## Soal

*Case*

**The Executip**

**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 is 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.

**The Executip** is an exclusive dress store, selling the best dresses in town for all occasions. They sell two types of dresses: **regular** dresses and **limited-edition** dresses. This season, their **regular** dresses are on **discount**, unlike the **limited-edition** dresses. The designers working at **The Executip** have come up with so many designs that they need a list to keep track of all the dresses they have made. Unfortunately, they can’t code. You as good programmer, have been asked to **create a program** that can **list all the regular dress and limited-edition dresses** they have designed, based on the following criteria using **Java Programming Language**:

At the start of the application, it will show a **menu**. The menu consists of the following specifications :

1. **Insert new dress**
2. **View all dresses**
3. **Delete a dress**Background pattern

   Description automatically generated with low confidence

**Figure 1. home menu**

* If the user chooses **menu 1 (Insert new dress):** 
  + Validate the user input **must be 1 or 2**



**Figure 2. insert new dress**

* + If the user chooses **sub menu 1 (Regular Dress),** ask the user to input:
    - **Name**, which **length** must be between **5 to 20 characters (inclusive)**
    - **Fabric price**, which must be **a number** between **10000 to 100000 (inclusive)**
    - **Fabric type**, which must be either **‘Cotton’ or ‘Wool’ (case sensitive)**
    - **Discount,** the discount for the regular dress, which must be **a number** between **1 to 100 (inclusive)**
  + If the user chooses **sub menu 2 (Limited Edition Dress),** ask the user to input:
    - **Name**, which **length** must be between **5 to 20 characters (inclusive)**
    - **Fabric price**, the price of the fabric, which must be **a number** between **100000 to 300000 (inclusive)**
    - **Fabric type**, the type of fabric for the dress, which must be either **‘Satin’, ‘Chiffon’ or ‘Crepe’ (case sensitive)**
    - **Stock**, the stock of the dress, which must be **a number** between **1 to 10 (inclusive)**
  + If the user chooses **sub menu 3 (Back)**, the program will return to the **main menu**.
  + After the user **successfully** input all the data, **random an ID** for each dress, based on the following format:

|  |
| --- |
| **Regular Dress ID = ‘REXXX’ | X = random between 0-9 (Inclusive)**  **example: RE123** |
|  |
| **Limited Edition Dress ID = ‘LIXXX’ | X = random between 0-9 (Inclusive)**  **example: LI345** |
|  |

* + **Add** the data to **Array / Vector / ArrayList.**
  + Finally, **show success message**:
    - For Regular Dress: **“Successfully added a new Regular Dress!”**
    - For Limited Edition Dress: **“Successfully added a new Limited Edition Dress!”**

A picture containing text

Description automatically generated

**Figure 3. insert regular dress**

Text

Description automatically generated with medium confidence

**Figure 4. insert limited edition dress**

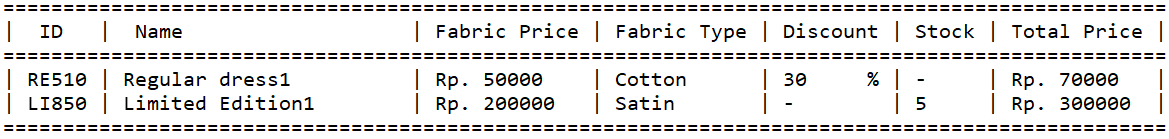
* If the user chooses **menu 2 (View all dresses):**
  + If **there is no dress data**, show the message **“No dresses available”,** then the program will return to **main menu.**



**Figure 5. no data message**

* + Otherwise, **show** the **list of all Regular Dresses and Limited Edition Dresses** along with their **total price,** which is calculated based on the following formula:

|  |
| --- |
| **Regular Dress =** (**Fabric Price** + 50000) \* ((100 – **discount**) / 100) |
|  |
| **Limited Edition Dress** = F**abric Price** + 50000 + ((10 – **stock**) \* 10000) |



**Figure 6. view all data**

* If the user chooses **menu 3 (Delete a dress):**
  + If **there is no dress data**, show the message **“No dresses available”,** then the program will return to the **main menu.**



**Figure 7. no data message**

* + Otherwise:
    - **Show** the **list of all Regular Dresses and Limited Edition Dresses,** along with their **total price**
    - Ask the user to input **Dress ID**, which must be **a valid ID**.
    - If the user inputs an **invalid ID,** the program will **display error message** and return to the **main menu.**

Diagram

Description automatically generated with medium confidence

**Figure 8. invalid ID error message**

* + - Otherwise, the program will **delete** the data according to the selected **ID**.

A picture containing table

Description automatically generated

**Figure 9. successfully delete dress**

* If the user chooses **menu 4 (Exit),** then the program will be closed.