# Creating a simple to-do list using ArrayLists

**Estimated time needed:** 20 minutes

In this lab, you will learn how to handle ArrayLists by creating a `To-do` ArrayList and providing it features to prioritize tasks and sort them by priority.

You are currently viewing this lab in a Cloud-based Integrated Development Environment (Cloud IDE). This fully online, integrated development environment is pre-installed with JDK 21 and allows you to code, develop, and learn in one location.

## Learning Objectives

After completing this lab, you will be able to:

- Understand how to work with ArrayLists
- Add entries to the ArrayList
- Manipulate the entries in an ArrayList
- Sort the entries in an ArrayList
- Filter the entries in an ArrayList

# ArrayList

An Array is a fixed-size, ordered collection of elements of the same type. For example `int[] iArr = new int[5]` is an `int` array of size 5. An ArrayList is a class that helps to create a collection that is resizable and orderable and is part of the Java Collections Framework. In this lab, you will create a `Task` class and add objects of that to the ArrayList.

1. Create a project directory by running the following command.

   ```
   mkdir my_arraylist_proj
   ```

2. Run the following code to create the directory structure.

   ```
   mkdir -p my_arraylist_proj/src
   mkdir -p my_arraylist_proj/classes
   mkdir -p my_arraylist_proj/test
   cd my_arraylist_proj
   ```

3. Now create a file named `Task.java` inside the src directory.

   ```
   touch /home/project/my_arraylist_proj/src/Task.java
   ```

4. Click the following button to open the file for editing.

Open **Task.java** in IDE

5. Copy and paste the code in `Task.java`.

```
public class Task {
    public static int HIGH = 3;
    public static int MEDIUM = 2;
    public static int LOW = 1;
    public static String COMPLETED = "Completed";
    public static String IN_PROGRESS = "In Progress";
    public static String NEW = "New";
    private String taskName;
    private int priority = LOW;
    private String status = NEW;
    public Task(String taskName) {
        this.taskName = taskName;
    }
    public Task(String taskName, int priority) {
        this.taskName = taskName;
        this.priority = priority;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public void setPriority(int priority) {
        this.priority = priority;
    }
    public String getStatus() {
        return this.status;
    }
    public int getPriority() {
        return this.priority;
    }
    public String toString() {
        return taskName.concat(" ").concat((priority+"")).concat(" ").concat(status);
    }
}
```

6. Compile the Java program, specifying the destination directory as the `classes` directory that you created.

```
javac -d classes src/Task.java
```

7. Set the `CLASSPATH` variable.

```
export CLASSPATH=$CLASSPATH:/home/project/my_arraylist_proj/classes
```

8. Now create a file named `ArrayListExample.java` inside the src directory.

```
touch /home/project/my_arraylist_proj/src/ArrayListExample.java
```

9. Click the following button to open the file for editing.

Open **ArrayListExample.java** in IDE

10. Read each statement in the following code carefully. Copy and paste it into `ArrayListExample.java`.

```
import java.util.Scanner;
import java.util.ArrayList;
public class ArrayListExample {
    public static void main(String s[]) {
        try {
            // Create a Scanner object to read user input
            Scanner scanner = new Scanner(System.in);
            // Create an ArrayList to store Task objects (to-do list)
            ArrayList<Task> todoList = new ArrayList<Task>();
            // Infinite loop to keep the program running until the user chooses to exit
            while (true) {
                // Display the menu options to the user
                System.out.println(
                    "Press 1 to add a task, " +
                    "\n2 to view all the tasks " +
                    "\n3 to change status of tasks " +
                    "\n4 to delete a task " +
                    "\nAny other key to exit");
                // Read the user's choice
                String userAction = scanner.nextLine();
                // Option 1: Add a task to the to-do list
                if (userAction.equals("1")) {
                    // Prompt the user to enter a task description
                    System.out.println("Enter the task ");
                    String taskStr = scanner.nextLine();
                    // Prompt the user to enter the priority of the task
                    System.out.println("Enter Priority - 1 Low, 2 Medium, 3 High ");
                    int priority = Integer.parseInt(scanner.nextLine());
                    // Validate the priority input (default to 1 if invalid)
                    priority = priority > 3 ? 1 : priority;
                    // Create a new Task object and add it to the to-do list
                    todoList.add(new Task(taskStr, priority));
                    System.out.println("The task has been added to the list");
                }
                // Option 2: View all tasks in the to-do list
                else if (userAction.equals("2")) {
                    // Use forEach to print each task in the to-do list
                    todoList.forEach(task -> System.out.println(task));
                }
                // Option 3: Change the status of a task
                else if (userAction.equals("3")) {
                    // Prompt the user to enter the index of the task to update
                    System.out.println("Enter the index of the status you want to change ");
                    int chgIdx = Integer.parseInt(scanner.nextLine());
                    // Check if the index is valid
                    if (chgIdx > (todoList.size() - 1)) {
                        System.out.println("There is no such index position in the list");
                    } else {
                        // Prompt the user to enter the new status for the task
                        System.out.println("Enter the new status for the task P for 'In Progress' or C for 'Completed'");
                        String updatedStatus = scanner.nextLine();
                        // Update the task status based on user input
                        if (updatedStatus.equalsIgnoreCase("P")) {
                            todoList.get(chgIdx).setStatus(Task.IN_PROGRESS);
                        } else if (updatedStatus.equalsIgnoreCase("C")) {
                            todoList.get(chgIdx).setStatus(Task.COMPLETED);
                        }
                    }
                    System.out.println("The task has been changed in the list");
                }
                // Option 4: Delete a task
                else if (userAction.equals("4")) {
                    // Prompt the user to enter the index of the task to delete
                    System.out.println("Enter the index of the status you want to delete ");
                    int rmvIdx = Integer.parseInt(scanner.nextLine());
                    // Check if the index is valid
                    if (rmvIdx > (todoList.size() - 1)) {
                        System.out.println("There is no such index position in the list");
                    } else {
                        todoList.remove(rmvIdx);
                        System.out.println("The task has been removed from the list");
                    }
                }
                // Exit the program if the user enters any other key
                else {
                    break;
                }
            }
        } catch (NumberFormatException nfe) {
            // Handle invalid number input (e.g., non-integer input for priority or index)
            System.out.println("Invalid input. Please enter a valid number.");
        }
    }
}
```

The program displays a menu with options to:

- Add a task - The user is prompted to enter a task description and priority. The priority is validated to ensure it is within the range (1, 2, or 3). A new Task object is created and added to the todoList.
- View all tasks - The forEach method is used to iterate over the todoList and print each task.
- Change the status of a task - The user is prompted to enter the index of the task they want to update. The program checks if the index is valid. The user is prompted to enter the new status (P for "In Progress" or C for "Completed"). The status of the selected task is updated.
- Delete a task - The user is prompted to enter the index of the task they want to delete. The program checks if the index is valid. The task is deleted.
- Exit the program.

11. Compile the Java program, specifying the destination directory as the `classes` directory that you created.

    ```
    javac -d classes src/ArrayListExample.java
    ```

12. Run the program and test with variable combinations.

    ```
    java ArrayListExample
    ```

A sample output would look like

```
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
1
Enter the task
laundry
Enter Priority - 1 Low, 2 Medium, 3 High
3
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
2
laundry 3 New
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
3
Enter the index of the status you want to change
0
Enter the new status for the task P for 'In Progress' or C for 'Completed'
C
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
4
Enter the index of the status you want to delete
0
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
2
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
```

# Sort and Filter

Collection class provides a sort method which takes a `Comparator` and compares two objects. In this case, you will compare the priority of tasks and sort the tasks from `High` to `Low`.

1. Click the following button to open the file for editing, if it is not already open.

Open **ArrayListExample.java** in IDE

2. Replace the existing code with the following code.

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
public class ArrayListExample {
    public static void main(String s[]) {
        try {
            // Create a Scanner object to read user input
            Scanner scanner = new Scanner(System.in);
            // Create an ArrayList to store Task objects (to-do list)
            ArrayList<Task> todoList = new ArrayList<Task>();
            // Infinite loop to keep the program running until the user chooses to exit
            while (true) {
```

```java
                // Display the menu options to the user
                System.out.println(
                    "Press 1 to add a task, " +
                    "\n2 to view all the tasks " +
                    "\n3 to change status of tasks " +
                    "\n4 to delete a task " +
                    "\n5 to sort the list based on priority " +
                    "\nAny other key to exit");
                // Read the user's choice
                String userAction = scanner.nextLine();
                // Option 1: Add a task to the to-do list
                if (userAction.equals("1")) {
                    // Prompt the user to enter a task description
                    System.out.println("Enter the task ");
                    String taskStr = scanner.nextLine();
                    // Prompt the user to enter the priority of the task
                    System.out.println("Enter Priority - 1 Low, 2 Medium, 3 High ");
                    int priority = Integer.parseInt(scanner.nextLine());
                    // Validate the priority input (default to 1 if invalid)
                    priority = priority > 3 ? 1 : priority;
                    // Create a new Task object and add it to the to-do list
                    todoList.add(new Task(taskStr, priority));
                    System.out.println("The task has been added to the list");
                }
                // Option 2: View all tasks in the to-do list
                else if (userAction.equals("2")) {
                    // Use forEach to print each task in the to-do list
                    todoList.forEach(task -> System.out.println(task));
                }
                // Option 3: Change the status of a task
                else if (userAction.equals("3")) {
                    // Prompt the user to enter the index of the task to update
                    System.out.println("Enter the index of the status you want to change ");
                    int chgIdx = Integer.parseInt(scanner.nextLine());
                    // Check if the index is valid
                    if (chgIdx > (todoList.size() - 1)) {
                        System.out.println("There is no such index position in the list");
                    } else {
                        // Prompt the user to enter the new status for the task
                        System.out.println("Enter the new status for the task P for 'In Progress' or C for 'Completed'");
                        String updatedStatus = scanner.nextLine();
                        // Update the task status based on user input
                        if (updatedStatus.equalsIgnoreCase("P")) {
                            todoList.get(chgIdx).setStatus(Task.IN_PROGRESS);
                        } else if (updatedStatus.equalsIgnoreCase("C")) {
                            todoList.get(chgIdx).setStatus(Task.COMPLETED);
                        }
                    }
                    System.out.println("The task has been changed in the list");
                }
                // Option 4: Delete a task
                else if (userAction.equals("4")) {
                    // Prompt the user to enter the index of the task to delete
                    System.out.println("Enter the index of the status you want to delete ");
                    int rmvIdx = Integer.parseInt(scanner.nextLine());
                    // Check if the index is valid
                    if (rmvIdx > (todoList.size() - 1)) {
                        System.out.println("There is no such index position in the list");
                    } else {
                        todoList.remove(rmvIdx);
                        System.out.println("The task has been removed from the list");
                    }
                }
                // Option 5: Sort tasks by priority
                else if (userAction.equals("5")) {
                    // Sort the ArrayList by age using a Comparator
                    Collections.sort(todoList, new Comparator<Task>() {
                                @Override
                                public int compare(Task t1, Task t2) {
                                        return Integer.compare(t2.getPriority(), t1.getPriority());
                                }
                    });
                    System.out.println("Tasks sorted by priority (High to Low):");
                    todoList.forEach(task -> System.out.println(task));
                }
                // Exit the program if the user enters any other key
                else {
                    break;
                }
            }
        } catch (NumberFormatException nfe) {
            // Handle invalid number input (e.g., non-integer input for priority or index)
            System.out.println("Invalid input. Please enter a valid number.");
        }
    }
}
```

3. Compile the Java program, specifying the destination directory as the `classes` directory that you created.

   ```
   javac -d classes src/ArrayListExample.java
   ```

4. Run the program and test with variable combinations.

   ```
   java ArrayListExample
   ```

Sample output would be as below.

```
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
1
Enter the task
laundry
Enter Priority - 1 Low, 2 Medium, 3 High
1
The task has been added to the list
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
1
Enter the task
finish homework
Enter Priority - 1 Low, 2 Medium, 3 High
3
The task has been added to the list
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
5
Tasks sorted by priority (High to Low):
finish homework 3 New
laundry 1 New
Press 1 to add a task,
2 to view all the tasks
3 to change status of tasks
4 to delete a task
Any other key to exit
2
finish homework 3 New
laundry 1 New
Press 1 to add a task,
2 to view all the tasks
```

```
3 to change status of tasks
4 to delete a task
Any other key to exit
```

# Practice Exercise

1. Create a class `Student` with attributes such as name, age and major. Create a console menu to add, update, delete items in the list and sort the students by age.

▶ Click here for sample code

# Conclusion

In this lab, you learned how about ArrayLists and how to manipulate them.

## Author(s)

Lavanya