

COMP 201 Data Structures and Algorithms Lab 4 - Generic Box Using Array List

Alice and Her Boxes



Alice has two boxes of chocolate and one box of pens.

Lab 4 - Tasks

Task 1:

- There are two types of boxes: one for chocolate and one for pens.
- Write a generic Box class which can store either chocolate or pens (not both at the same time).
 - o Internally, the Box class has a generic array list to store either chocolate or pens.
 - o The concrete type (Chocolate or Pen) stored in a Box object is specified when that Box object is created.
 - Box is basically a <u>last-in</u>, <u>first-out</u> data structure similar to a <u>stack</u>. Assuming that a Box object for storing Chocolate objects is created using Box<Chocolate> myBox = new Box<>0;
 - You can add a Chocolate object to the top of this Box object using myBox.add(chocolate1).
 - You can remove the Chocolate object at the top of this Box object using myBox.remove().
- Write Chocolate and Pen classes as well.
- <u>UML diagrams</u> of all 3 classes are shown on <u>the next slide</u>.

Task 2: Extra (Ungraded) Exercise

- A Person can have <u>multiple boxes</u>, such as two chocolate boxes and a pen box.
- Internally, the **Person class** stores these boxes in <u>a single array list</u>.
- See the <u>UML diagram</u> given on <u>slide 5</u>.

UML Class Diagrams for Task 1

Chocolate

+name: String +type: String

+Chocolate(name: String, type: String)

+toString(): String

Chocolate brand name such as Lindt, Godiva and Milka

Chocolate type such as bitter, milk and white

Constructor

Returns a string representation of the object such as: (N: Godiva, T: Milk)

Pen

+name: String
+type: String

+Pen(name: String, type: String)

+toString(): String

Pen brand name such as Lamy, Pilot and Faber Pen type such as ballpoint, gel and technical

Constructor

Returns a string representation of the object such as: (N: Lamy, T: Ballpoint)

Box<E>

+contents: ArrayList<E>

+Box()

+add(item: E): void

+remove(): E

+toString(): String

ArrayList to store items of generic type E

Constructor

Adds an item to the end of the list

Removes and returns the item at the end of the list

Returns box content information as a string •

Example

(N: Lindt, T: Bitter) (N: Milka, T: Milk)

(N: Godiva, T: Bitter)

(N: Lindt, T: White)

(N: Cadbury, T: Bitter)

(N: Cadbury, T: Milk)

Tips:

- Remember: An ArrayList of String objects is created as ArrayList<String> myArrayList = new ArrayList<>();
- How can you create a generic ArrayList of E objects (E is the generic type)?
 ArrayList<?> contents = new ArrayList<>();

Additional UML Class Diagram for Task 2

Person

+name: String

+boxes: ArrayList<Object>

+Person(name: String) +addBox(box: Object): void

+printPerson(): void

Person name

Array list to store multiple boxes with different types

Constructor

Adds the given box to the boxes array list

Prints person information: name and contents of all boxes •

Example

Person Information: Alice has 3 boxes.

(N: Lindt, T: Bitter)

(N: Milka, T: Milk)

(N: Godiva, T: Bitter)

(N: Lindt, T: White)

(N: Cadbury, T: Bitter)

(N: Cadbury, T: Milk)

(N: Lamy, T: Technical)

(N: Lamy, T: Ballpoint)

(N: Faber, T: Colored)

(N: Guylian, T: Bitter)

(N: Guylian, T: Milk)

(N: Guylian, T: White)

Hint:

- For Alice, how can you store <u>multiple different boxes in a single ArrayList</u> structure?
 - o Think about the **Object class:** superclass of all the classes in Java.

```
public class TestApp { // Test program for Lab 4 - Generic Box Using Array List
  public static void main(String[] args) {
     // creating the first box for storing Chocolate objects, adding objects to
     // the box and printing information about the contents of the box
     Box<Chocolate> myBox1 = new Box<>(); ●
     myBox1.add(new Chocolate("Lindt", "Bitter"));
     myBox1.add(new Chocolate("Milka", "Milk"));
     myBox1.add(new Chocolate("Godiva", "Bitter"));
     myBox1.add(new Chocolate("Lindt", "White"));
     myBox1.add(new Chocolate("Cadbury", "Bitter"));
     myBox1.add(new Chocolate("Cadbury", "Milk"));
     System.out.println("Box 1 Contents After Add Operations:");
     System.out.println(myBox1); // using toString method of the Box class
     // removing 3 objects from the end of myBox1 then printing myBox1
     for (int i = 0; i < 3; i++) myBox1.remove();</pre>
     System.out.println("Box 1 Contents After Remove Operations:");
     System.out.println(myBox1); // using toString method of the Box class
     // creating the second box for storing Pen objects, adding objects to
     // the box and printing information about the contents of the box
     myBox2.add(new Pen("Lamy", "Technical"));
     myBox2.add(new Pen("Lamy", "Ballpoint"));
     myBox2.add(new Pen("Faber", "Colored"));
     System.out.println("Box 2 Contents After Add Operations:");
     System.out.println(myBox2); // using toString method of the Box class
     // creating the third box for storing Chocolate objects, adding objects to
     // the box and printing information about the contents of the box
     Box<Chocolate> myBox3 = new Box<>();
     myBox3.add(new Chocolate("Guylian", "Bitter"));
     myBox3.add(new Chocolate("Guylian", "Milk"));
     myBox3.add(new Chocolate("Guylian", "White"));
     System.out.println("Box 3 Contents After Add Operations:");
     System.out.println(myBox3); // using toString method of the Box class
     // creating a Person object named Alice, adding her boxes and printing info
     // Person p = new Person("Alice");
     // p.addBox(myBox1);
     // p.addBox(myBox2);
     // p.addBox(myBox3);
     // p.printPerson();
```

Main Application for Task 1

Generic box creation

```
Console Output of the Program
Box 1 Contents After Add Operations:
```

```
(N: Cadbury, T: Milk)
(N: Cadbury, T: Bitter)
(N: Lindt, T: White)
(N: Godiva, T: Bitter)
(N: Milka, T: Milk)
(N: Lindt, T: Bitter)
Box 1 Contents After Remove Operations:
(N: Godiva, T: Bitter)
(N: Milka, T: Milk)
(N: Lindt, T: Bitter)
Box 2 Contents After Add Operations:
(N: Faber, T: Colored)
(N: Lamy, T: Ballpoint)
(N: Lamy, T: Technical)
Box 3 Contents After Add Operations:
(N: Guylian, T: White)
(N: Guylian, T: Milk)
(N: Guylian, T: Bitter)
```



Note: The box contents are printed from the last item to the first item (like a stack).

```
public class TestApp { // Test program for Lab 4 - Generic Box Using Array List
   public static void main(String[] args) {
      // creating the first box for storing Chocolate objects, adding objects to
     // the box and printing information about the contents of the box
      Box<Chocolate> myBox1 = new Box<>();
      myBox1.add(new Chocolate("Lindt", "Bitter"));
      myBox1.add(new Chocolate("Milka", "Milk"));
      myBox1.add(new Chocolate("Godiva", "Bitter"));
      myBox1.add(new Chocolate("Lindt", "White"));
      myBox1.add(new Chocolate("Cadbury", "Bitter"));
      myBox1.add(new Chocolate("Cadbury", "Milk"));
     System.out.println("Box 1 Contents After Add Operations:");
     System.out.println(myBox1); // using toString method of the Box class
     // removing 3 objects from the end of myBox1 then printing myBox1
      for (int i = 0; i < 3; i++) myBox1.remove();
     System.out.println("Box 1 Contents After Remove Operations:");
     System.out.println(myBox1); // using toString method of the Box class
      // creating the second box for storing Pen objects, adding objects to
     // the box and printing information about the contents of the box
      Box<Pen> myBox2 = new Box<>();
     myBox2.add(new Pen("Lamy", "Technical"));
      myBox2.add(new Pen("Lamy", "Ballpoint"));
      myBox2.add(new Pen("Faber", "Colored"));
     System.out.println("Box 2 Contents After Add Operations:");
     System.out.println(myBox2); // using toString method of the Box class
      // creating the third box for storing Chocolate objects, adding objects to
     // the box and printing information about the contents of the box
      Box<Chocolate> myBox3 = new Box<>();
      myBox3.add(new Chocolate("Guylian", "Bitter"));
      myBox3.add(new Chocolate("Guylian", "Milk"));
      myBox3.add(new Chocolate("Guylian", "White"));
     System.out.println("Box 3 Contents After Add Operations:");
     System.out.println(myBox3); // using toString method of the Box class
     // creating a Person object named Alice, adding her boxes and printing info
     // Person p = new Person("Alice");
     // p.addBox(myBox1);
                                                  Uncomment these lines in
     // p.addBox(myBox2);
                                                   TestApp.java
     // p.addBox(myBox3);
      // p.printPerson();
```

Main Application for Task 2

```
public class TestApp { // Test program for Lab 4 - Generic Box Using Array List
                                                                                  Console Output of the Program:
  public static void main(String[] args) {
                                                                                  Box 1 Contents After Add Operations:
     // creating the first box for storing Chocolate objects, adding objects to
                                                                                  (N: Cadbury, T: Milk)
     // the box and printing information about the contents of the box
                                                                                  (N: Cadbury, T: Bitter)
     Box<Chocolate> myBox1 = new Box<>();
     myBox1.add(new Chocolate("Lindt", "Bitter"));
                                                                                  (N: Lindt, T: White)
     myBox1.add(new Chocolate("Milka", "Milk"));
                                                                                  (N: Godiva, T: Bitter)
     myBox1.add(new Chocolate("Godiva", "Bitter"));
                                                                                  (N: Milka, T: Milk)
     myBox1.add(new Chocolate("Lindt", "White"));
                                                                                  (N: Lindt, T: Bitter)
     myBox1.add(new Chocolate("Cadbury", "Bitter"));
     myBox1.add(new Chocolate("Cadbury", "Milk"));
                                                                                  Box 1 Contents After Remove Operations:
     System.out.println("Box 1 Contents After Add Operations:");
                                                                                  (N: Godiva, T: Bitter)
     System.out.println(myBox1); // using toString method of the Box class
                                                                                  (N: Milka, T: Milk)
     // removing 3 objects from the end of myBox1 then printing myBox1
     for (int i = 0; i < 3; i++) myBox1.remove();
                                                                                  (N: Lindt, T: Bitter)
     System.out.println("Box 1 Contents After Remove Operations:");
     System.out.println(myBox1); // using toString method of the Box class
                                                                                  Box 2 Contents After Add Operations:
                                                                                  (N: Faber, T: Colored)
     // creating the second box for storing Pen objects, adding objects to
                                                                                  (N: Lamy, T: Ballpoint)
     // the box and printing information about the contents of the box
                                                                                  (N: Lamy, T: Technical)
     Box < Pen > myBox2 = new Box <>();
     myBox2.add(new Pen("Lamy", "Technical"));
     myBox2.add(new Pen("Lamy", "Ballpoint"));
                                                                                  Box 3 Contents After Add Operations:
     myBox2.add(new Pen("Faber", "Colored"));
                                                                                  (N: Guylian, T: White)
     System.out.println("Box 2 Contents After Add Operations:");
                                                                                  (N: Guylian, T: Milk)
     System.out.println(myBox2); // using toString method of the Box class
                                                                                  (N: Guylian, T: Bitter)
     // creating the third box for storing Chocolate objects, adding objects to
     // the box and printing information about the contents of the box
                                                                                  Person Information:
     Box<Chocolate> myBox3 = new Box<>();
                                                                                  Alice has 3 boxes.
     myBox3.add(new Chocolate("Guylian", "Bitter"));
     myBox3.add(new Chocolate("Guylian", "Milk"));
                                                                                  (N: Godiva, T: Bitter)
     myBox3.add(new Chocolate("Guylian", "White"));
                                                                                  (N: Milka, T: Milk)
     System.out.println("Box 3 Contents After Add Operations:");
                                                                                  (N: Lindt, T: Bitter)
     System.out.println(myBox3); // using toString method of the Box class
     // creating a Person object named Alice, adding her boxes and printing info
                                                                                  (N: Faber, T: Colored)
     Person p = new Person("Alice");
                                                                                  (N: Lamy, T: Ballpoint)
     p.addBox(myBox1);
                                                                                  (N: Lamy, T: Technical)
     p.addBox(myBox2);
     p.addBox(myBox3);
                                                                                  (N: Guylian, T: White)
     p.printPerson();
                                                                                  (N: Guylian, T: Milk)
                                                                                  (N: Guylian, T: Bitter)
```

Submission of Lab Work

- Submit your Java code file(s) (no report required) to Blackboard.
- You can work as a team of 2 students, if you like. <u>Each team member should upload</u> his/her code to Blackboard individually.

Grading

Your lab work will be graded on a scale of: 0: Incorrect/NA, 1: Partially correct, 2: Correct