Demo — Null Objects vs Java 8 Optional
When is it better to use an anonymous inner class than a standalone class?

- If the class is used only once and is easy to implement
- When we have only a few lines of code in the anonymous class
- When it is very small
- When you need to pass many values and used only for one thing
When is it better to use a lambda than an anonymous inner class?

if just one method is used
Why is it better to implement the TicTacToe GUI as a separate class rather than by extending the existing TicTacToe class?

- To separate the view from the model
- Because the GUI and the TicTacToe class are to separate things
- Easier to implement, easier handling of listeners, better structure
- Because of the responsibility
- The GUI shouldn't know about the game and vice versa
- Better OOD
- Because the GUI has new different responsibilities
- Improves flexibility for future changes
- MVC
Why is it better to implement the TicTacToe GUI as a separate class rather than by extending the existing TicTacToe class?

- easier to debug
- you could use the gui without the tictactoe game and with another game
- we can use it for other Classes
- "Logic" should be separated from GUI, in case e.g. you want to change GUI
- makes it easier to for example change between guis
- both could be reused in separate, independent applications
Why does the TicTacToe model notify the GameGUI of changes through the Observer interface instead of directly updating the GUI display?

- Message passing instead of taking over GUI's responsibility
- Because every observer displays the GUI differently
- Because the GUI is thread-based and not sequential
- If there are multiple observers they will all be notified
- Concurrency?
Last chance for questions