Lesson 3b

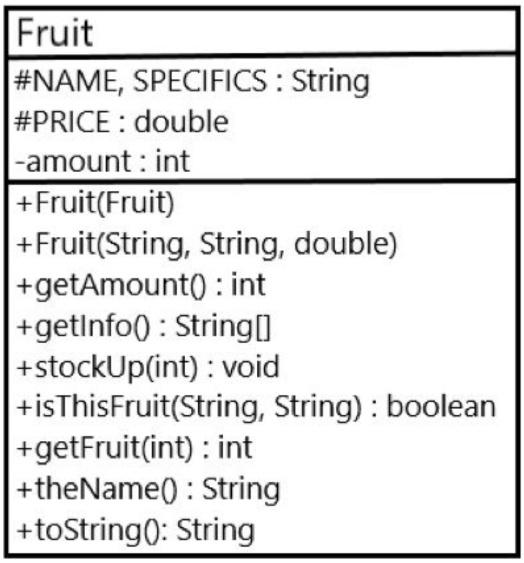
Mark Eric Dieckmann

February 6, 2023

Lesson 3a deals with hierarchical inheritance. We implement a framework that allows users to set up products that are sold in a supermarket. The products are strawberries and apples. We practice casting of classes and we practice how we can store different types of information in the superclass and subclass.

1 Task 1: The superclass Fruit

Fruit has two purposes. (1) We use it as a standalone class to create instances of it as types of fruits sold by the supermarket. (2) We use it as a superclass when we create instances of apples and strawberries that get sold to customers.



Fruit

#NAME, SPECIFICS : String
#PRICE : double
-amount : int

+Fruit(Fruit)
+Fruit(String, String, double)
+getAmount() : int
+getInfo() : String[]
+stockUp(int) : void
+isThisFruit(String, String) : boolean
+getFruit(int) : int
+theName() : String
+toString(): String

NAME, *SPECIFICS*, and *PRICE* are constants. They are initialized in the constructor.

amount in the superclass stores the number of items in stock.

Fruit(arg) initializes the values of the three constants with the values of *arg*.

Fruit(arg1, arg2, arg3) initializes the constants with the three arguments and sets *amount* to zero.

getAmount() returns *amount*.

getInfo() returns *NAME* and *SPECIFICS* in a array with two elements.

stockUp(arg) increases the value of *amount* by *arg*.

isThisFruit(arg1, arg2) compares the values of *NAME* and *SPECIFICS* of the calling instance with the arguments. It checks if two instances of **FRUIT** are the same.

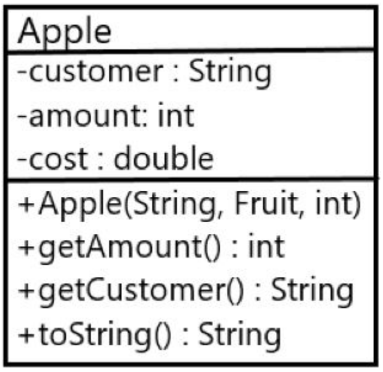
getFruit(arg) checks if *arg* $\leq$ *amount*. If it is, it reduces *amount* by *arg* and sends *arg* back. If it is not, it sends *amount* back and sets it to zero.

theName() sends back ”Apple” or ”Strawberry” if the object is an instance of a subclass. It sends back ”Fruit” otherwise.

toString() returns a formatted string ”Name: ” followed by *NAME* on 8 characters followed by ”Specifics: ” followed by *SPECIFICS* on 8 characters followed by ”, Price: ” and *PRICE* on 5 characters (2 after the ”,”) followed by ”, Amount: ” followed by *AMOUNT:* on 3 characters (integer).

2 Task 2: The subclasses Apple/Strawberry

Fruit has the subclasses *Apple* and **Strawberry**. Both have the same variables and methods so we discuss only **Apple**.



Apple

-customer : String
-amount: int
-cost : double

+Apple(String, Fruit, int)
+getAmount() : int
+getCustomer() : String
+toString() : String

customer stores the name of the customer that bought the apples.

amount stores the number of apples the customer bought.

cost stores the money the customer paid ($cost = amount \cdot PRICE$).

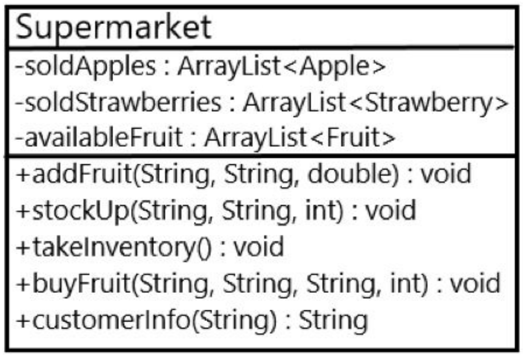
Apple(arg1, arg2, arg3) initializes the superclass with the values of *arg2*, *customer* with *arg1* and *amount* with the return value of the superclasses *getFruit(arg3)*.

getCustomer() and *getAmount()* do what their names say.

toString() returns the formatted string ”Customer ” followed by *customer* on 6 characters followed by ” bought ” followed by the int value of *amount* on three characters followed by ” apples for ” followed by *cost* on 5 characters (2 after the ”,”) followed by ” SEK”.

3 Task 3: The class Supermarket

Supermarket has methods that can stock up on fruits and sell them to customers. It has dynamic arrays that list what the supermarket sells and what it sold to customers.



Supermarket

-soldApples : ArrayList<Apple>
-soldStrawberries : ArrayList<Strawberry>
-availableFruit : ArrayList<Fruit>

+addFruit(String, String, double) : void
+stockUp(String, String, int) : void
+takeInventory() : void
+buyFruit(String, String, String, int) : void
+customerInfo(String) : String

soldApples and *soldStrawberries* list all purchases made by customers.

availableFruit stores all fruits sold by the supermarket.

addFruit(String, String, double) initializes an instance of **Fruit** with its arguments and adds it to *availableFruit*.

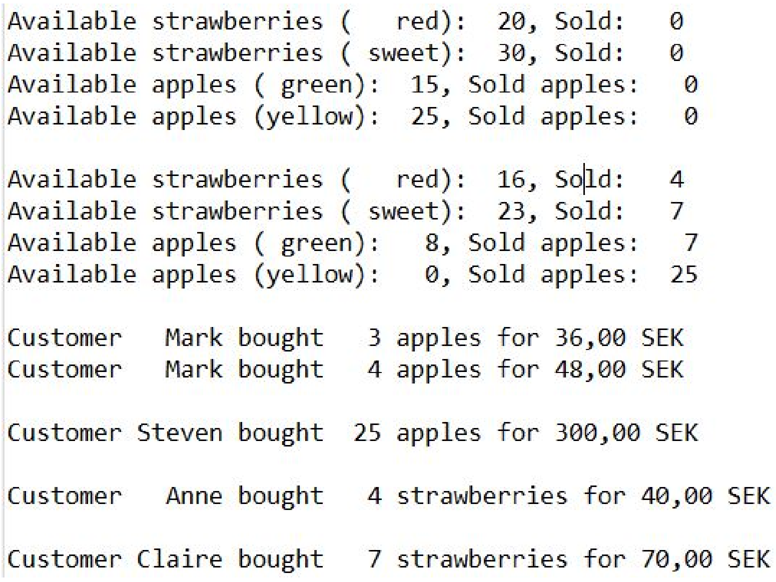
stockUp(arg1, arg2, arg3) increases by *arg3* the value of *amount* of the fruit with name *arg1* and specifics *arg2*.

takeInventory() takes the inventory of all elements of *availableFruit*. It checks how many it has left and how many it sold. It writes a formatted string directly to the console. The string starts with ”Available apples ” followed by *SPECIFICS* on 6 characters and enclosed by (). It is followed by ”: ” and the value of *amount* of the superclass on 3 characters. This is followed by ”, Sold apples: ” followed by the value of *amount* of the subclass (here Apple) on 3 characters.

buyFruit(arg1, arg2, arg3, arg4) checks if there is a fruit with the name *arg1* and specifics *arg2*. It then creates an instance of **Apple** or **Strawberry**, initializes the customer with *arg3* and buys *arg4* fruit (if available in stock). It puts the instance into the appropriate dynamic array.

customerInfo(arg) finds all purchases made by the customer with the name *arg*. It calls the *toString()* methods of the apples and strawberries and concatenates them to a string. Each result from the *toString()* method is written on one line.

The class **Lesson3b** is fully implemented and should give the console output:



```
Available strawberries (   red):  20, Sold:   0
Available strawberries ( sweet):  30, Sold:   0
Available apples ( green):  15, Sold apples:   0
Available apples (yellow):  25, Sold apples:   0

Available strawberries (   red):  16, Sold:   4
Available strawberries ( sweet):  23, Sold:   7
Available apples ( green):   8, Sold apples:   7
Available apples (yellow):   0, Sold apples:  25

Customer   Mark bought   3 apples for 36,00 SEK
Customer   Mark bought   4 apples for 48,00 SEK

Customer Steven bought  25 apples for 300,00 SEK

Customer   Anne bought   4 strawberries for 40,00 SEK

Customer Claire bought   7 strawberries for 70,00 SEK
```