|  |  |
| --- | --- |
| **Onsite Case** | image001 |
| JavaIH4 |
| **Periode Berlaku** Semester Ganjil 2015/2016  ***Valid on*** *Odd Year 2015/2016* | **Software Laboratory Center**  **Assistant Recruitment 16-1** |

## Materi

*Material*

## Bubble sort

## Selection sort

## Insertion sort

## Random

## Delay program

## Calendar

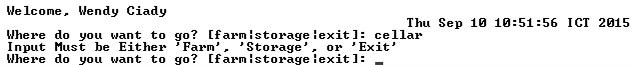
## Input/Output Dialog

## Locale dan Formatter

## Soal

*Case*

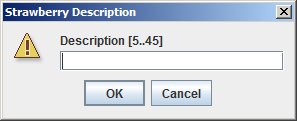
**Fruit Farmer**

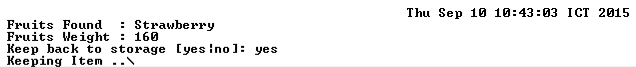
1. Create a program themed farming by using **Java Programming Language**. At the beginning, program will ask the user to input the user **name** with the following requirements:
   * Name must be **minimal two words**
   * Name must **be alphabet**
2. After the user input **name**, the program will show the **main menu** that consist :
   * **Welcoming message** showing the user **name**
   * Show **current date and time** based on the **time zone** where the program is running
   * **Farm** : Going to **fruit farm** to **collect fruit**
   * **Storage** : Going to **storage to view fruit list** that had been collected by the farmer at the fruit farm
   * **Exit** : **Close or exit** the program
   * All the input must be in **case insensitive**
   * After the user input correctly, there will be **shown a loading animation** before the program do any action
3. **Farm Menu** 
   * Program will show **current date and time** based on the **time zone** where the program is running
   * Program will **validate user can only choose “explore” or “back” (case insensitive)**
4. **Explore**

* Program will **random number to define what fruit that will be collected** by following requirement

|  |  |  |
| --- | --- | --- |
| **Random Result Number** | **Fruits Name** | **Weight** |
| **0** – **20** | **Apple** | Random number : **250** – **299** |
| **21** – **40** | **Orange** | Random number : **170** – **229** |
| **41** – **60** | **Grape** | Random number : **200** – **249** |
| **61** – **80** | **Strawberry** | Random number : **125** – **144** |
| **81** – **100** | **Lemon** | Random number : **70** – **139** |

* + - Program will ask user **whether want to keep fruit**. Program will **validate user can only input** “**y**” and “**n**” for confirmation
    - If user chooses to **keep the fruit**, program will **show an input dialog** to give description for the collected fruit. Program will **validate description must be between 5 and 45** character. Otherwise, program will **show error message in form of message box.**
    - If user **does not want to keep the food**, fruit **will not be stored in the storage**
    - If user has **input correctly**, program will **show a loading animation** before the program do any action

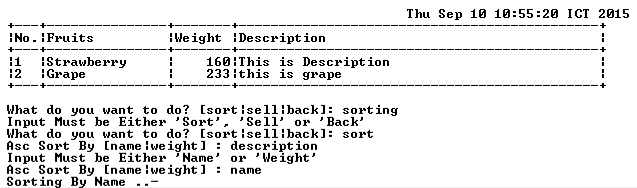




1. **If user choose “Back”** program will back to **main menu**
2. **Storage Menu**
   * Program will **validate whether there is fruit or not in the storage**. if there is **no fruits in the storage**, then the program **show message** “**Storage is empty**” and program will back to **main menu**
   * Show **current date and time** based on the **time zone** where the program is running
   * Program will **view all information about collected fruits**, such as **fruit name**, **weight** and **description**
   * Show all **collected fruits name, weight, and description**
   * Program will ask user to choose what to do with the list of collected fruit. Program will **validate** user can only do **sorting**, **selling** and **going back to main menu (case insensitive)**.
   * **Sort** : Do **ascending sort** by **fruit name** and **weight**. Program will

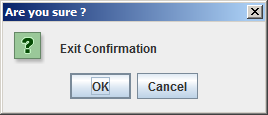
**validate user can only input** **name** and **weight** (**case insensitive**)

* + **Sell** : Selling fruit to the market, with the following condition
    - User **can only choose those listed fruit to sell**. Program will validate user can only **input listed fruit index**
    - If user **successfully select the fruit**, the program will **show message dialog** containing **information about the selected fruits** that had been sold successfully



* + **Back** : Going back to **main menu**
  + After the user input correctly, there will be **shown a loading animation** before the program do any action

1. If the user input “**Exit**”, the program will show a **confirmation dialog**.



**Please run the EXE file to see the sample program.**