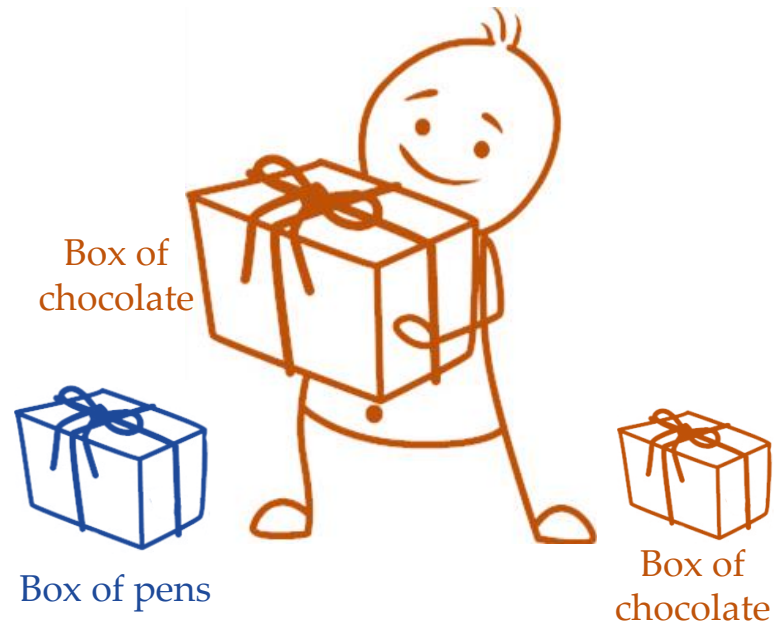




COMP 201 Data Structures and Algorithms

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**).
 - Internally, the Box class has **a generic array list** to store either chocolate or pens.
 - The **concrete type** (**Chocolate** or **Pen**) stored in a Box object is specified when that Box object is created.
 - Box is basically a last-in, first-out data structure similar to a stack. Assuming that a Box object for storing Chocolate objects is created using `Box<Chocolate> myBox = new Box<>();`
 - 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.
- UML diagrams of all 3 classes are shown on the next slide.

Task 2: **Extra (Ungraded) Exercise**

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

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

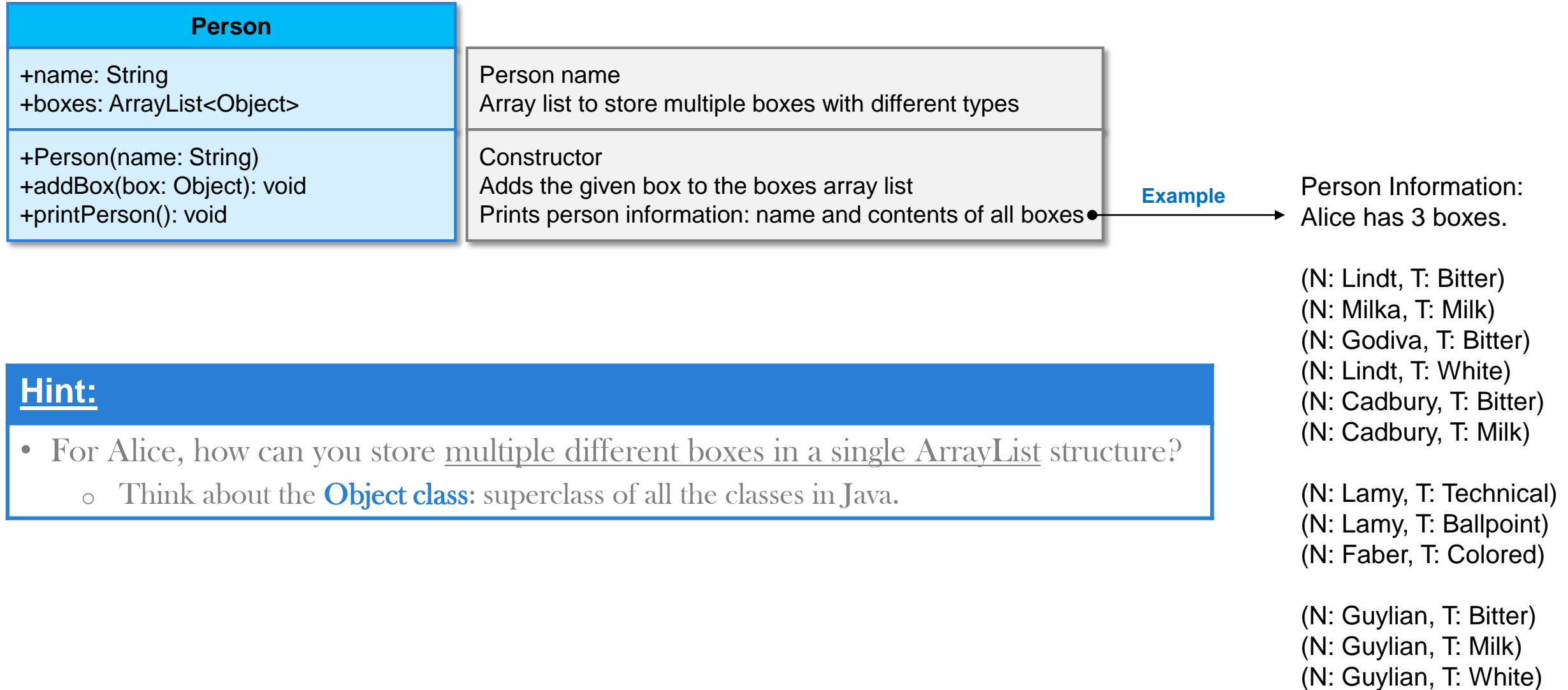
Example

(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



Main Application for Task 1

```
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);
        // p.addBox(myBox2);
        // p.addBox(myBox3);
        // p.printPerson();
    }
}
```

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).

Main Application for Task 2

```
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);
        // p.addBox(myBox2);
        // p.addBox(myBox3);
        // p.printPerson();
    }
}
```



Uncomment these lines in
TestApp.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();
        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);
        p.addBox(myBox2);
        p.addBox(myBox3);
        p.printPerson();
    }
}

```

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)

Person Information:

Alice has 3 boxes.

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

(N: Faber, T: Colored)
 (N: Lamy, T: Ballpoint)
 (N: Lamy, T: Technical)

(N: Guylian, T: White)
 (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. Each team member should upload 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