player's name, P001, Russo Brothers  Delete hero, H002  CallHeroSkill, H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence: 400  Add hero, H004, tony stark, Warrior  Add hero, H003, scarlet witch, Warrior  Create player, P002, Stan Lee  Add hero, H002, john wick, Warlock  Add hero, H001, peter pang, Warrior  Create player, P001, Thomas Yiu  -- End of undo list --  Redo List:  -- End of redo list --  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Russo Brothers  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Change player's name, P001, Russo Brothers) is undone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Delete hero, H002) is undone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (CallHeroSkill, H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence: 400) is undone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Thomas Yiu (P001)  Heroes:  H001, peter pang, Warrior, Hp: 500, Damage: 0, Defence Point: 500  H002, john wick, Warlock, Hp: 100, Damage: 200, Mp: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Add hero, H004, tony stark, Warrior) is undone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Add hero, H003, scarlet witch, Warrior) is undone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input player ID:-  Changed current player to P002.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Stan Lee (P002)  Heroes:  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Create player, P002, Stan Lee) is undone.  Current player is changed to P001.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input player ID:-  Player P002 is not found!!  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Undo List:  Add hero, H002, john wick, Warlock  Add hero, H001, peter pang, Warrior  Create player, P001, Thomas Yiu  -- End of undo list --  Redo List:  Create player, P002, Stan Lee  Add hero, H003, scarlet witch, Warrior  Add hero, H004, tony stark, Warrior  CallHeroSkill, H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence: 400  Delete hero, H002  Change player's name, P001, Russo Brothers  -- End of redo list --  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Create player, P002, Stan Lee) is redone.  Current player is changed to P002.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Add hero, H003, scarlet witch, Warrior) is redone.  Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (Add hero, H004, tony stark, Warrior) is redone.  Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Stan Lee (P002)  Heroes:  H003, scarlet witch, Warrior, Hp: 500, Damage: 0, Defence Point: 500  H004, tony stark, Warrior, Hp: 500, Damage: 0, Defence Point: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (CallHeroSkill, H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence: 400) is redone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P002 Stan Lee  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input player ID:-  Changed current player to P001.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Thomas Yiu (P001)  Heroes:  H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence Point: 400  H002, john wick, Warlock, Hp: 100, Damage: 200, Mp: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Undo List:  CallHeroSkill, H001, peter pang, Warrior, Hp: 500, Damage: 250, Defence: 400  Add hero, H004, tony stark, Warrior  Add hero, H003, scarlet witch, Warrior  Create player, P002, Stan Lee  Add hero, H002, john wick, Warlock  Add hero, H001, peter pang, Warrior  Create player, P001, Thomas Yiu  -- End of undo list --  Redo List:  Delete hero, H002  Change player's name, P001, Russo Brothers  -- End of redo list --  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P001 Thomas Yiu  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :- |
| Cap screen |
|  |

## Test Case 2 – call hero skill in redo

|  |
| --- |
| Input |
| c  P1  Andy  a  H1, Andy Hero  1  s  m  H1  s  u  s  m  H1  l  r  s  l  x |
| Output |
| Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player ID:-  Player Name:-  Player Andy is created.  Current player is changed to P1.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero information (id, name):-  Hero Type (1 = Warrior | 2 = Warlock):- Hero is added.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Andy (P1)  Heroes:  H1, Andy Hero, Warrior, Hp: 500, Damage: 0, Defence Point: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero ID:-  H1 Andy Hero's attributes are changed to:  H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence Point: 400  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Andy (P1)  Heroes:  H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence Point: 400  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Command (CallHeroSkill, H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence: 400) is undone.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Andy (P1)  Heroes:  H1, Andy Hero, Warrior, Hp: 500, Damage: 0, Defence Point: 500  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please input hero ID:-  H1 Andy Hero's attributes are changed to:  H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence Point: 400  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Undo List:  CallHeroSkill, H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence: 400  Add hero, H1, Andy Hero, Warrior  Create player, P1, Andy  -- End of undo list --  Redo List:  CallHeroSkill, H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence: 400  -- End of redo list --  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Hero's data is changed, can't redo.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Player Andy (P1)  Heroes:  H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence Point: 400  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Undo List:  CallHeroSkill, H1, Andy Hero, Warrior, Hp: 500, Damage: 250, Defence: 400  Add hero, H1, Andy Hero, Warrior  Create player, P1, Andy  -- End of undo list --  Redo List:  -- End of redo list --  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  The current player is P1 Andy  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :- |
| Cap Screen |
|  |

## Test Case 3 – Error test

|  |
| --- |
| Input |
| G  A  M  D  S  T  U  R  x |
| Output |
| Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  No player is created yet!!  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please create / select a player first.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please create / select a player first.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please create / select a player first.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please create / select a player first.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  Please create / select a player first.  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  No commands to undo  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-  No commands to redo  Fantastic World (FW)  c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system  Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :- |
| Cap Screen |
|  |

# Source Code

|  |
| --- |
| Run Batch |
| File: run.bat  @echo off  echo Press d to delete all class files...  echo Press c to compile and run Main.java...  echo Press r to run Main.java  set /p choice="Enter your choice: "  if /i "%choice%"=="d" (  echo Deleting all class files...  del /s /q \*.class  ) else if /i "%choice%"=="c" (  echo Compiling Java files...  javac \*.java  javac ./FantasticWorld/\*.java  echo Press any key to run the program...  pause  java Main  ) else if /i "%choice%"=="r" (  java Main  ) else (  echo Invalid choice.  )  pause |

|  |
| --- |
| Main |
| File: Main.java  import java.util.\*;  import FantasticWorld.\*;  public class Main {  public static Scanner scanner = new Scanner(System.in);    public static void main(String[] args) {  Stack<Command> commands = new Stack<Command>();  Stack<Command> redos = new Stack<Command>();  Vector<Player> players = new Vector<Player>();  HashMap<String, CommandFactory> factories = new HashMap<String, CommandFactory>();  RefCurrentPlayerAdapter currentPlayer = new RefCurrentPlayerAdapter();    boolean playing = true;  HeroFactory[] heroType = {new WarriorFactory(), new WarlockFactory()};  /\*  \* c = create player, g = set current player, a = add hero, m = call hero skill,  \* d = delete hero, s = show player, p = display all players, t = change  \* player's name, u = undo, r = redo, l = list undo/redo, x = exit system  \*/  factories.put("c", new CreatePlayerCommandFactory(currentPlayer,commands, players, scanner));  factories.put("g", new SetCurrentPlayerCommandFactory(currentPlayer,commands, players, scanner));  factories.put("a", new AddHeroCommandFactory(currentPlayer,commands, heroType, scanner));  factories.put("m", new CallHeroSkillCommandFactory(currentPlayer,commands, players, scanner));  factories.put("d", new DeleteHeroCommandFactory(currentPlayer,commands, players, scanner));  factories.put("s", new ShowPlayerDetailCommandFactory(currentPlayer,commands, players, scanner));  factories.put("p", new DisplayAllPlayersCommandFactory(commands, players, scanner));  factories.put("t", new ChangePlayerNameCommandFactory(currentPlayer,commands, players, scanner));  factories.put("u", new UndoCommandFactory(commands, redos));  factories.put("r", new RedoCommandFactory(commands, redos));  factories.put("l", new ListUndoRedoCommandFactory(commands, redos));  factories.put("x", new ExitCommandFactory());  while (playing) {  System.out.println("\nFantastic World (FW)");  System.out.println(  "c = create player, g = set current player, a = add hero, m = call hero skill, d = delete hero, s = show player, p = display all players, t = change player's name, u = undo, r = redo, l = list undo/redo, x = exit system");  if (currentPlayer.getCurrentPlayer() != null) {  System.out.println(  "The current player is " + currentPlayer.getPlayerID() + " " + currentPlayer.getPlayerName());  }  System.out.print("Please enter command [ c | g | a | m | d | s | p | t | u | r | l | x ] :-");  String InputCom = scanner.nextLine().trim().toLowerCase();  if(!factories.containsKey(InputCom)){  System.out.println("Invalid command: " + InputCom);  continue;  }  try {  Command com = factories.get(InputCom).create();  com.execute();  } catch (Exception e) {  System.out.println(e.getMessage());  }    }  }  } |

|  |
| --- |
| package FantasticWorld |
| File: AddHeroCommand.java  package FantasticWorld;  import java.util.Scanner;  import java.util.Stack;  public class AddHeroCommand extends RecordString implements Command {  private Player currentPlayer;  private Stack<Command> commands;  private HeroFactory[] heroType;  private Scanner scanner;  Hero h;  // int index;  Player targetPlayer;  public void execute() {  if (((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer() == null) {  System.out.println("\nPlease create / select a player first.");  return;  }  targetPlayer = ((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer();  String s;  System.out.print("\nPlease input hero information (id, name):- ");  s = scanner.nextLine();  String[] heroInfo = s.split(",| ", 2);  System.out.print("\nHero Type (");  for (int i = 0; i < heroType.length; i++) {  System.out.print((i + 1) + " = " + heroType[i]);  if (i < heroType.length - 1) {  System.out.print(" | ");  }  }  System.out.print("):- ");  s = scanner.nextLine().trim();  int intHeroType = Integer.parseInt(s);  if (intHeroType < 1 || intHeroType > heroType.length) {  throw new IllegalArgumentException("Invalid hero type.");  }  h = heroType[intHeroType - 1].create(heroInfo);  // index = currentPlayer.getHeroes().size();  targetPlayer.addHero(h);  System.out.println("Hero is added.");  setRecordString("Add hero, " + h.getHeroID() + ", " + h.getHeroName() + ", " + h.getClass().getSimpleName());  commands.push(this);  }  public void undo() {  targetPlayer.removeHero(h);  }  public AddHeroCommand(Player currentPlayer, Stack<Command> commands, HeroFactory[] heroType, Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.heroType = heroType;  this.scanner = scanner;  }  @Override  public boolean redo() {  targetPlayer.addHero(h);  System.out.println("Hero is added.");  setRecordString("Add hero, " + h.getHeroID() + ", " + h.getHeroName() + ", " + h.getClass().getSimpleName());  return true;  }  }  File: AddHeroCommandFactory.java  package FantasticWorld;  import java.util.Scanner;  import java.util.Stack;  public class AddHeroCommandFactory implements CommandFactory {  private RefCurrentPlayerAdapter currentPlayer;  private Stack<Command> commands;  private HeroFactory[] heroType;  private Scanner scanner;  public Command create() {  return new AddHeroCommand(currentPlayer,commands, heroType, scanner);  }  public AddHeroCommandFactory(RefCurrentPlayerAdapter currentPlayer,Stack<Command> commands, HeroFactory[] heroType,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.heroType = heroType;  this.scanner = scanner;  }  }  File: CallHeroSkillCommand.java  package FantasticWorld;  import java.util.Vector;  import java.lang.reflect.InvocationTargetException;  import java.util.Scanner;  import java.util.Stack;  public class CallHeroSkillCommand extends RecordString implements Command {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  private Hero h;  private HeroMemento heroMemento;  public void execute() {  if (((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer() == null) {  System.out.println("\nPlease create / select a player first.");  return;  }  System.out.print("\nPlease input hero ID:- ");  String heroID = scanner.nextLine().trim();  System.out.println();  for (Hero hero : currentPlayer.getHeroes()) {  if (hero.getHeroID().equals(heroID)) {  h = hero;  heroMemento = makHeroMemento(hero);  hero.callSkill();  heroMemento.setRecordString();  System.out.println(hero.getHeroID() + " " + hero.getHeroName() + "'s attributes are changed to:");  hero.showHeroStatus();  heroMemento.setRecordString();  setRecordString("CallHeroSkill, " + heroMemento.getRecordString());  commands.push(this);  return;  }  }  }  public HeroMemento makHeroMemento(Hero hero) {  HeroMemento heroMemento;  String heroClass = hero.getClass().getSimpleName();  try {  heroMemento = (HeroMemento) Class.forName("FantasticWorld." + heroClass + "Memento")  .getConstructor(Class.forName("FantasticWorld." + heroClass)).newInstance(hero);  heroMemento.save();  } catch (InstantiationException | IllegalAccessException | IllegalArgumentException  | InvocationTargetException | NoSuchMethodException | SecurityException  | ClassNotFoundException e) {  // TODO Auto-generated catch block  e.printStackTrace();  return null;  }  return heroMemento;  }  public void undo() {  heroMemento.restore();  }  public boolean redo() {  // save for fail  HeroMemento TempHeroMemento = makHeroMemento(h);  int[] nowData = TempHeroMemento.getHeroData();  int[] orgData = heroMemento.getHeroData();  boolean diff = false;  for (int i = 0; i < nowData.length; i++) {  if (nowData[i] != orgData[i]) {  diff = true;  break;  }  }  if (diff) {  System.out.println("Hero's data is changed, can't redo.");  return false;  }  h.callSkill();  return true;  }  public CallHeroSkillCommand(Player currentPlayer, Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: CallHeroSkillCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class CallHeroSkillCommandFactory implements CommandFactory {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new CallHeroSkillCommand(currentPlayer,commands, players, scanner);  }  public CallHeroSkillCommandFactory(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: ChangePlayerNameCommand.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class ChangePlayerNameCommand extends RecordString implements Command {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  private PlayerMemento playerMemento;  private Player player;  public void execute() {  if (((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer() == null) {  System.out.println("\nPlease create / select a player first.");  return;  }  System.out.print("\nPlease input new name of the current player:- ");  String newPlayerName = scanner.nextLine().trim();  player = ((RefCurrentPlayerAdapter)currentPlayer).getCurrentPlayer();  playerMemento = new PlayerMemento(player);  playerMemento.save();  player.setPlayerName(newPlayerName);  System.out.println("\nPlayer's name is updated.");  setRecordString("Change player's name, "+ player.getPlayerID() +", "+player.getPlayerName());  commands.push(this);  }  public void undo() {  PlayerMemento newMemento = new PlayerMemento(player);  newMemento.save();  playerMemento.restore();  playerMemento = newMemento;  }  public boolean redo() {  undo();  return true;  }  public ChangePlayerNameCommand(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: ChangePlayerNameCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class ChangePlayerNameCommandFactory implements CommandFactory {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new ChangePlayerNameCommand(currentPlayer,commands, players, scanner);  }  public ChangePlayerNameCommandFactory(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: Command.java  package FantasticWorld;  public interface Command {  public void execute();  public void undo();  public boolean redo();  }  File: CommandFactory.java  package FantasticWorld;  public interface CommandFactory extends Factory {  public Command create();  }  File: CreatePlayerCommand.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class CreatePlayerCommand extends RecordString implements Command {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  Player p, lastPlayer;  String msg = "Current player is changed to ";  public void execute() {  if (p == null) {  p = new PlayerFactory(scanner).create();  lastPlayer = ((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer();  }  players.add(p);  ((RefCurrentPlayerAdapter) currentPlayer).setCurrentPlayer(p,msg);  setRecordString("Create player, " + p.getPlayerID() + ", " + p.getPlayerName());  commands.push(this);  }  public void undo() {  players.remove(p);  ((RefCurrentPlayerAdapter) currentPlayer).setCurrentPlayer(lastPlayer,msg);  }  public boolean redo() {  players.add(p);  ((RefCurrentPlayerAdapter) currentPlayer).setCurrentPlayer(p,msg);  return true;  }  public CreatePlayerCommand(Player currentPlayer, Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: CreatePlayerCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class CreatePlayerCommandFactory implements CommandFactory {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new CreatePlayerCommand(currentPlayer,commands, players, scanner);  }  public CreatePlayerCommandFactory(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: DeleteHeroCommand.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class DeleteHeroCommand extends RecordString implements Command {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  private Hero hero;  private Player targetPlayer;  public void execute() {  if (((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer() == null) {  System.out.println("\nPlease create / select a player first.");  return;  }  System.out.print("\nPlease input hero ID:- ");  String heroID = scanner.nextLine().trim();  System.out.println();  for (Hero h : currentPlayer.getHeroes()) {  if (h.getHeroID().equals(heroID)) {  targetPlayer = ((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer();  targetPlayer.removeHero(h);  hero = h;  System.out.println(h.getHeroID() + " " + h.getHeroName() + " is deleted.");  setRecordString("Delete hero, "+h.getHeroID());  commands.push(this);  return;  }  }  System.out.println("Hero " + heroID + " is not found!!");  }  public void undo() {  targetPlayer.addHero(hero);  }  public boolean redo() {  targetPlayer.removeHero(hero);  return true;  }  public DeleteHeroCommand(Player currentPlayer, Stack<Command> commands, Vector<Player> players, Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: DeleteHeroCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class DeleteHeroCommandFactory implements CommandFactory {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new DeleteHeroCommand(currentPlayer,commands, players, scanner);  }  public DeleteHeroCommandFactory(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: DisplayAllPlayersCommand.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class DisplayAllPlayersCommand implements Command {  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public void execute() {  System.out.println();  for (Player p : players) {  System.out.println("Player " + p.getPlayerName() + " (" + p.getPlayerID() + ")");  }  }  public void undo() {  }  public DisplayAllPlayersCommand(Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  @Override  public boolean redo() {  return true;  }  }  File: DisplayAllPlayersCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class DisplayAllPlayersCommandFactory implements CommandFactory {  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new DisplayAllPlayersCommand(commands, players, scanner);  }  public DisplayAllPlayersCommandFactory(Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: ExitCommand.java  package FantasticWorld;  public class ExitCommand implements Command {  public void execute(){  System.exit(0);  }  public void undo(){    }  public boolean redo(){  return true;  }  }  File: ExitCommandFactory.java  package FantasticWorld;  public class ExitCommandFactory implements CommandFactory{  public Command create(){  return new ExitCommand();  }  }  File: Factory.java  package FantasticWorld;  public interface Factory {  public Object create();  }  File: Hero.java  package FantasticWorld;  public abstract class Hero {  private String heroID;  private String heroName;  private int hp;  private int damage;  public Hero(String heroID, String heroName) {  this.heroID = heroID;  this.heroName = heroName;  this.hp = 200;  }  public String getHeroID() {  return heroID;  }  public String getHeroName() {  return heroName;  }  public void setHeroName(String heroName) {  this.heroName = heroName;  }  public int getHp() {  return hp;  }  public void setHp(int hp) {  this.hp = hp;  }  public int getDamage() {  return damage;  }  public void setDamage(int damage) {  this.damage = damage;  }    public abstract void callSkill();    public abstract void showHeroStatus();    }  File: HeroFactory.java  package FantasticWorld;  public abstract class HeroFactory implements Factory {  String[] heroInfo;  public Object create() {  throw new UnsupportedOperationException();  }  public Hero create(String[] heroInfo) {  this.heroInfo = heroInfo;  Hero h= createHero();  return h;  }  public abstract Hero createHero();  public abstract String toString();  }  File: HeroMemento.java  package FantasticWorld;  public abstract class HeroMemento extends RecordString implements Memento {  private Hero hero;  private int hp, damage;  public void restore() {  hero.setHp(hp);  hero.setDamage(damage);  }  public void save() {  hp = hero.getHp();  damage = hero.getDamage();  }  public HeroMemento(Hero hero) {  this.hero = hero;  save();  }  Hero getHero() {  return hero;  }  public int getHp() {  return hp;  }    public int getDamage() {  return damage;  }  public abstract void setRecordString();  public abstract int[] getHeroData();  }  File: ListUndoRedoCommand.java  package FantasticWorld;  import java.util.Stack;  public class ListUndoRedoCommand implements Command {  private Stack<Command> commands, redos;  public void execute() {  System.out.println("\nUndo List:");  for (int i = commands.size() - 1; i >= 0; i--) {  System.out.println(commands.get(i).toString());  }  System.out.println("-- End of undo list --");  System.out.println("Redo List:");  for (int i = redos.size() - 1; i >= 0; i--) {  System.out.println(redos.get(i).toString());  }  System.out.println("-- End of redo list --");  }  public void undo() {  return;  }  public ListUndoRedoCommand(Stack<Command> commands, Stack<Command> redos) {  this.commands = commands;  this.redos = redos;  }  public boolean redo() {  return true;  }  }  File: ListUndoRedoCommandFactory.java  package FantasticWorld;  import java.util.Stack;  public class ListUndoRedoCommandFactory implements CommandFactory {  private Stack<Command> commands, redos;  public Command create() {  return new ListUndoRedoCommand(commands, redos);  }  public ListUndoRedoCommandFactory(Stack<Command> commands, Stack<Command> redos) {  this.commands = commands;  this.redos = redos;  }  }  File: Memento.java  package FantasticWorld;  public interface Memento {  public void restore();  public void save();  }  File: Player.java  package FantasticWorld;  import java.util.Vector;  public class Player {  private String playerID;  private String playerName;  private Vector<Hero> heroes;  public Player(String playerID,String playerName) {  this.playerID = playerID;  this.playerName = playerName;  this.heroes = new Vector<Hero>();  }  public String getPlayerID() {  return playerID;  }  public String getPlayerName() {  return playerName;  }  public void setPlayerName(String playerName) {  this.playerName = playerName;  }  public Vector<Hero> getHeroes() {  return heroes;  }  public void addHero(Hero hero) {  heroes.add(hero);  }    public void removeHero(Hero hero) {  heroes.remove(hero);  }    public void showPlayerDetails(){  System.out.println("Player " + playerName + " (" + playerID + ")");  System.out.println("Heroes:");  for (Hero hero : heroes) {  hero.showHeroStatus();  }  }  }  File: PlayerFactory.java  package FantasticWorld;  import java.util.Scanner;  public class PlayerFactory implements Factory {  private Scanner scanner;  public Player create() {  Player p;  String id,name;  System.out.print("\nPlayer ID:- ");  id = scanner.nextLine().trim();  System.out.print("\nPlayer Name:- ");  name = scanner.nextLine().trim();  p = new Player(id,name);  System.out.println("\nPlayer " + p.getPlayerName() + " is created.");  return p;  }  public PlayerFactory(Scanner scanner) {  this.scanner = scanner;  }  }  File: PlayerMemento.java  package FantasticWorld;  public class PlayerMemento implements Memento {  private String playerName;  private Player player;  public PlayerMemento(Player player) {  //this.playerName.push(player.getPlayerName());  this.player = player;  save();  }  public void restore() {  player.setPlayerName(playerName);  }  public void save() {  playerName=player.getPlayerName();  }  public Player getPlayer() {  return player;  }  }  File: RecordString.java  package FantasticWorld;  public abstract class RecordString {  private String recordString;  public void setRecordString(String recordString) {  this.recordString = recordString;  }  public String getRecordString() {  return recordString;  }  public String toString() {  return getRecordString();  }  }  File: RedoCommand.java  package FantasticWorld;  import java.util.EmptyStackException;  import java.util.Stack;  public class RedoCommand implements Command {  private Stack<Command> commands, redos;  public void execute() {  System.out.println();  try {  Command c = redos.pop();  boolean is\_done = c.redo();  if (is\_done) {  System.out.println("Command (" + c.toString() + ") is redone.");  commands.push(c);  }  } catch (EmptyStackException e) {  System.out.println("No commands to redo");  }  }  public void undo() {  }  public boolean redo() {  return true;  }  public RedoCommand(Stack<Command> commands, Stack<Command> redos) {  this.commands = commands;  this.redos = redos;  }  }  File: RedoCommandFactory.java  package FantasticWorld;  import java.util.Stack;  public class RedoCommandFactory implements CommandFactory {  private Stack<Command> commands;  private Stack<Command> redos;  public Command create() {  return new RedoCommand(commands, redos);  }  public RedoCommandFactory(Stack<Command> commands, Stack<Command> redos) {  this.commands = commands;  this.redos = redos;  }  }  File: RefCurrentPlayerAdapter.java  package FantasticWorld;  import java.util.Vector;  public class RefCurrentPlayerAdapter extends Player {  private Player currentPlayer = null;  public Player getCurrentPlayer() {  return currentPlayer;  }  public void setCurrentPlayer(Player currentPlayer,String msg) {  this.currentPlayer = currentPlayer;  if (currentPlayer != null) {  System.out.println(msg + this.currentPlayer.getPlayerID() + ".");  }  }  @Override  public String getPlayerID() {  return currentPlayer.getPlayerID();  }  @Override  public String getPlayerName() {  return currentPlayer.getPlayerName();  }  @Override  public void setPlayerName(String playerName) {  currentPlayer.setPlayerName(playerName);  }  @Override  public Vector<Hero> getHeroes() {  return currentPlayer.getHeroes();  }  @Override  public void addHero(Hero hero) {  currentPlayer.addHero(hero);  }  @Override  public void removeHero(Hero hero) {  currentPlayer.removeHero(hero);  }  @Override  public void showPlayerDetails() {  System.out.println("Player " + getPlayerName() + " (" + getPlayerID() + ")");  System.out.println("Heroes:");  for (Hero hero : getHeroes()) {  hero.showHeroStatus();  }  }  public RefCurrentPlayerAdapter() {  super(null,null);  }  }  File: SetCurrentPlayerCommand.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class SetCurrentPlayerCommand extends RecordString implements Command {  Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  Player player,lastPlayer;  String msg = "Changed current player to ";  public void execute() {  if(players.isEmpty()){  System.out.println("\nNo player is created yet!!");  return;  }  System.out.print("\nPlease input player ID:- ");  String playerID = scanner.nextLine().trim();  System.out.println();  for (Player p : players) {  if (p.getPlayerID().equals(playerID)) {  player = p;  lastPlayer = ((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer();  ((RefCurrentPlayerAdapter) currentPlayer).setCurrentPlayer(p,msg);  setRecordString("Set current player, " + p.getPlayerID() + ", " + p.getPlayerName());    return;  }  }  System.out.println("Player " + playerID + " is not found!!");  }  public void undo() {  ((RefCurrentPlayerAdapter) currentPlayer).setCurrentPlayer(lastPlayer,msg);  }  public boolean redo() {  ((RefCurrentPlayerAdapter) currentPlayer).setCurrentPlayer(player,msg);  return true;  }  public SetCurrentPlayerCommand(Player currentPlayer, Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: SetCurrentPlayerCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class SetCurrentPlayerCommandFactory implements CommandFactory {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new SetCurrentPlayerCommand(currentPlayer,commands, players, scanner);  }  public SetCurrentPlayerCommandFactory(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: ShowPlayerDetailCommand.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class ShowPlayerDetailCommand implements Command {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public void execute() {  if (((RefCurrentPlayerAdapter) currentPlayer).getCurrentPlayer() == null) {  System.out.println("\nPlease create / select a player first.");  return;  }  System.out.println();  currentPlayer.showPlayerDetails();  }  public void undo() {}  public boolean redo() {  return true;  }  public ShowPlayerDetailCommand(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: ShowPlayerDetailCommandFactory.java  package FantasticWorld;  import java.util.Vector;  import java.util.Scanner;  import java.util.Stack;  public class ShowPlayerDetailCommandFactory implements CommandFactory {  private Player currentPlayer;  private Stack<Command> commands;  private Vector<Player> players;  private Scanner scanner;  public Command create() {  return new ShowPlayerDetailCommand(currentPlayer,commands, players, scanner);  }  public ShowPlayerDetailCommandFactory(Player currentPlayer,Stack<Command> commands, Vector<Player> players,  Scanner scanner) {  this.currentPlayer = currentPlayer;  this.commands = commands;  this.players = players;  this.scanner = scanner;  }  }  File: UndoCommand.java  package FantasticWorld;  import java.util.Stack;  public class UndoCommand implements Command {  private Stack<Command> commands, redos;  public void execute() {  System.out.println();  try{  Command c = commands.pop();  System.out.println("Command ("+c.toString()+") is undone.");  c.undo();  redos.push(c);}  catch(Exception e){  System.out.println("No commands to undo");  }  }  public void undo() {}  public boolean redo() {  return true;  }  public UndoCommand(Stack<Command> commands, Stack<Command> redos) {  this.commands = commands;  this.redos = redos;  }  }  File: UndoCommandFactory.java  package FantasticWorld;  import java.util.Stack;  public class UndoCommandFactory implements CommandFactory {  private Stack<Command> commands;  private Stack<Command> redos;  public Command create() {  return new UndoCommand(commands, redos);  }  public UndoCommandFactory(Stack<Command> commands, Stack<Command> redos) {  this.commands = commands;  this.redos = redos;  }  }  File: Warlock.java  package FantasticWorld;  public class Warlock extends Hero{  private int mp;    public Warlock(String heroID, String heroName) {  super(heroID, heroName);  this.setHp(100);  this.mp = 500;  this.setDamage(200);  }    public int getMp() {  return mp;  }  public void setMp(int mp) {  this.mp = mp;  }  @Override  public void callSkill() {  if (mp > 0) mp -= 100;  else mp = 0;  setDamage(100);  }  @Override  public void showHeroStatus() {  System.out.println(getHeroID() +  ", "+getHeroName() + ", Warlock, Hp: "+getHp() +  ", Damage: "+getDamage() + ", Mp: "+mp);  }  }  File: WarlockFactory.java  package FantasticWorld;  public class WarlockFactory extends HeroFactory {  public Hero createHero() {  return new Warlock(heroInfo[0].trim(),heroInfo[1].trim());  }  public String toString() {  return "Warlock";  }  }  File: WarlockMemento.java  package FantasticWorld;  public class WarlockMemento extends HeroMemento {  private int mp;  public WarlockMemento(Warlock warlock) {  super(warlock);  }  @Override  public void restore() {  super.restore();  ((Warlock) this.getHero()).setMp(mp);  }  @Override  public void save() {  super.save();  mp = ((Warlock) this.getHero()).getMp();    }  public int getMp() {  return mp;  }  public void setRecordString() {  Hero h = this.getHero();  setRecordString(h.getHeroID() +  ", "+h.getHeroName() + ", Warlock, Hp: "+h.getHp() +  ", Damage: "+h.getDamage() + ", Mp: "+((Warlock)h).getMp());  }  public int[] getHeroData() {  int[] data = new int[3];  data[0] = this.getHp();  data[1] = this.getDamage();  data[2] = this.getMp();  return data;  }  }  File: Warrior.java  package FantasticWorld;  public class Warrior extends Hero{  private int defencePoint;  public Warrior(String heroID, String heroName) {  super(heroID, heroName);  this.defencePoint = 500;  this.setHp(500);  this.setDamage(0);  }  public int getDefencePoint() {  return defencePoint;  }  public void setDefencePoint(int defencePoint) {  this.defencePoint = defencePoint;  }  @Override  public void callSkill() {  if (defencePoint<0) defencePoint = 0;  this.setDamage(defencePoint/2);  defencePoint -= 100;  }  @Override  public void showHeroStatus() {  System.out.println(getHeroID() +  ", "+getHeroName() + ", Warrior, Hp: "+getHp() +  ", Damage: "+getDamage() + ", Defence Point: "+defencePoint);  }  }  File: WarriorFactory.java  package FantasticWorld;  public class WarriorFactory extends HeroFactory {  public Hero createHero() {  return new Warrior(heroInfo[0].trim(),heroInfo[1].trim());  }  public String toString() {  return "Warrior";  }  }  File: WarriorMemento.java  package FantasticWorld;  public class WarriorMemento extends HeroMemento {  private int defencePoint;  public WarriorMemento(Warrior warrior) {  super(warrior);  }  @Override  public void restore() {  super.restore();  ((Warrior) this.getHero()).setDefencePoint(defencePoint);  }  @Override  public void save() {  super.save();  defencePoint = ((Warrior) this.getHero()).getDefencePoint();  }  public int getDefencePoint() {  return defencePoint;  }  public void setRecordString() {  Hero h = this.getHero();  setRecordString(h.getHeroID() +  ", "+h.getHeroName() + ", Warrior, Hp: "+h.getHp() +  ", Damage: "+h.getDamage() + ", Defence: "+((Warrior)h).getDefencePoint());  }  public int[] getHeroData() {  int[] data = new int[3];  data[0] = this.getHp();  data[1] = this.getDamage();  data[2] = this.getDefencePoint();  return data;  }  } |

-- End --