|  |  |
| --- | --- |
| **Assignment Case** |  |
| ISYS6197003  Business Application Development |
| **Computer Science** | **O203-ISYS6197-VC08-01** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **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* |
| 40% | 60% | - |

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  XAMPP 7.4.7  MySQL Java Connection Library 5.1.49 |

## 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 | JAVA, CLASS, SQL | - |

## Soal

*Case*

**Five Food Street**

**Criteria**:

1. Abstract Class

You need to design at least **three** classes, **one abstract** class, and **two concrete** classes. Abstract class consists of all **common** attributes and behavior that both of concrete class had. Concrete class consist of **specific** attribute and behavior that not common between the concrete classes

1. Encapsulation

To **hide** the data of a class from an **illegal** direct access, all of the attributes of the class must be **encapsulated** and will be accessed using an **accessor** and **mutator** that may perform validation before accessing the encapsulated attribute

1. Inheritance

All of the concrete class **must inherit all** attribute and behavior from the abstract class

1. Polymorphism

If the concrete class has **a specific implementation** of the inherited behavior (method) that **differ** from the abstract class, the concrete class can **override** or **overload** the behavior from the abstract class

Five Food Street is a small food restaurant that sells Vegetarian Food and Non-Vegetarian Food. This restaurant is getting bigger to the point where the seller started to get into trouble when serving the customer. John, your manager, asks you to make a program that will help the seller to sell their foods. As a good programmer, you must make this program based on the following criteria:

1. **Menu**

At first, the program will show the menu. In this menu, the seller can choose whether they want to **insert new menu**, **view all the menu**, **sell a menu item**, or **exit the program**.



**Figure 1. Menu**

1. **Insert New Menu**

If the seller chooses to **insert a new menu**, then ask what menu they want to insert. The criteria for the menu detail:

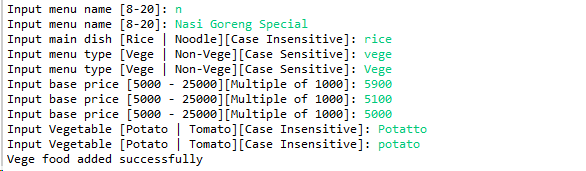
* First, ask for the **food name**. Validate the length of the menu name must be **between 8 and 20 characters** (**inclusive**)
* After that, ask for the **main dish**. Validate the input can only be either ‘**Rice’**, or ‘**Noodle’** (**case insensitive**)
* Then, ask for the **menu type**. Validate the input can only be either ‘**Vege’**, or ‘**Non-Vege’** (**case sensitive**)
* Then, ask for the menu’s **base price**. Validate the price of the menu must be **between 5000 and 25000** (**inclusive**) and must be **a multiple of 1000**
  + - * If the seller chooses **Vege**, then ask for more details about the menu based on the following criteria:
* First, ask for the **vegetable**. Validate the input only can be either ‘**Potato’**, or ‘**Tomato’** (**case insensitive**)
* After asking for the menu’s details, **generate** an ID for the menu based on the following format:

**ID = ‘VV[0-9][0-9][0-9]’**

**example: VV001**

ID number will **start** from **1** and will be **increased by 1** for each new menu. The ID increment is **shared** between Vege and Non-Vege menu

* Finally, add the food to an Array / Array List / Vector



**Figure 2. Insert vege food**

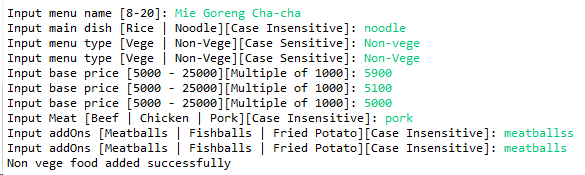
* + - * If the seller chooses **Non-Vege**, then ask for more details about the menu based on the following criteria:
* First, ask for the **meat**. Validate the input can only be either ‘**Beef’**, ‘**Chicken**’, or ‘**Pork**’ (**case insensitive**)
* After that, ask for the **add-ons**. Validate the input can only be either ‘**Meatballs’**, ‘**Fishballs**’, or ‘**Fried Potato’** (**case insensitive**)
* After asking for the menu’s details, **generate** an ID for the food based on the following format:

**ID = ‘NV[0-9][0-9][0-9]’**

**example: NV001**

ID number will **start** from **1** and will be **increased by 1** for each new menu. The ID increment is **shared** between Vege and Non-Vege menu

* Finally, add the food to an Array / Array List / Vector



**Figure 3. Insert non-vege food**

1. **View All Menu**

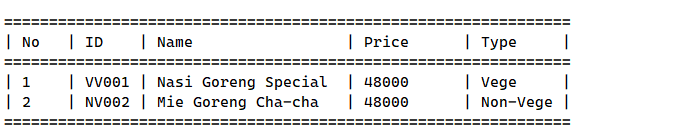
If the seller chooses to **view all the menu**, then validate the view based on the following criteria:

* + - * If there are **no** menu, then show a message



**Figure 4. View menus with no menu**

* + - * If there are menu(s), then show all menus’ details



**Figure 5. View menus**

* The price will be **calculated** using an **abstract method** in the **superclass** and it will be overridden in the **subclass**. The calculation formula will be different for each menu type. Below are the criteria:
* **Vege and Non-Vege**

|  |  |  |
| --- | --- | --- |
| **Main Dish** | **Rice** | **Noodle** |
| **Price** | 5000 | 3000 |

* **Vege**

|  |  |  |
| --- | --- | --- |
| **Vege Food** | **Potato** | **Tomato** |
| **Sub Total Price** | Menu name length \* 2000 | Menu name length \* 1000 |
| **Total Price** | Base Price + Sub Total Price + Main Dish Price | |

* **Non-Vege**

|  |  |  |  |
| --- | --- | --- | --- |
| **Addons** | **Meatballs** | **Fishballs** | **Fried Potato** |
| **Price** | 4000 | 3000 | 5000 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Non-Vege** | **Beef** | **Chicken** | **Pork** |
| **Sub Total Price** | Menu name length \* 2500 | Menu name length \* 1000 | Menu name length \* 2000 |
| **Total Price** | Base Price + Sub Total + Addons Price + Main Dish Price | | |

1. **Sell Menu Item**

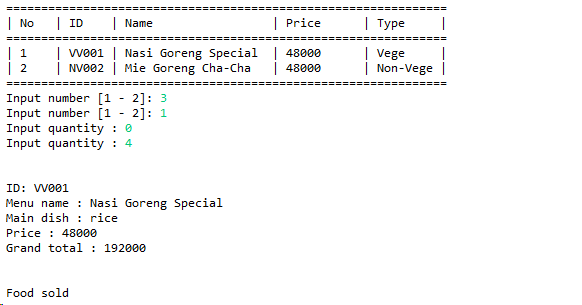
If the seller chooses to **sell a menu item**, then validate the menu based on the following criteria:

* + - * If there are **no** menu, then show a message



**Figure 6. Sell menu with no menu**

* + - * If there are menus available, then ask the seller about the menu based on the following criteria:
* First, ask which food they want to sell. Validate the seller can only input between **1** tothe **amount of menus** that **exists (inclusive)**
* Then, ask for the quantity of the menu item. Validate the quantity must be **more than 0**
* Lastly, **remove** the selected menu item from the menu list and **print** a **receipt** for the customer with the detail of the payment consisting of the menu’s **ID**, **Name**, **Main Dish**, **Price** and the **Grand Total (Food Price \* Quantity)**.

****

**Figure 8. Receipt**

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