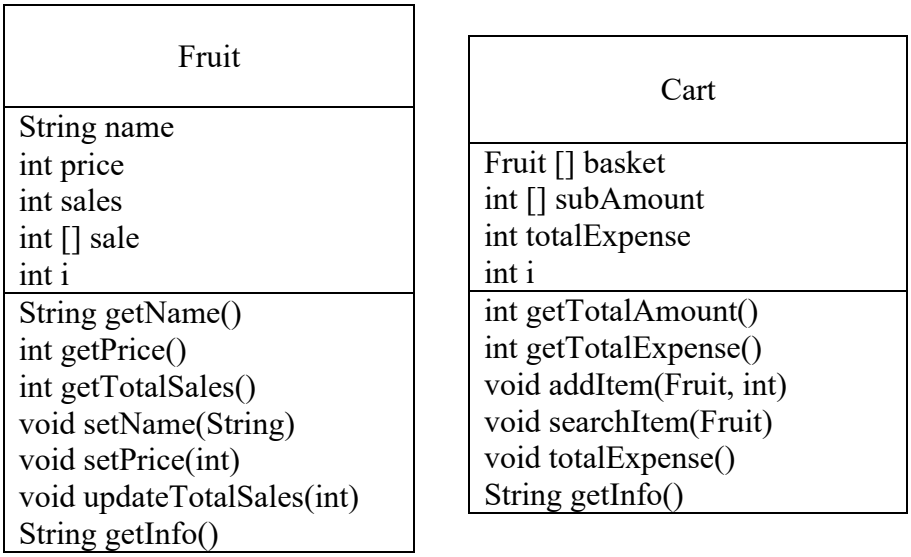
Lab 7

Requirements:

- Create a Java project named yourStudentId_OOP_Lab7
- Read instructions and create classes needed. You are supposed to add 3 classes (2 required + 1 Tester) to the project.
- Note that all instance variables are private. Please use public interfaces to access private variables.
- Your code must be properly formatted with sensible variable names! Refer to the text for code format examples.
- Please import the package you will use.
- The instruction for Tester and outputs is your reference.
- The following diagram describes two class you need to implement.



1. Create **Fruit** class

Fruit	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	name The name of the fruit.
int	price The price of the fruit.
int[]	sale The individual sale of the fruit.
int	totalSales The total sales of the fruit.
int	i The index of the sale array. To record the current index.
Constructor	

Fruit(String name, int price) Constructs a Fruit object with given name, price and an empty array of sale that can store 3 records. Meanwhile, initializes the totalSales, and i as 0.	
Instance methods	
-	3 setter for 3 attributes (getName(),getPrice(), and getTotalSales ()). 2 getter for 2 attributes (setName(...), setPrice(...)).
void	updateTotalSales(int amount) Accumulate all sales and update to the totalSales attribute. a. Use i variable to add the amount to the sale array. b. Update the i variable to the next index. c. Accumulate the amount into the totalSales attribute.
String	getInfo() a. Return the name, price, individual sale, and total sale of Fruit object. b. Individual sale should sort out the sale array from small to large. c. Use "for-each" concept to print out the content of sale array. d. You should follow the following formatted print out:

F	r	u	i	t		n	a	m	e	:			A	p	p	l	e
F	r	u	i	t		p	r	i	c	e	:		10				
I	n	d	i	v		s	a	l	e	s	:		5	,		20	,
T	o	t	a	l		s	a	l	e	s	:		25				

2. Create **Cart** class

Cart	
Modifier and type	**Method (or Variable) and description**
Instance variable	
Fruit[]	basket The shopping basket provides individual consumers to store the purchased products.
int[]	subAmount Record the purchase amount of each product of this consumer.
int	totalExpense Record all consumer spending on this shopping cart.
int	i The index of the sale array. To record the current index.
Constructor	
Cart() Initializes the basket, and subAmount array that can store 3 records. Meanwhile, sets the totalExpense, and i to 0.	

<table>
<tr><th colspan="2">Instance methods</th></tr>
<tr><td>-</td><td>1 getter for 1 attributes (getTotalExpense()).</td></tr>
<tr><td>void</td><td>addItem(Fruit fruit, int amount)
Store the products and quantities purchased by consumers in respective Arrays.
a. Use i variable to add the fruit to the basket array, and add the amount to the subAmount array, respectively.
b. Update the i variable to the next index.
c. Call the fruit’s updateTotalSales method to calculate the sales.</td></tr>
<tr><td>void</td><td>searchItem(Fruit fruit)
Determine whether the consumer buys the specified fruit.
a. Use "for-each" to determine whether the basket contains the fruit object
b. If it contains, print "Your basket has this product." If not, print "Your basket does not have this product."</td></tr>
<tr><td>void</td><td>totalExpense()
Calculate the expense of the user’s current shopping cart.
a. Use “for-loop” statement to call all the contents in the array.
b. Use “If-else” statement to determine whether the content stored in the basket array is not null, multiply the number of items purchased by the unit price and store the result in the totalExpense attribute.</td></tr>
<tr><td>String</td><td>getInfo()
Print out all costs and details as sample output.
a. Use “for-loop” concept to print out the content of basket and subAmount array.
b. You should follow the following formatted print out:
<table>
<tr><td colspan="22">The current expense is:NT$455</td></tr>
<tr><td>N</td><td>a</td><td>m</td><td>e</td><td></td><td></td><td></td><td>P</td><td>r</td><td>i</td><td>c</td><td>e</td><td>(</td><td>$</td><td>N</td><td>T</td><td>)</td><td></td><td></td><td>U</td><td>n</td><td>i</td><td>t</td></tr>
<tr><td>A</td><td>p</td><td>p</td><td>l</td><td>e</td><td></td><td>:</td><td></td><td></td><td></td><td>1</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td><td>2</td><td>0</td><td></td></tr>
<tr><td>B</td><td>a</td><td>n</td><td>a</td><td>n</td><td>a</td><td>:</td><td></td><td></td><td></td><td>1</td><td>2</td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td><td>1</td><td>5</td><td></td></tr>
<tr><td>O</td><td>r</td><td>a</td><td>n</td><td>g</td><td>e</td><td>:</td><td></td><td></td><td></td><td>1</td><td>5</td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td><td></td><td>5</td><td></td></tr>
</table>
</td></tr>
</table>

Tester	Output

public class Tester {
 	
  public static void main(String[] args) {
  // **TODO** Auto-generated method stub
  Fruit apple = **new** Fruit("Apple",10);
  Fruit banana = **new** Fruit("Banana",12);
  Fruit orange = **new** Fruit("Orange",15);
 	
  System.***out***.println("Shopping cart1 information:");
  Cart cart1 = **new** Cart();
 	
  cart1.searchItem(apple);
  cart1.addItem(apple, 20);
  cart1.searchItem(apple);
 	
  cart1.addItem(banana, 15);
  cart1.addItem(orange, 5);
  System.***out***.println();
  System.***out***.println(...);
 	
  System.***out***.println("Shopping cart2 information:");
 	
  Cart cart2 = **new** Cart();
  cart2.addItem(apple, 5);
  System.***out***.println(...);
 	
  System.***out***.println("Product Information:");
  System.***out***.println(...);
  }
 	
}
	Shopping cart1 information:
	Your basket does not have this product.
	Your basket has this product.
	
	The current expense is:NT$455
	Name Price($NT) Unit
	Apple : 10 * 20
	Banana: 12 * 15
	Orange: 15 * 5
	
	Shopping cart2 information:
	The current expense is:NT$50
	Name Price($NT) Unit
	Apple : 10 * 5
	
	Product Information:
	Fruit name: Apple
	Fruit price: 10
	Indiv sales: 5, 20,
	Total sales: 25

Submission: Submit your project as **".zip file"** via Moodle. No other submissions will be graded.

Reminder: Please zip the whole project

Deadline: Tomorrow's midnight (for both Mon56 and Tue23)