|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6122  Framework Layer Architecture |
| **Computer Science** | **O213-COMP6122-SK01-02** |
| ***Valid on*** *Odd Semester Year 2020/2021* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + 1. Melihat sebagian atau seluruh jawaban mahasiswa lain,

*Seeing a part or the whole answer from other student*

* + 1. Menyadur sebagian maupun seluruh jawaban dari buku,

*Adapted a part or the whole answer from the book*

* + 1. Mendownload sebagian maupun seluruh jawaban dari internet,

*Downloading a part or the whole answer from the internet,*

* + 1. Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal,

*Working with another theme which is not in accordance with the existing theme in the matter of the case,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika mahasiswa terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai mahasiswa** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the student is proved to the actions described in point 1 above, the score of the student which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan jawaban, segala jenis pengumpulan jawaban di luar jadwal tidak dilayani.

*Pay attention to the submission schedule, all kinds of submission outside the schedule will not be accepted*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 100% | - | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Java 8  Eclipse 2020.6 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| JAVA, CLASS | - | - |

## Soal

*Case*

**AXForGirlfriend**

**Criteria**:

1. Template  
   First **Template**. **Template** pattern is a pattern that lets you break down an algorithm into a series of steps, turn these steps into methods, and put a series of calls to these methods inside a single template method.
2. State

Last **state**. **state** is a behavioral design pattern that lets an object alter its behavior when its internal state changes. It appears as if the object changed its class.

AXForGirlfriend is a dating simulator game. Here your goal is to make the girl you meet love you and want to be your girlfriend. Now you are a new programmer that assigns to develop this game. The program will be created with the following criteria:

1. **Menu**

First, the program will show the menu. In this menu, the lecturer can choose whether they want to **Ask For Girlfriend, How to play,** and **Exit**.

Text

Description automatically generated

Figure 1. AXForGirlfriend Menu

1. **Ask For Girlfriend**

This menu will be the gameplay menu of the game. The gameplay always follow a template set of methods. Below is the sequence of methods that will be executed in the given order:

|  |  |  |
| --- | --- | --- |
| **Method** | **Player** | **Girlfriend** |
| String run() | This method will run all the method below in the given order | This method will run all the method below in the given order |
| Void introduction() | Player will print the current day | The girlfriend will greet the player according to its state |
| Void action() | Print the list of actions and prompt the user to choose 1 | Do nothing |
| Void updateResource() | Increment the current day by 1 | Take the action given by the player then give an appropriate response based on how much the girl likes the action (point). Also don’t forget to update the girlfriend’s happiness. |
| String response() | Return the action chosen by the player | If default state do nothing else if angry or happy state return a message to notify the game has finished |

If the player chooses **Ask For Girlfriend**, display the current day then show all the action and ask the user to input the option to select the action. The day always start from 0 and increment by 1 at the end of every turn. The list of available actions are as below:

* **Give Compliment**
* **Give Hug**
* **Spend Time Together**
* **Give a Gift**
* **Help With Chores**

When the player inputs an action validate the input must be **exactly the same** as the **action.**

Graphical user interface, text, application

Description automatically generated

Figure 2. Ask For Girlfriend

After choosing the action to perform for the girl, you need to play the scenario between the player and the girl. The girl will give a respond according to their liking. You need to generate each girl liking **by randomly assign 10 points across the five-given action**. Below are the criteria to determine if the girl likes the given action or not:

|  |  |
| --- | --- |
| **Points** | **Respond** |
| Less than 5 | ”Hehehe... thankyouu...” |
| 5 | “\*GASP\* Thankyou, dear.” |
| More than 5 | “\*GASP\* Thankyou!!!"” |

Every action the player perform for the girl will increase the girl’s happiness meter for a certain point based on how much the girl likes the action. There are 3 possible criteria that could happen based on the girlfriend happiness meter :

|  |  |  |
| --- | --- | --- |
| **Case** | **Scenario** | **Trigger** |
| The girlfriend despise the given action (has 0 point for it) | The girlfriend will be disappointed and now despise you. This will result in the player losing the game and redirect them back to main menu. | When the player do an action that the girl has 0 point of liking for |
| The girlfriend accept the given action (has positive point for it) | The girlfriend accepts the action and the happiness meter will continue to accumulate. The player will now continue to play until the happiness meter reaches 10. | This is the default state, the girlfriend will always start in this state |
| The girlfriend has fallen in love (current happiness meter had reached 10 points) | The girlfriend is now in love with the player, thus resulting in you winning the game. | When the girl’s happiness point reaches 10 |

Below are the screenshot for every scenario above:

Text

Description automatically generated

Figure 4. When the girlfriend accepts the action

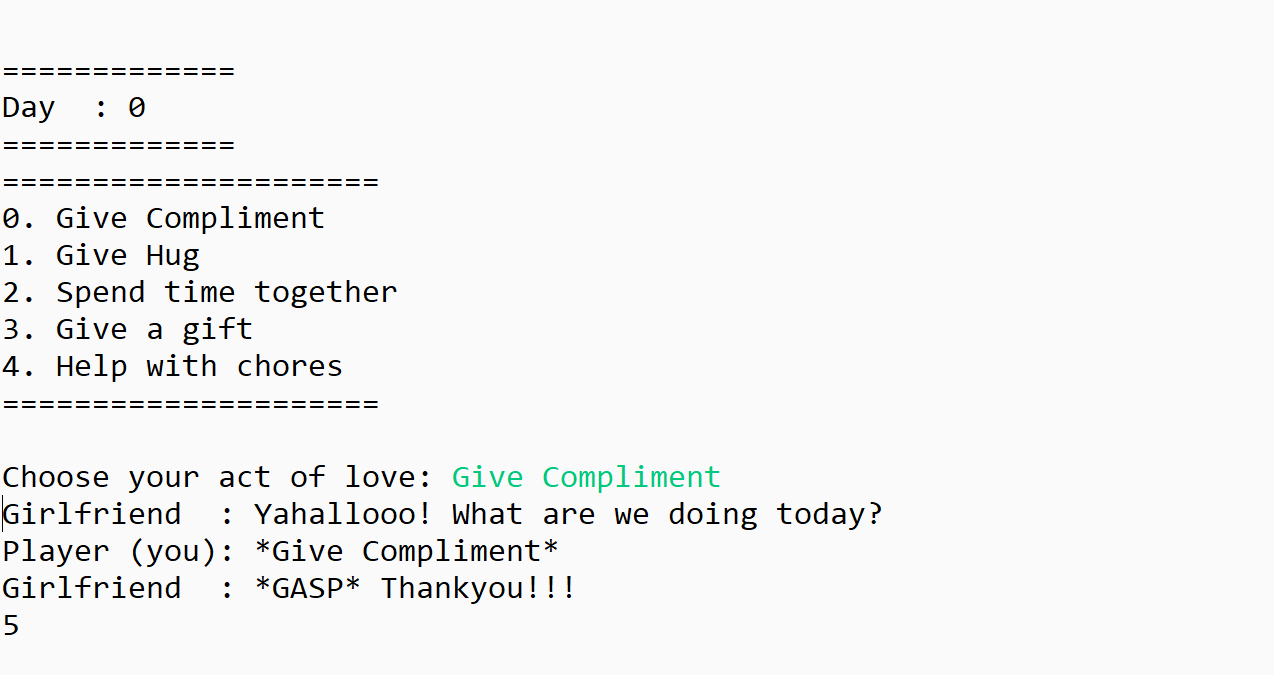


Figure 4. When the girlfriend accepts the action

Text

Description automatically generated with medium confidence

Figure 5. When the girlfriend accepts the action

1. **Exit**  
   If user choose **Exit,** Exit the program clearly.

***If there are any problems, please ask your assistant!***