Assessment: Lab exercise 2

Student Name: Yanzhang Wu

Lab Professor Name: *Fedor Ilitchev*

Lab Section Number: *CST8116\_323*

Due Date: June 17th 2022

# Understand the problem

This program will print several lines:

1. Input weight of the cabbage
2. In put cost of the cabbage
3. The return value of getWeight method
4. The return value the getCostPerKilogram method
5. The return value of the calculatePrice method (this method need some mathematical calculations, total price = weight \* cost per kilogram)
6. The output of printReport method
7. Testing overloaded constuctor with Cabbage2
8. The return value of the getWeight method and the getCostPerKilogram method
9. The output of printReport method
10. My name

Text, letter

Description automatically generated

Figure 1. The output of the console

To solve the problem, I will build two classes. The first class is class Cabbage. There will be get-set methods, calculatePrice() method in which price=unit price \* weight, and the printReport() method. The second class is class Exercise02 where the main() method locates.

# Develop and Describe an Algorithm

## UML Class Diagrams

Graphical user interface, text, application

Description automatically generated

Figure 2. UML diagram for class Cabbage

Table

Description automatically generated

Figure 3. UML diagram for class Exercise02

## Pseudocode for class Cabbage

Start

Declarations

// number containing decimal places

private double weight

private double costPerKilogramst

//no-argument constructor

Cabbage()

//overloaded Constructor for weight and costPerKilogram

Cabbage(double weight, double costPerKilogram)

// method getWeight()

public double getWeight()

return weight

//method setWeight()

Public void setWeight(double weight)

this.weight = weight

// method getCostPerKilogram()

public double getCostPerKilogram ()

return costPerKilogram

// method setCostPerKilogram()

Public double setCostPerKilogram (double costPerKilogramst)

this.costPerKilogramst = costPerKilogramst

//method calculatePrice

public double calculatePrice()

double price = weight \* costPerKilogram;

return price;

//method printReport()

printout(weight ”+” costPerKilogram ”+” price)

end

## Pseudocode for class Exercise02

Start

double inputWeight

double inputCost

// input weight should be a positive number

output "Please enter weight:"

enter inputWeight

if inputWeight is a positive number

break

else

output "Invalid input." and go back to output "Please enter weight:"

// input cost should be a positive number

output "Please enter cost:"

enter inputCost

if inputWeight is a positive number

break

else

output "Invalid input." and go back to output "Please enter cost:"

// non-arg constructor, set-get method

Cabbage cabbage1

cabbage1.setWeight(inputWeight)

cabbage1.setCostPerKilogram(inputCost)

double weight1 = cabbage1.getWeight()

double costPerKilogram1 = cabbage1.getCostPerKilogram()

double price1 = cabbage1.calculatePrice()

// check if methods work well

output "Input Weight = " + inputWeight

output "Input Cost = " + inputCost

output "cabbage1.getWeight() is: " + weight1

output "cabbage1.getCostPerKilogram() is: " + costPerKilogram1

output "cabbage.calculatePrice() is: " + price1

output "Cabbage1.printReport() is"

cabbage1.printReport()

// constructor with arguments, get method

output "Testing overloaded constructor with Cabbage2"

Cabbage cabbage2(5,10)

double weight2 = cabbage2.getWeight();

double getCostPerKilogram2 = cabbage2.getCostPerKilogram();

// check if methods work well

output "Weight " + weight2 + ", " + "cost per kilogram " + getCostPerKilogram2

output "Cabbage2.printReport() is"

cabbage2.printReport()

// output my name

output "Program by Yanzhang Wu"

end

## Flowchart for class Cabbage

A picture containing table, indoor

Description automatically generated

Figure 4. Flowchart for class Cabbage

## Flowchart for class Exercise02

Diagram

Description automatically generated

Figure 5. Flowchart for class Exercise02 part 1

Diagram

Description automatically generated

Figure 6. Flowchart for class Exercise02 part 2

# Test plan

|  |  |  |  |
| --- | --- | --- | --- |
| input | Expected output | Actual output | Description |
| 2.95  2.25 | Please enter weight:  2.95  Please enter cost:  2.25  Input Weight = 2.95  Input Cost = 2.25  cabbage1.getWeight() is: 2.95  cabbage1.getCostPerKilogram() is: 2.25  cabbage.calculatePrice() is: 6.6375  Cabbage1.printReport() is  Cabbage: weight: 2.95, cost per kilogram 2.25  Testing overloaded constructor with Cabbage2  Weight 5.0, cost per kilogram 10.0  Cabbage2.printReport() is  Cabbage: weight: 5.0, cost per kilogram 10.0  Program by Yanzhang Wu |  |  |
| String  or  negative number | Print out error messages and ask for input again |  |  |

# Source code

Text, application

Description automatically generated

Figure 7. Source code part 1-1

Text

Description automatically generated

Figure 8. Source code part 1-2

Graphical user interface, text, application

Description automatically generated

Figure 9. Source code part 1-3

Graphical user interface, text, application

Description automatically generated

Figure 10. Source code part 2-1

Graphical user interface, text, application

Description automatically generated

Figure 11. Source code part 2-2

Text, letter

Description automatically generated

Figure 12. Console output

# Test Plan for the Program

|  |  |  |  |
| --- | --- | --- | --- |
| input | Expected output | Actual output | Description |
| 2.95  2.25 | Please enter weight:  2.95  Please enter cost:  2.25  Input Weight = 2.95  Input Cost = 2.25  cabbage1.getWeight() is: 2.95  cabbage1.getCostPerKilogram() is: 2.25  cabbage.calculatePrice() is: 6.6375  Cabbage1.printReport() is  Cabbage: weight: 2.95, cost per kilogram 2.25  Testing overloaded constructor with Cabbage2  Weight 5.0, cost per kilogram 10.0  Cabbage2.printReport() is  Cabbage: weight: 5.0, cost per kilogram 10.0  Program by Yanzhang Wu | Please enter weight:  2.95  Please enter cost:  2.25  Input Weight = 2.95  Input Cost = 2.25  cabbage1.getWeight() is: 2.95  cabbage1.getCostPerKilogram() is: 2.25  cabbage.calculatePrice() is: 6.6375  Cabbage1.printReport() is  Cabbage: weight: 2.95, cost per kilogram 2.25  Testing overloaded constructor with Cabbage2  Weight 5.0, cost per kilogram 10.0  Cabbage2.printReport() is  Cabbage: weight: 5.0, cost per kilogram 10.0  Program by Yanzhang Wu | The actual output is identical to the expected output |
| asd  -5  (input of weight) | Invalid input.  Please enter weight: | Invalid input.  Please enter weight: | The actual output is identical to the expected output  (snapshot in next page) |
| qwe  0  (input of cost) | Invalid input.  Please enter cost: | Invalid input.  Please enter cost: | The actual output is identical to the expected output (snapshot in next page) |

Text, letter

Description automatically generated

Figure 13. Enter "asd" for weight

Text, letter

Description automatically generated

Figure 14. Enter "-5" for weight

Text

Description automatically generated

Figure 15. Enter "qwe" for cost

Text

Description automatically generated

Figure 16. Enter "0" for cost