

Goal Practice how to create a Java GUI application with NetBeans that tracks the mouse.

Mouse Tracker

1. Create a new project:
 - (a) File > New Project. . .
 - (b) Categories: Java; Projects: Java Application > **Next**
 - (c) Project Name: MouseTracker; *uncheck* Create Main Class > **Finish**
2. Create a new package `gui`:
 - (a) In the top-left navigation panel, in the Projects tab, show the Source Packages of the MouseTracker project (click on the small triangles in the tree view).
 - (b) Right-click on Source Packages, and select New > Java Package. . .
 - (c) Package Name: gui > **Finish**
3. Create a new frame (window):
 - (a) Right-click on the newly created package, and select New > JFrame Form. . .
 - (b) Class Name: MainFrame > **Finish**
4. Add a panel to the frame (it is recommended not to do things directly on a frame):
 - (a) Make sure that you see the **Design** view of `MainFrame`.
 - (b) From the Palette on the right, drag a Panel onto the form (the dark grey rectangle).
 - (c) Resize the panel to cover the entire form.
5. Add handlers for the mouse events to the panel:
 - (a) Right-click in the newly added panel, and select Events > Mouse > MousePressed
 - (b) You are now switched to the **Source** view of `MainFrame`, with the cursor in an empty method:

```
1 private void jPanellMousePressed(java.awt.event.MouseEvent evt) {
2     // TODO add your handling code here:
3 }
```

6. Add code to handle the event.

- (a) Above the event handling method, add an auxiliary method:

```
private void printPoint(final String msg, final Point point) {  
    System.out.println(msg + "(" + point.x + ", " + point.y + ")");  
}
```

- (b) Right-click on the source code, and select **Fix Imports**. This will insert the code `import java.awt.Point` to make the definition of `Point` available.

- (c) Now replace the comment in method `jPanellMousePressed` by

```
printPoint("Mouse pressed at ", evt.getPoint());
```

- (d) Repeat these steps to add and handle `jPanellMouseReleased`

7. Run the program:

- (a) Run > Run Main Project

- (b) On the first run, you will get a dialogue window **Run Project**, to **Select the main class**: `gui.MainFrame`.

- (c) Move the mouse into the frame of the running application, and press the mouse button. Observe the console output. Move the mouse, and release the mouse button. Again, observe the console output. Etc.

8. Find out what the coordinates are of the four corners of the frame.

9. Exit the application by closing its frame.

10. Other things to do:

- You can also add handlers for the events `MouseEntered`, `MouseExited`.
- You can change some *properties* of the panel.
Right-click in the panel, and select **Properties**.
There you can change, for example, the color of the **background**, and set a `toolTipText`.

Notes:

- By right-clicking on the project, and selecting **Properties**, you can change some settings. For example, the main class, but also javadoc processing.
- It is easier to work on a project when it is set as the main project. You can do this by right-clicking on the project, and selecting **Set as Main Project**.