# [To-Do List Application in Java](https://copyassignment.com/to-do-list-app-in-java/)

Thammishetti nikhitha  February,2024

[](https://copyassignment.com/to-do-list-app-in-java/)

Hello, everyone👋

Today we are going to build a simple and elegant To-Do List App in Java. This project will teach you how to perform adding, and delete tasks to a List. We will build a cool UI with the help of Java Swing and Java AWT. So let’s start together!!!

**Project Overview: To-Do List Application in Java**

|  |  |
| --- | --- |
| **Project Name:** | To-Do List Application in Java |
| **Abstract** | It’s a GUI-based project used with the swing module to organize all the elements that work under To-Do List App in Java. |
| **Language/s Used:** | Java |
| **IDE** | Vs code |
| **Java version (Recommended):** | **Java(latest version)** |
| **Database:** | None |
| **Type:** | Desktop Application |
| Recommended for | Intermediate Java Programmers |
| Time to build: | 30 mins – 60 mins |

**What you will learn after creating To-Do List Application in Java?**

* Handling Classes and Objects creation (very important as this is the standard used in industry-ready programs)
* Dealing with data structures like List in Java
* Functions, Loops, Conditionals, and variables
* Java Swing and Java AWT for designing a simple and attractive User Interface.

**Features of Project:**

* Adding a task
* Marking the task done when it’s finished
* Deleting all the finished tasks at once

Now, we will understand the code for To-Do List App in Java, we will understand each code using comments.

**Complete Code for To-Do List in java**

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.BorderFactory;

import javax.swing.Box;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.Border;

import java.awt.Color;

import java.awt.BorderLayout;

import java.awt.Component;

class Task extends JPanel {

JLabel index;

JTextField taskName;

JButton done;

Color pink = new Color(255, 161, 161);

Color green = new Color(188, 226, 158);

Color doneColor = new Color(233, 119, 119);

private boolean checked;

Task() {

this.setPreferredSize(new Dimension(400, 20)); // set size of task

this.setBackground(pink); // set background color of task

this.setLayout(new BorderLayout()); // set layout of task

checked = false;

index = new JLabel(""); // create index label

index.setPreferredSize(new Dimension(20, 20)); // set size of index label

index.setHorizontalAlignment(JLabel.CENTER); // set alignment of index label

this.add(index, BorderLayout.WEST); // add index label to task

taskName = new JTextField("Write something.."); // create task name text field

taskName.setBorder(BorderFactory.createEmptyBorder()); // remove border of text field

taskName.setBackground(pink); // set background color of text field

this.add(taskName, BorderLayout.CENTER);

done = new JButton("Done");

done.setPreferredSize(new Dimension(80, 20));

done.setBorder(BorderFactory.createEmptyBorder());

done.setBackground(doneColor);

done.setFocusPainted(false);

this.add(done, BorderLayout.EAST);

}

public void changeIndex(int num) {

this.index.setText(num + ""); // num to String

this.revalidate(); // refresh

}

public JButton getDone() {

return done;

}

public boolean getState() {

return checked;

}

public void changeState() {

this.setBackground(green);

taskName.setBackground(green);

checked = true;

revalidate();

}

}

class List extends JPanel {

Color lightColor = new Color(252, 221, 176);

List() {

GridLayout layout = new GridLayout(10, 1);

layout.setVgap(5); // Vertical gap

this.setLayout(layout); // 10 tasks

this.setPreferredSize(new Dimension(400, 560));

this.setBackground(lightColor);

}

public void updateNumbers() {

Component[] listItems = this.getComponents();

for (int i = 0; i < listItems.length; i++) {

if (listItems[i] instanceof Task) {

((Task) listItems[i]).changeIndex(i + 1);

}

}

}

public void removeCompletedTasks() {

for (Component c : getComponents()) {

if (c instanceof Task) {

if (((Task) c).getState()) {

remove(c); // remove the component

updateNumbers(); // update the indexing of all items

}

}

}

}

}

class Footer extends JPanel {

JButton addTask;

JButton clear;

Color orange = new Color(233, 133, 128);

Color lightColor = new Color(252, 221, 176);

Border emptyBorder = BorderFactory.createEmptyBorder();

Footer() {

this.setPreferredSize(new Dimension(400, 60));

this.setBackground(lightColor);

addTask = new JButton("Add Task"); // add task button

addTask.setBorder(emptyBorder); // remove border

addTask.setFont(new Font("Sans-serif", Font.ITALIC, 20)); // set font

addTask.setVerticalAlignment(JButton.BOTTOM); // align text to bottom

addTask.setBackground(orange); // set background color

this.add(addTask); // add to footer

this.add(Box.createHorizontalStrut(20)); // Space between buttons

clear = new JButton("Clear finished tasks"); // clear button

clear.setFont(new Font("Sans-serif", Font.ITALIC, 20)); // set font

clear.setBorder(emptyBorder); // remove border

clear.setBackground(orange); // set background color

this.add(clear); // add to footer

}

public JButton getNewTask() {

return addTask;

}

public JButton getClear() {

return clear;

}

}

class TitleBar extends JPanel {

Color lightColor = new Color(252, 221, 176);

TitleBar() {

this.setPreferredSize(new Dimension(400, 80)); // Size of the title bar

this.setBackground(lightColor); // Color of the title bar

JLabel titleText = new JLabel("To Do List"); // Text of the title bar

titleText.setPreferredSize(new Dimension(200, 60)); // Size of the text

titleText.setFont(new Font("Sans-serif", Font.BOLD, 20)); // Font of the text

titleText.setHorizontalAlignment(JLabel.CENTER); // Align the text to the center

this.add(titleText); // Add the text to the title bar

}

}

class AppFrame extends JFrame {

private TitleBar title;

private Footer footer;

private List list;

private JButton newTask;

private JButton clear;

AppFrame() {

this.setSize(400, 600); // 400 width and 600 height

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // Close on exit

this.setVisible(true); // Make visible

title = new TitleBar();

footer = new Footer();

list = new List();

this.add(title, BorderLayout.NORTH); // Add title bar on top of the screen

this.add(footer, BorderLayout.SOUTH); // Add footer on bottom of the screen

this.add(list, BorderLayout.CENTER); // Add list in middle of footer and title

newTask = footer.getNewTask();

clear = footer.getClear();

addListeners();

}

public void addListeners() {

newTask.addMouseListener(new MouseAdapter() {

@override

public void mousePressed(MouseEvent e) {

Task task = new Task();

list.add(task); // Add new task to list

list.updateNumbers(); // Updates the numbers of the tasks

task.getDone().addMouseListener(new MouseAdapter() {

@override

public void mousePressed(MouseEvent e) {

task.changeState(); // Change color of task

list.updateNumbers(); // Updates the numbers of the tasks

revalidate(); // Updates the frame

}

});

}

});

clear.addMouseListener(new MouseAdapter() {

@override

public void mousePressed(MouseEvent e) {

list.removeCompletedTasks(); // Removes all tasks that are done

repaint(); // Repaints the list

}

});

}

}

public class ToDoList {

public static void main(String args[]) {

AppFrame frame = new AppFrame(); // Create the frame

}

}

@interface override {

}

**Output:**

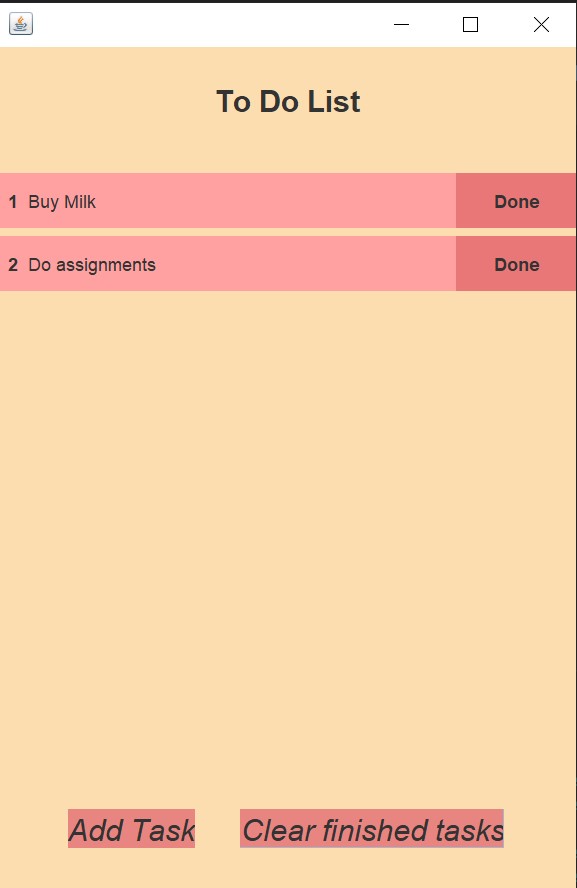
Now, hit the run button near the top of the main method or go to the left top corner and find the run button. After running To-Do List App in Java, we will encounter the following output, let’s see.



After the task is finished, you can mark them done.



If you want to remove all the tasks you have marked as done just click on the clear finished tasks button and it will be removed.



Once again you can click on the add task button and add more tasks to the list of tasks to accomplish.

So that was it. Hope you enjoyed building this beautiful project and learned from it.