

# Java Development Kit

## Семинар 1

Простейшие интерфейсы пользователя



## Компиляция и интерпретация кода

1

Лекция 1.  
Графический интерфейс  
пользователя

2

Семинар 1.  
Простейшие интерфейсы  
пользователя

3

Лекция 2.  
Интерфейсы

4

Семинар 2.  
Интерфейсы и API

5

Лекция 3.  
Обобщённое  
программирование

6

Семинар 3.  
Универсальное  
проектирование

7

Лекция 4.  
ООП и исключения  
Многопоточность

8

Семинар 4.  
Параллелизм и  
асинхронность

9

Лекция 5.  
Коллекции

10

Семинар 5.  
Структурирование данных

11

Лекция 6.  
Управление проектом:  
сборщики проектов

12

Семинар 6.  
Сборка проекта



## Цели семинара

1. Практика создания простых экранных форм — кнопки, компоненты форм;
2. Передача данных между формами и внутри формы;
3. Написание полноценного окна клиента чата;
4. Написание графической оболочки для настроек игры в крестики-нолики.





## План семинара

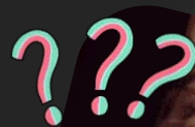
1. Приветствие;
2. Quiz;
3. Рассмотрение домашнего задания;
4. Вопросы и ответы;
5. Практика;
6. Домашнее задание;
7. Рефлексия.





Вопросы?

Вопросы?



Вопросы?





# Викторина





## Свойства окна, такие как размер и заголовок возможно задать

1. Написанием методов  
в классе-наследнике;

2. Вызовом методов  
в конструкторе;

3. Созданием констант  
в первых строках класса.



## Свойства окна, такие как размер и заголовок возможно задать

1. Написанием методов  
в классе-наследнике;

2. Вызовом методов  
в конструкторе;

3. Созданием констант  
в первых строках класса.



**Эти свойства задаются  
при создании**





## Свойства окна, такие как размер и заголовок возможно задать

1. Написанием методов  
в классе-наследнике;

✗ Неверно (это расширение функциональности)

2. Вызовом методов  
в конструкторе;

✓ Верно

3. Созданием констант  
в первых строках класса.

✗ Неверно (сами по себе константы ничего не дают)



## Для выполнения кода по нажатию кнопки на интерфейсе нужно

1. Создать обработчик кнопки и вписать код в него;
2. Переопределить метод нажатия у компонента кнопки;
3. Написать код непосредственно в методе создания кнопки.



## Для выполнения кода по нажатию кнопки на интерфейсе нужно

1. Создать обработчик кнопки и вписать код в него;

2. Переопределить метод нажатия у компонента кнопки;

3. Написать код непосредственно в методе создания кнопки.



**Нажатие — это событие, на которое нужно отреагировать**



## Для выполнения кода по нажатию кнопки на интерфейсе нужно

1. Создать обработчик кнопки и вписать код в него;

✓ Верно

2. Переопределить метод нажатия у компонента кнопки;

✗ Неверно (JVM не узнает какое переопределение вызвать)

3. Написать код непосредственно в методе создания кнопки.

✗ Неверно (нет такой перегрузки)



## Что такое Java Swing?

1. Набор библиотек для создания реактивных GUI;

2. Среда разработки для Java;

3. Язык программирования.



## Что такое Java Swing?

1. Набор библиотек для создания реактивных GUI;

2. Среда разработки для Java;

3. Язык программирования.





## Что такое Java Swing?

1. Набор библиотек для создания реактивных GUI;

✓ Верно

2. Среда разработки для Java;

✗ Неверно (написана на Swing)

3. Язык программирования.

✗ Неверно (мы точно помним, какой язык изучаем?)



## Какая библиотека используется для создания графических интерфейсов на Java?

1. Java Swing;

2. JavaFX;

3. Обе библиотеки.





## Какая библиотека используется для создания графических интерфейсов на Java?

1. Java Swing;

2. JavaFX;

3. Обе библиотеки.



JavaFX вывели из стандарта начиная с Java9



## Какая библиотека используется для создания графических интерфейсов на Java?

1. Java Swing;

✗ Неверно (стандартный классический механизм)

2. JavaFX;

✗ Неверно (библиотека со встроенной поддержкой паттерна MVC)

3. Обе библиотеки.

✓ Верно



# Рассмотрение домашнего задания





## Задание:

Полностью разобраться с кодом.

## Решение:

Решение не требует проверки.



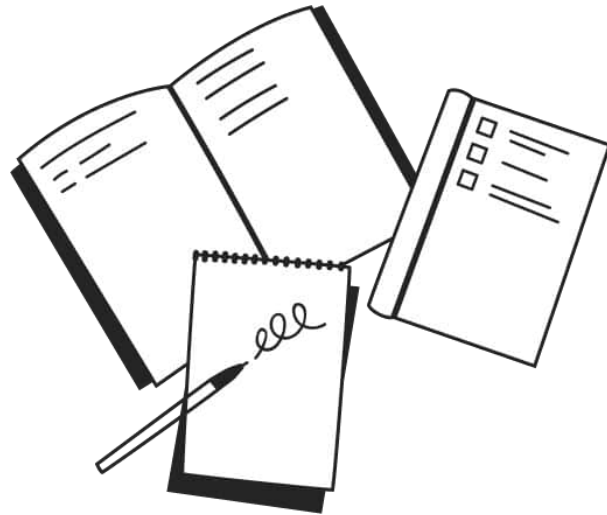


## Задание:

Переделать проверку победы, чтобы она не была реализована просто набором условий.

## Решение:

Задание подразумевало переписывание изменяющейся индексации в циклы, но, поскольку для игры в крестики-нолики нужно также иметь возможность работать на любом поле с любым числом фигур подряд для победы – имеет смысл рассматривать сразу следующее задание.





## Задание:

Попробовать переписать логику проверки победы, чтобы она работала для поля 5x5 и количества фигур 4

## Решение:

```
197
198 private boolean checkLine(int x, int y, int vx, int vy, int len, int c) {
199     final int far_x = x + (len - 1) * vx;
200     final int far_y = y + (len - 1) * vy;
201     if (!isValidCell(far_x, far_y)) return false;
202     for (int i = 0; i < len; i++) {
203         if (field[y + i * vy][x + i * vx] != c) return false;
204     }
205     return true;
206 }
207
```

```
207
208 private boolean checkWin(int c) {
209     for (int i = 0; i < fieldSizeX; i++) {
210         for (int j = 0; j < fieldSizeY; j++) {
211             if (checkLine(i, j, 1, 0, winLength, c)) return true;
212             if (checkLine(i, j, 1, 1, winLength, c)) return true;
213             if (checkLine(i, j, 0, 1, winLength, c)) return true;
214             if (checkLine(i, j, 1, -1, winLength, c)) return true;
215         }
216     }
217     return false;
218 }
219
```



## Задание:

\*\* Доработать искусственный интеллект, чтобы он мог примитивно блокировать ходы игрока, и примитивно пытаться выиграть сам

## Решение:

```
178
179     private void aiTurn() {
180         if (turnAIWinCell()) return;
181         if (turnHumanWinCell()) return;
182         int x, y;

189
190     private boolean turnAIWinCell() {
191         for (int i = 0; i < fieldSizeY; i++) {
192             for (int j = 0; j < fieldSizeX; j++) {
193                 if (isEmptyCell(j, i)) {
194                     field[i][j] = AI_DOT;
195                     if (checkWin(AI_DOT)) return true;
196                     field[i][j] = EMPTY_DOT;
197                 }
198             }
199         }
200         return false;
201     }
202 }
```

## Решение:

```
202
203     private boolean turnHumanWinCell() {
204         for (int i = 0; i < fieldSizeY; i++) {
205             for (int j = 0; j < fieldSizeX; j++) {
206                 if (isEmptyCell(j, i)) {
207                     field[i][j] = HUMAN_DOT;
208                     if (checkWin(HUMAN_DOT)) {
209                         field[i][j] = HUMAN_DOT;
210                         return true;
211                     }
212                     field[i][j] = EMPTY_DOT;
213                 }
214             }
215         }
216         return false;
217     }
218 }
```



Вопросы?



Вопросы?

Вопросы?







# Практика

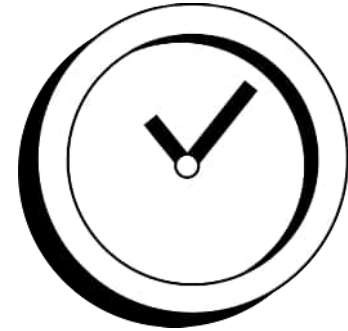




## Задание 1

**Задача:** На лекции был написан фрейм, содержащий одну кнопку – начать игру и расположением самого окна настроек автоматически, относительно игрового окна.

Добавить на экран компоновщик-сетку с одним столбцом и добавить над существующей кнопкой следующие компоненты в заданном порядке: JLabel с заголовком «Выберите режим игры», сгруппированные в ButtonGroup переключаемые JRadioButton с указанием режимов «Человек против компьютера» и «Человек против человека», JLabel с заголовком «Выберите размеры поля», JLabel с заголовком «Установленный размер поля:», JSlider со значениями 3..10, JLabel с заголовком «Выберите длину для победы», JLabel с заголовком «Установленная длина:», JSlider со значениями 3..10.

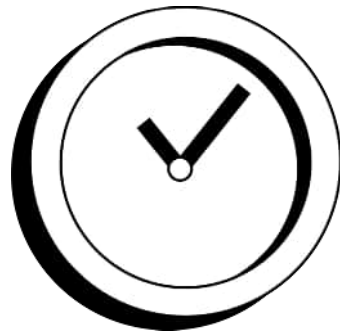
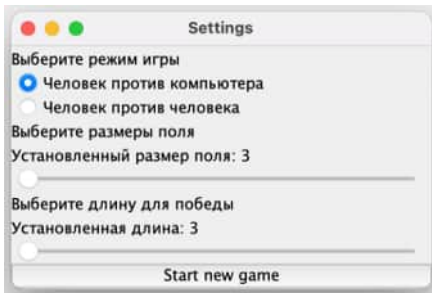


**15 минут**



## Задание 1

**Задача:** Добавить на экран компоновщик-сетку с одним столбцом и добавить над существующей кнопкой следующие компоненты в заданном порядке: JLabel с заголовком «Выберите режим игры», сгруппированные в ButtonGroup переключаемые JRadioButton с указанием режимов «Человек против компьютера» и «Человек против человека», JLabel с заголовком «Выберите размеры поля», JLabel с заголовком «Установленный размер поля:», JSlider со значениями 3..10, JLabel с заголовком «Выберите длину для победы», JLabel с заголовком «Установленная длина:», JSlider со значениями 3..10.



<<15:00->>



## Задание 1

**Задача:** Добавить на экран компоновщик-сетку с одним столбцом и добавить над существующей кнопкой следующие компоненты в заданном порядке: JLabel с заголовком «Выберите режим игры», сгруппированные в ButtonGroup переключаемые JRadioButton с указанием режимов «Человек против компьютера» и «Человек против человека», JLabel с заголовком «Выберите размеры поля», JLabel с заголовком «Установленный размер поля:», JSlider со значениями 3..10, JLabel с заголовком «Выберите длину для победы», JLabel с заголовком «Установленная длина:», JSlider со значениями 3..10.

Пример, на вашем экране →

```
78      setLayout(new GridLayout(10,1));
79      add(new JLabel("Выберите режим игры"));
80      ButtonGroup bg = new ButtonGroup();
81      JRadioButton pvc = new JRadioButton("Человек против компьютера");
82      JRadioButton pvp = new JRadioButton("Человек против человека");
83      bg.add(pvc);
84      bg.add(pvp);
85      add(pvc);
86      add(pvp);
87      add(new JLabel("Выберите размеры поля"));
88      add(new JLabel("Установленный размер поля:"));
89      add(new JSlider(3, 10, 3));
90      add(new JLabel("Выберите длину для победы"));
91      add(new JLabel("Установленная длина:"));
92      add(new JSlider(3, 10, 3));
93      add(btnStart);
```



## Задание 1\*

**Задача:** Сгруппировать объявление компонентов в два метода по смыслу – регулирование режима игры, регулирование параметров поля. Объявить компоненты так, чтобы они оказались доступны для обработчика кнопки.



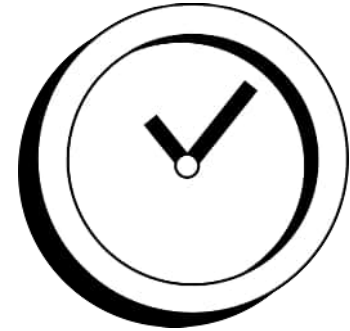
## Задание 1\*\*

**Задача:** Вынести неизменяемую часть изменяемых сообщений (подписи к слайдерам) в константы класса. Вынести размеры, применяемые в слайдерах в константы класса (избавиться от магических чисел). Вынести обработчик кнопки в отдельный метод.



## Задание 2

**Задача:** Добавить компонентам интерактивности, а именно, при перемещении ползунка слайдера в соответствующих лейблах должны появляться текущие значения слайдеров. Для этого необходимо добавить к слайдеру слушателя изменений (как это было сделано для действия кнопки).

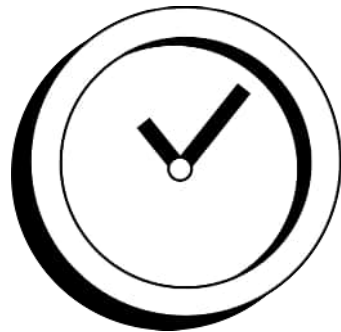


**15 минут**

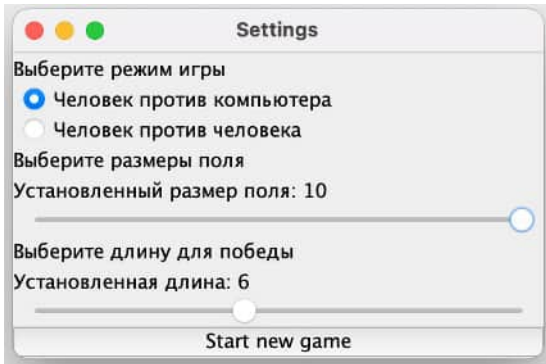
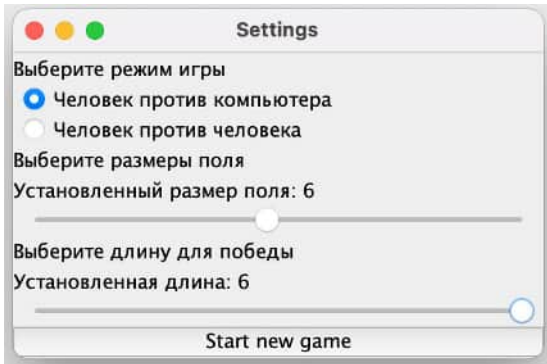


## Задание 2

**Задача:** Добавить компонентам интерактивности, а именно, при перемещении ползунка слайдера в соответствующих лейблах должны появляться текущие значения слайдеров. Для этого необходимо добавить к слайдеру слушателя изменений (как это было сделано для действия кнопки).



<<15:00->>







## Задание 2

**Задача:** Добавить компонентам интерактивности, а именно, при перемещении ползунка слайдера в соответствующих лейблах должны появляться текущие значения слайдеров. Для этого необходимо добавить к слайдеру слушателя изменений (как это было сделано для действия кнопки).

Пример, на вашем экране →

```
38 JLabel lbFieldSize = new JLabel(FIELD_SIZE_PREFIX + MIN_FIELD_SIZE);
39 JLabel lbWinLength = new JLabel(WIN_LENGTH_PREFIX + MIN_FIELD_SIZE);
40 slideFieldSize = new JSlider(MIN_FIELD_SIZE, MAX_FIELD_SIZE, MIN_FIELD_SIZE);
41 slideWinLen = new JSlider(MIN_WIN_LENGTH, MIN_FIELD_SIZE, MIN_FIELD_SIZE);
42 slideWinLen.addChangeListener(new ChangeListener() {
43     @Override
44     public void stateChanged(ChangeEvent e) {
45         lbWinLength.setText(WIN_LENGTH_PREFIX + slideWinLen.getValue());
46     }
47 });
48 slideFieldSize.addChangeListener(new ChangeListener() {
49     @Override
50     public void stateChanged(ChangeEvent e) {
51         int currentValue = slideFieldSize.getValue();
52         lbFieldSize.setText(FIELD_SIZE_PREFIX + currentValue);
53         slideWinLen.setMaximum(currentValue);
54     }
55 });
```



## Задание 2\*

**Задача:** Добавить автоматическое регулирование максимального значения у слайдера выигрышной длины при изменении значения слайдера размера поля.



## Задание 2\*\*

**Задача:** Добавить центрирование окна настроек относительно главного (родительского) окна. То есть, в центре родительского окна должен быть не левый верхний угол окна настроек (как это сделано сейчас), а также его центр.



## Задание 3

**Задача:** В методе обработчика нажатия кнопки необходимо заменить константы в аргументе вызова метода старта игры на текущие показания компонентов (какая радио-кнопка активна, значение слайдера размеров поля, значение слайдера выигрышной длины).



**10 минут**



## Задание 3

**Задача:** В методе обработчика нажатия кнопки необходимо заменить константы в аргументе вызова метода старта игры на текущие показания компонентов (какая радио-кнопка активна, значение слайдера размеров поля, значение слайдера выигрышной длины).



<<10:00->>



## Задание 3

**Задача:** В методе обработчика нажатия кнопки необходимо заменить константы в аргументе вызова метода старта игры на текущие показания компонентов (какая радио-кнопка активна, значение слайдера размеров поля, значение слайдера выигрышной длины).

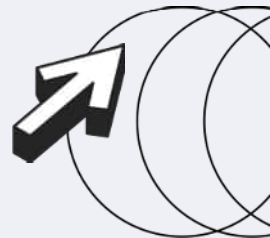
Пример, на вашем экране →

```
9      public class Map extends JPanel {
10          public static final int MODE_HVA = 0;
11          public static final int MODE_HVH = 1;

84      private void btnStartDelegate() {
85          int gameMode;
86          if (humVSAI.isSelected()) {
87              gameMode = Map.MODE_HVA;
88          } else if (humVShum.isSelected()) {
89              gameMode = Map.MODE_HVH;
90          } else {
91              throw new RuntimeException("Unknown game mode");
92          }
93          int fieldSize = slideFieldSize.getValue();
94          int winLength = slideWinLen.getValue();
95          gameWindow.startNewGame(gameMode, fieldSize, fieldSize, winLength);
96          setVisible(false);
97      }
```



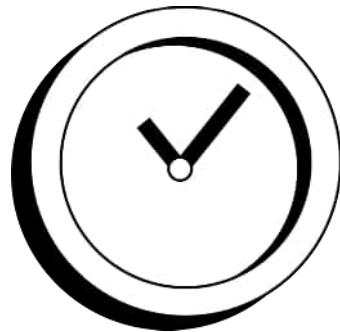
**Перерыв**  
<<5:00->>





## Задание 4

**Задача:** Создать простейшее окно управления сервером (по сути, любым), содержащее две кнопки (JButton) – запустить сервер и остановить сервер. Кнопки должны просто логировать нажатие (имитировать запуск и остановку сервера, соответственно) и выставлять внутри интерфейса соответствующее булево `isServerWorking`.



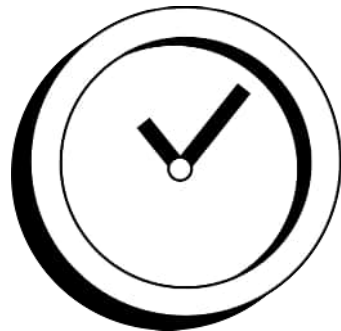
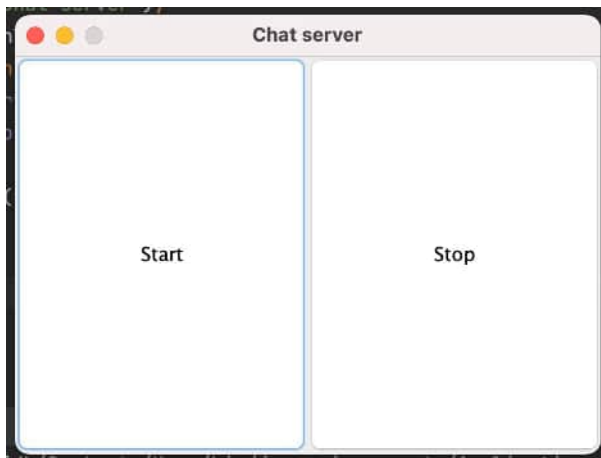
**10 минут**





## Задание 4

**Задача:** Создать простейшее окно управления сервером (по сути, любым), содержащее две кнопки (JButton) – запустить сервер и остановить сервер. Кнопки должны просто логировать нажатие (имитировать запуск и остановку сервера, соответственно) и выставлять внутри интерфейса соответствующее булево `isServerWorking`.



<<10:00->>



## Задание 4

**Задача:** Создать простейшее окно управления сервером (по сути, любым), содержащее две кнопки (JButton) – запустить сервер и остановить сервер. Кнопки должны просто логировать нажатие (имитировать запуск и остановку сервера, соответственно) и выставлять внутри интерфейса соответствующее булево `isServerWorking`.

Пример, на вашем экране →

```
8 ▶ public class ServerWindow extends JFrame {
9     private static final int POS_X = 500;
10    private static final int POS_Y = 550;
11    private static final int WIDTH = 400;
12    private static final int HEIGHT = 300;
13
14    private final JButton btnStart = new JButton("Start");
15    private final JButton btnStop = new JButton("Stop");
16    private final JTextArea log = new JTextArea();
17    private boolean isServerWorking;
18
19 ▶ public static void main(String[] args) {
20     new ServerWindow();
21 }
```

```
22
23 private ServerWindow() {
24     isServerWorking = false;
25     btnStop.addActionListener(new ActionListener() {
26         @Override
27         public void actionPerformed(ActionEvent e) {
28             isServerWorking = false;
29             System.out.println("Server stopped " + isServerWorking + "\n");
30         }
31     });
32
33     btnStart.addActionListener(new ActionListener() {
34         @Override
35         public void actionPerformed(ActionEvent e) {
36             isServerWorking = true;
37             System.out.println("Server started " + isServerWorking + "\n");
38         }
39     });
40 }
```

```
40
41     setDefaultCloseOperation(EXIT_ON_CLOSE);
42     setBounds(POS_X, POS_Y, WIDTH, HEIGHT);
43     setResizable(false);
44     setTitle("Chat server");
45     setAlwaysOnTop(true);
46     setLayout(new GridLayout(1, 2));
47     add(btnStart);
48     add(btnStop);
49
50     setVisible(true);
51 }
52 }
```



## Задание 4\*

**Задача:** Если сервер не запущен, кнопка остановки должна сообщить, что сервер не запущен и более ничего не делать. Если сервер запущен, кнопка старта должна сообщить, что сервер работает и более ничего не делать.



## Задание 4\*\*

**Задача:** Добавить на окно компонент `JTextArea` и выводить сообщения сервера в него, а не в терминал.



## Задание 5

**Задача:** Создать окно клиента чата. Окно должно содержать JTextField для ввода логина, пароля, IP-адреса сервера, порта подключения к серверу, область ввода сообщений, JTextArea область просмотра сообщений чата и JButton подключения к серверу и отправки сообщения в чат. Желательно сразу сгруппировать компоненты, относящиеся к серверу сгруппировать на JPanel сверху экрана, а компоненты, относящиеся к отправке сообщения – на JPanel снизу

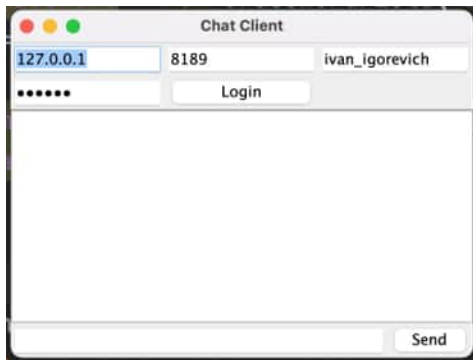


**20 минут**



## Задание 5

**Задача:** Создать окно клиента чата. Окно должно содержать JTextField для ввода логина, пароля, IP-адреса сервера, порта подключения к серверу, область ввода сообщений, JTextArea область просмотра сообщений чата и JButton подключения к серверу и отправки сообщения в чат. Желательно сразу сгруппировать компоненты, относящиеся к серверу сгруппировать на JPanel сверху экрана, а компоненты, относящиеся к отправке сообщения – на JPanel снизу



<<20:00->>



## Задание 5

**Задача:** Создать окно клиента чата. Окно должно содержать JTextField для ввода логина, пароля, IP-адреса сервера, порта подключения к серверу, область ввода сообщений, JTextArea области просмотра сообщений чата и JButton подключения к серверу и отправки сообщения в чат. Желательно сразу сгруппировать компоненты, относящиеся к серверу сгруппировать на JPanel сверху экрана, а компоненты, относящиеся к отправке сообщения – на JPanel снизу

Пример, на вашем экране →

```
1 public class ClientGUI extends JFrame {
2     private static final int WIDTH = 400;
3     private static final int HEIGHT = 300;
4
5     private final JTextArea log = new JTextArea();
6
7     private final JPanel panelTop = new JPanel(new GridLayout(2, 3));
8     private final JTextField tfIPAddress = new JTextField("127.0.0.1");
9     private final JTextField tfPort = new JTextField("8189");
10    private final JTextField tfLogin = new JTextField("ivan_igorovich");
11    private final JPasswordField tfPassword = new JPasswordField("123456");
12    private final JButton btnLogin = new JButton("Login");
13
14    private final JPanel panelBottom = new JPanel(new BorderLayout());
15    private final JTextField tfMessage = new JTextField();
16    private final JButton btnSend = new JButton("Send");
17
18    ClientGUI() {
19        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
20        setLocationRelativeTo(null);
21        setSize(WIDTH, HEIGHT);
22        setTitle("Chat Client");
23
24        panelTop.add(tfIPAddress);
25        panelTop.add(tfPort);
26        panelTop.add(tfLogin);
27        panelTop.add(tfPassword);
28        panelTop.add(btnLogin);
29        add(panelTop, BorderLayout.NORTH);
30
31        panelBottom.add(tfMessage, BorderLayout.CENTER);
32        panelBottom.add(btnSend, BorderLayout.EAST);
33        add(panelBottom, BorderLayout.SOUTH);
34
35        log.setEditable(false);
36        JScrollPane scrollLog = new JScrollPane(log);
37        add(scrollLog);
38
39        setVisible(true);
40    }
41
42    public static void main(String[] args) {
43        new ClientGUI();
44    }
45 }
```



## Задание 5\*

**Задача:** Добавить на экран компонент `JList` – имитацию списка пользователей, заполнить этот список несколькими выдуманными именами пользователей чата. Подсказка: компонент не может добавлять или убирать элементы списка, он работает с методом `setListData()`, изучите его аргументы





# Домашнее задание





## Домашнее задание

### Задачи:

1. Выполнить все задания семинара, если они не были решены, без ограничений по времени;
2. Отправлять сообщения из текстового поля сообщения в лог по нажатию кнопки или по нажатию клавиши Enter на поле ввода сообщения;
3. Продублировать импровизированный лог (историю) чата в файле;
4. При запуске клиента чата заполнять поле истории из файла, если он существует. Обратите внимание, что чаще всего история сообщений хранится на сервере и заполнение истории чата лучше делать при соединении с сервером, а не при открытии окна клиента.





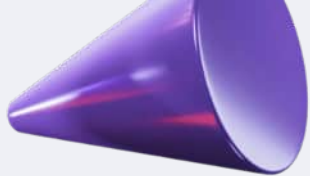
Вопросы?



Вопросы?

Вопросы?





**Спасибо за внимание**

