

# Язык Java и разработка Java-приложений

Библиотека **Eclipse Standard Widget Toolkit**

*Разработка интерфейса пользователя  
на примере программы «Блокнот настольных игр».*

Романов Владимир Юрьевич,  
Московский Государственный Университет им. М.В.Ломоносова  
Факультет Вычислительной Математики и Кибернетики  
[vromanov@cs.msu.su](mailto:vromanov@cs.msu.su),  
[vladimir.romanov@gmail.com](mailto:vladimir.romanov@gmail.com)

# Настольные игры

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

<https://www.chessprogramming.org/Games>

The screenshot shows a web browser displaying the 'Games' page of the Chess Programming Wiki (CPW). The URL in the address bar is <https://www.chessprogramming.org/Games>. The page title is 'Games'. On the left, there's a sidebar with links like 'Main page', 'Recent changes', 'Random page', 'Help', 'Tools', 'What links here', 'Related changes', 'Special pages', 'Printable version', 'Permanent link', and 'Page information'. A tooltip over the 'Help' link says 'A list of recent changes in the wiki [alt-shift-r]'. The main content area starts with 'Home \* Games' and a paragraph about John von Neumann's classification of chess as a two-player, zero-sum, abstract strategy game. Below that is a 'Contents [hide]' section with a list of categories: '1 Board Games' (which further branches into '1.1 Chess Variants', '1.2 Abstract Board Games', '1.3 Mancala Games', '1.4 Games of Chance', and '1.5 Without perfect information').

# Шахматы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Шашки

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Games Notebook

The screenshot shows a checkers game interface. On the left, the player selection panel shows two teams: 'Черные' (Black) and 'Белые' (White). The Black team includes 'Homo sapience' (selected), 'Незнайка', 'Знайка', 'Спрутс', and 'Скуперфильд'. The White team includes 'Homo sapience' (selected), 'Незнайка', 'Знайка', 'Спрутс', and 'Скуперфильд'. A 'Старт' button is at the bottom. The main board is an 8x8 grid with columns labeled A-H and rows 1-8. The board state is as follows:

	A	B	C	D	E	F	G	H
8		●		●		●		●
7	●		●		●		●	
6				●			●	
5			●		●			
4		●				●		●
3	●					●		
2		●	●	●	●	●	●	
1	●		●		●		●	

A red arrow points to a black piece on square E5. Two green dots are on squares F4 and H4. The right panel shows the game history: 1. e3-f4 h6-g5 2. f4xh6 b6-c5 3. c3-b4. The title of the right panel is 'Homo sapience - Незнайка'.

# Китайские шахматы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Games Notebook

**Игроки**

**Черные**  
Homo sapience  
Незнайка  
Сунь-Цзы  
**Конфуций**

**Белые**  
Homo sapience  
Незнайка  
**Сунь-Цзы**  
Конфуций

**Старт**

**Сунь-Цзы - Конфуций**

- Ec1-e3
- Pe7-e6
- Hb1-d2
- Pe6-e5
- Pe4xe5
- Ch8-h4
- Ch3xh10
- Ri10xh10
- Hd2-e4
- Ch4-h2
- Eg1xh2
- Rh10xh2
- Af1-e2
- Rh2xh1
- Ri1xh1
- Cb8-b4
- Cb3xb10
- Ra10xb10
- Rh1-h10**
- Cb4-b1
- Ra1xb1
- Rb10xb1
- Rh10xg10
- Rb1xd1
- Rg10xf10

0-1

# Викинги

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

**Games Notebook**

Шахматы | Шашки | Сянци | **Викинги** | Тамерлан | Реверси | Го | Халма 8x8 | Ренджю

**Игроки**

**Черные**  
Homo sapience  
Незнайка  
**Рюрик (Новгород)**  
Харальд III (Норвегия)  
Вильгельм I (Нормандия)

**Белые**  
Homo sapience  
Незнайка  
**Рюрик (Новгород)**  
Харальд III (Норвегия)  
Вильгельм I (Нормандия)

**Старт**

**Доска**  
● 9 x 9  
● 11 x 11

**Homo sapience - Рюрик (Новгород)**

1. f5-f7 f1xf5 2. d5-d6 d1xd5
3. e6-f6 d9xd7 4. Ke5-e6 d7-b7
5. Ke6-c6 d5-e5 6. Kc6-c9 e5xe6
7. Kc9-a9 Kc9-a9 8. i6xg6

1-0

# Шахматы Тамерлана

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Games Notebook

Шахматы    Шашки    Сянци    Викинги    Тамерлан    Реверси    Го    Халма 8x8    Рэндзю

**Игроки**

**Черные**

- Номо sapience
- Незнайка
- Тамерлан (Узбек)**
- Тохтамыш (З.Орда)
- Баязид (Турция)
- Насир (Индия)

**Белые**

- Номо sapience**
- Незнайка
- Тамерлан (Узбек)
- Тохтамыш (З.Орда)
- Баязид (Турция)
- Насир (Индия)

**Старт**

**Homo sapience - Тамерлан (Узбек)**

1. Wf2-g4 e8-e7

# Реверси

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Games Notebook

Игроки

**Черные**

- Homo sapience
- Винни**
- Сова
- Тигра

**Белые**

- Homo sapience
- Винни
- Сова**
- Тигра

**Старт**

Счет

Белые: 18

Черные: 5

Сова - Винни

1. d6	c6	2. b6	c4
3. d3	e6	4. f6	d2
5. d1	c2	6. b3	d7
7. e8	a6	8. f5	f7
9. g8	d8	10. c8	g7
11. h7	g5	12. h5	h8
13. g6	f4	14. g4	c5
15. a7	e7	16. f8	b8
17. a5	b5	18. a4	c7
19. b7	c3	20. b4	a2
21. a3	a8	22. Pass	c1
23. b1	h3	24. h4	g3
25. h6	e2	26. h2	e1
27. f1	e3	28. f3	f2
29. g2	h1	30. g1	b2
31. Pass			

0-1

# УГОЛКИ

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Games Notebook

**Игроки**

**Черные**  
Homo sapience  
Незнайка  
Муравьи  
**Жуки**

**Белые**  
Homo sapience  
Незнайка  
**Муравьи**  
Жуки

**Старт**

**Доска**

- 8 x 8
- 10 x 10
- 16 x 16

**Счет**

Белые: 174  
Черные: 176

**Муравьи - Жуки**

1. d10xf10	h2xf2	2. b10xd10	j2xh1
3. a7xa5	g1xg3	4. a9xa7	i1xg1
5. d10xd8	j4xh4	6. a7xc7	i3xi5
7. b8xb6	g1xe1	8. c9xe9	g2xg
9. c8xe8	i2xg2	10. a6xc6	i4xi6
11. b7xd7	h3xh5	12. d9xf9	h1xh
13. e10xg10	h3xf3	14. f10xf8	g2xe
15. e9xe7	f1xd1	16. c7xc5	g3xe
17. d8xd6	h4xh6	18. f9xf7	f2xd
19. d7xd5	e1xc1	20. e8xg8	i5xg1
21. c6xc4	d1xd3	22. e7xg7	i6xg1
23. c5xe5	e2xc2	24. f8xh8	h5xf
25. d6xd4	d2xb2	26. f7xh7	e3xc
27. c4xe4	h6xf6	28. g8xi8	c2xc
29. d4xf4	f5xf7	30. d5xf5	g6xg
31. g7xi7	g4xg6	32. e5xe3	g5xg
33. h8xh6	d3xb3	34. h7xh5	f3xd
35. e4xg4	g6xe6	36. i8xi6	c3xc

1-0

# Практикум по языку Java

## Возможные темы практикума

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Реализация **правил игры** для новой настольной игры  
(например, с сайта <https://www.chessprogramming.org/Games>)
  - Японские шахматы
  - ...
- Реализация **новых алгоритмов игры** для уже существующих игр (**Незнайка, Винни Пух, Сова, ...**)
- Реализация интерфейса пользователя **с помощью новой библиотеки**
  - Swing
  - JavaFX
  - Google Web Toolkit (клиент и сервер оба написаны на Java)
- Расширение интерфейса пользователя
  - Организация соревнований между алгоритмами (матчи, турниры, ...)
  - Хранение архивов партий и соревнований в БД (реляционных (JDBC),...)
  - Редактор начальных позиций игр

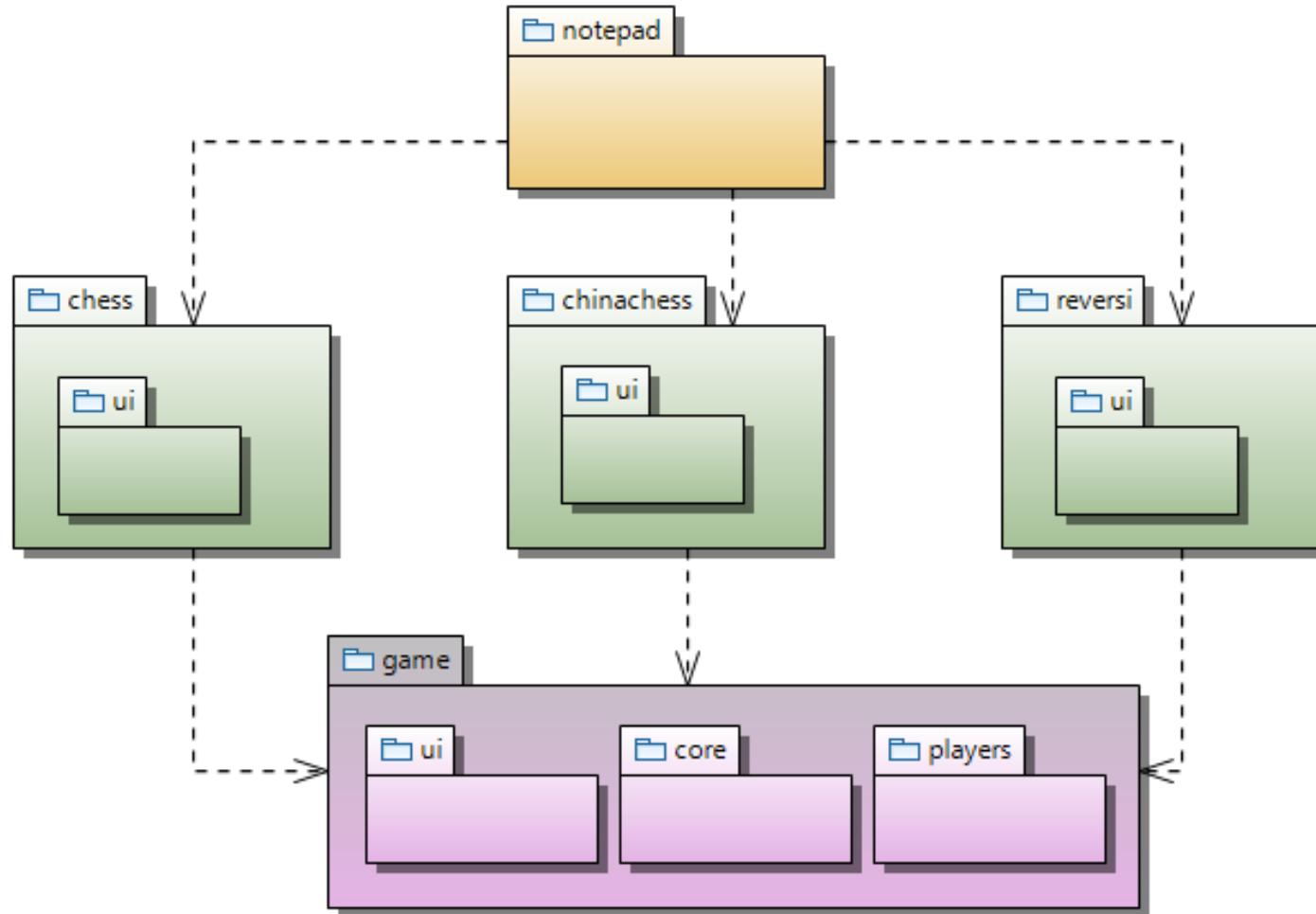
# Проектирование архитектуры.

## Выделение уровней в настольных играх

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

### Идентификация пакетов верхнего уровня

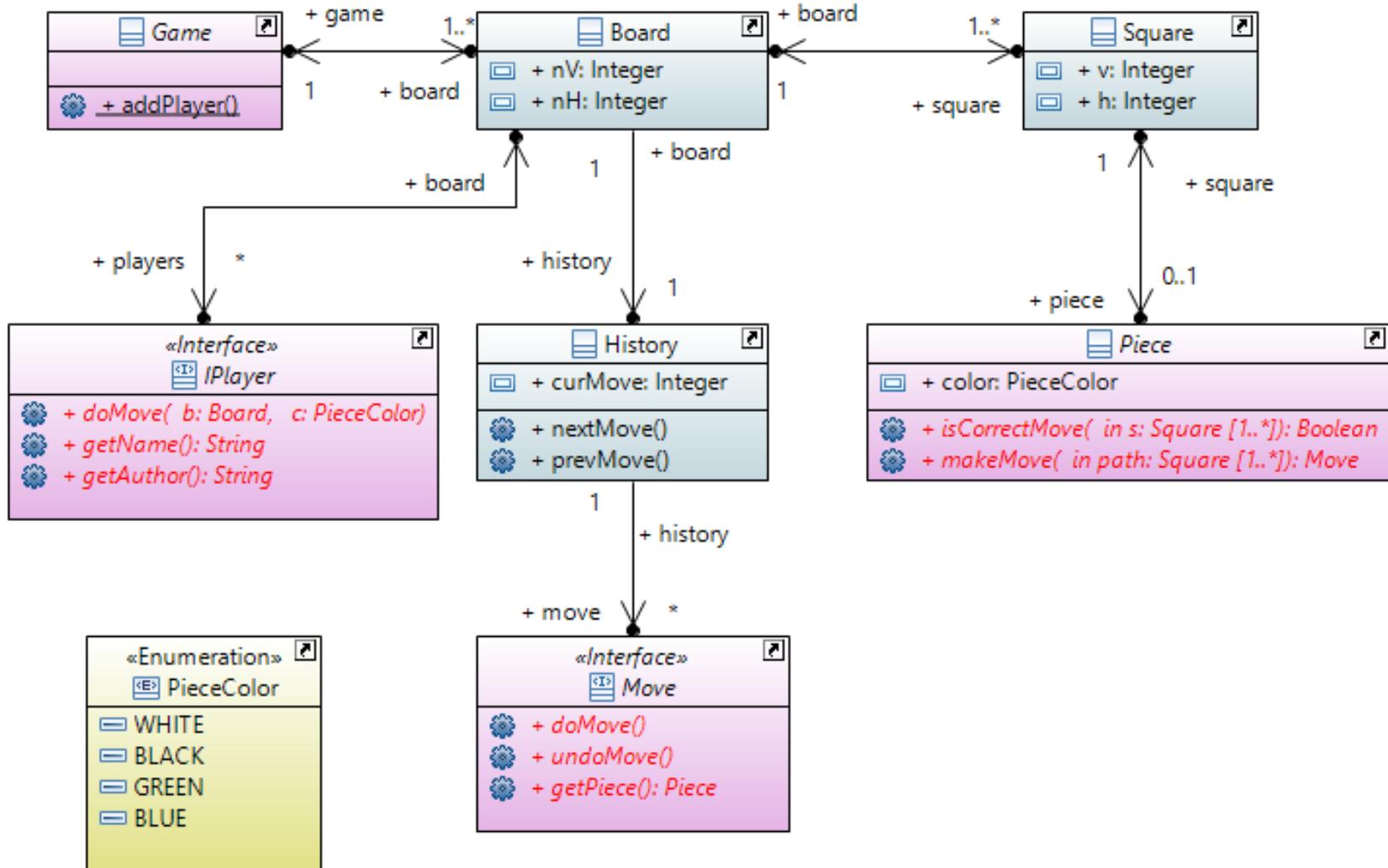


# Проектирование архитектуры.

## Ядро настольных игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

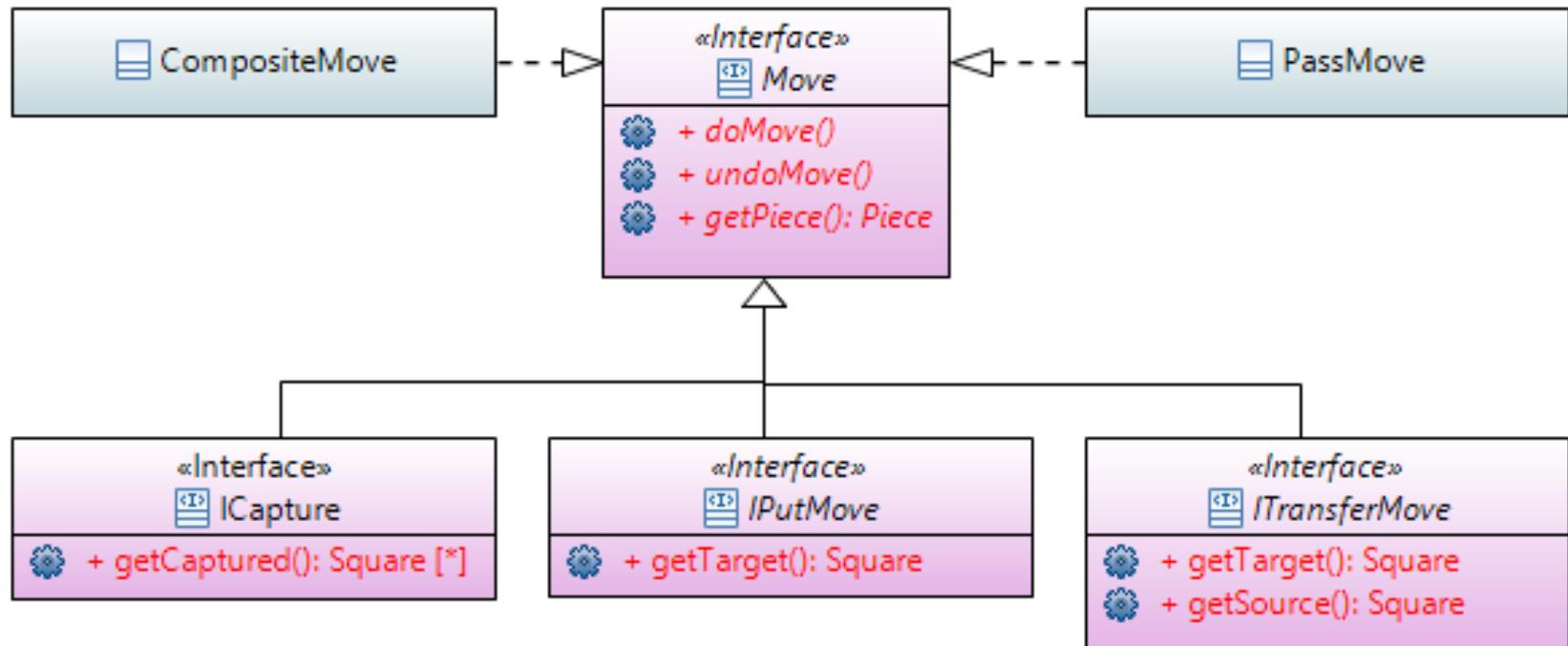


# Проектирование архитектуры.

## Ядро настольных игр. Ходы настольных игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

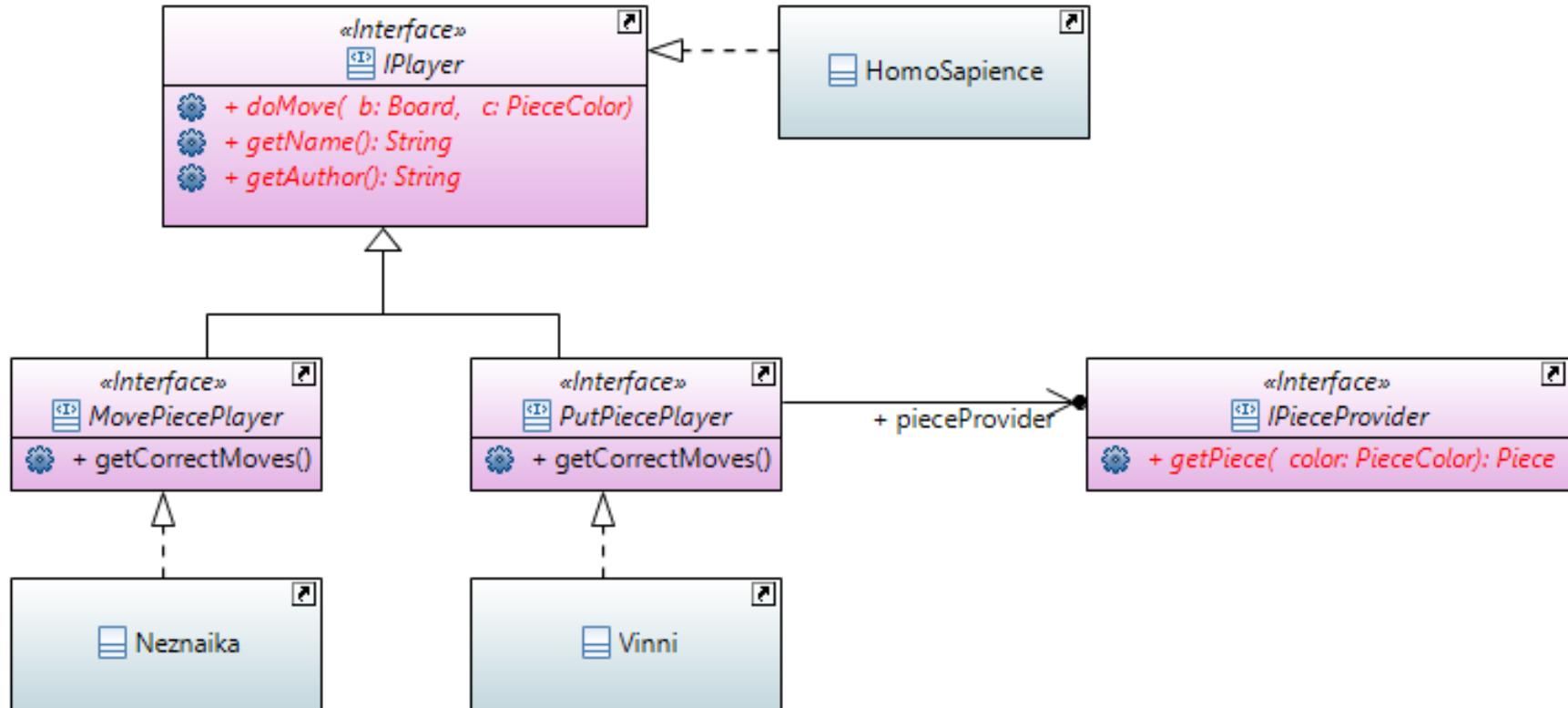


# Проектирование архитектуры.

## Ядро настольных игр. Игроки настольных игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



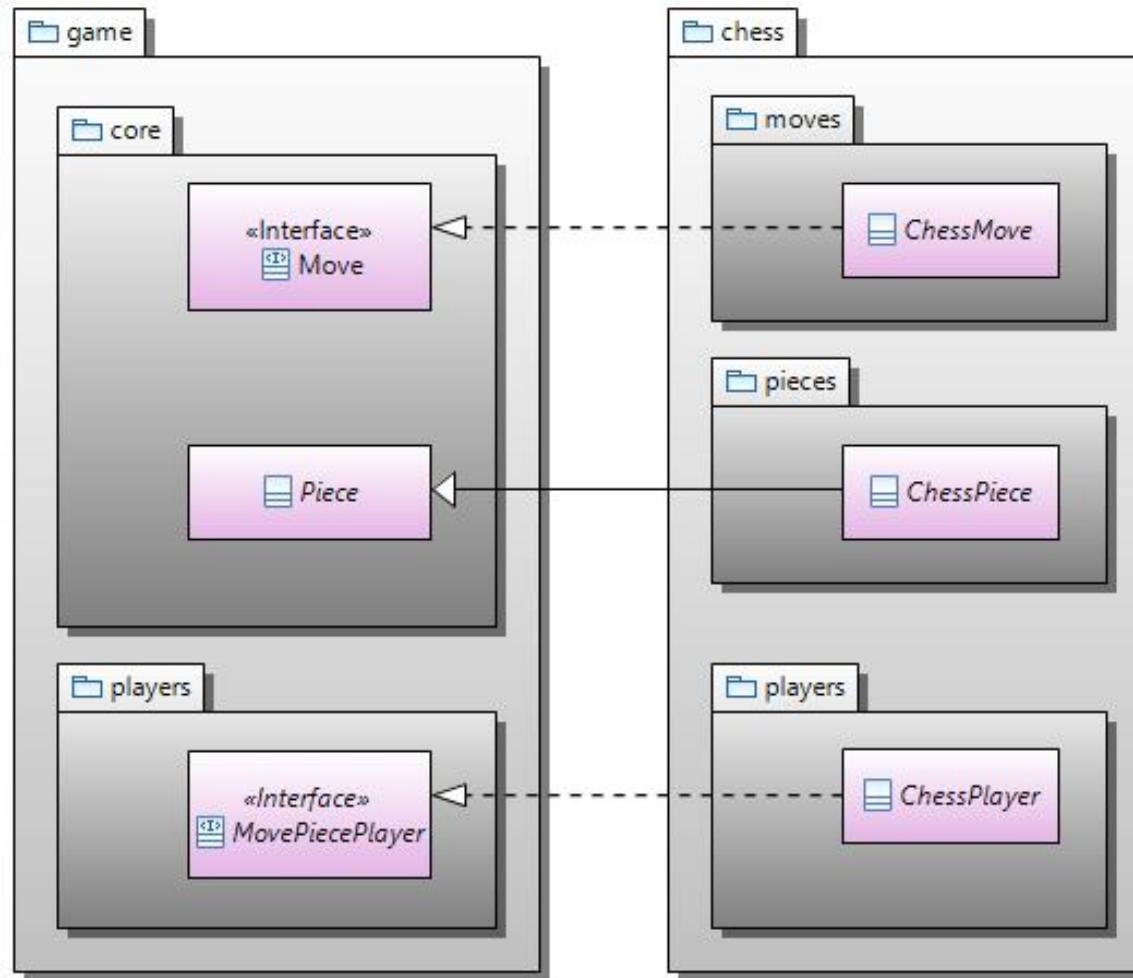
# Проектирование архитектуры

## Зависимости между пакетами.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Идентификация зависимостей между пакетами.

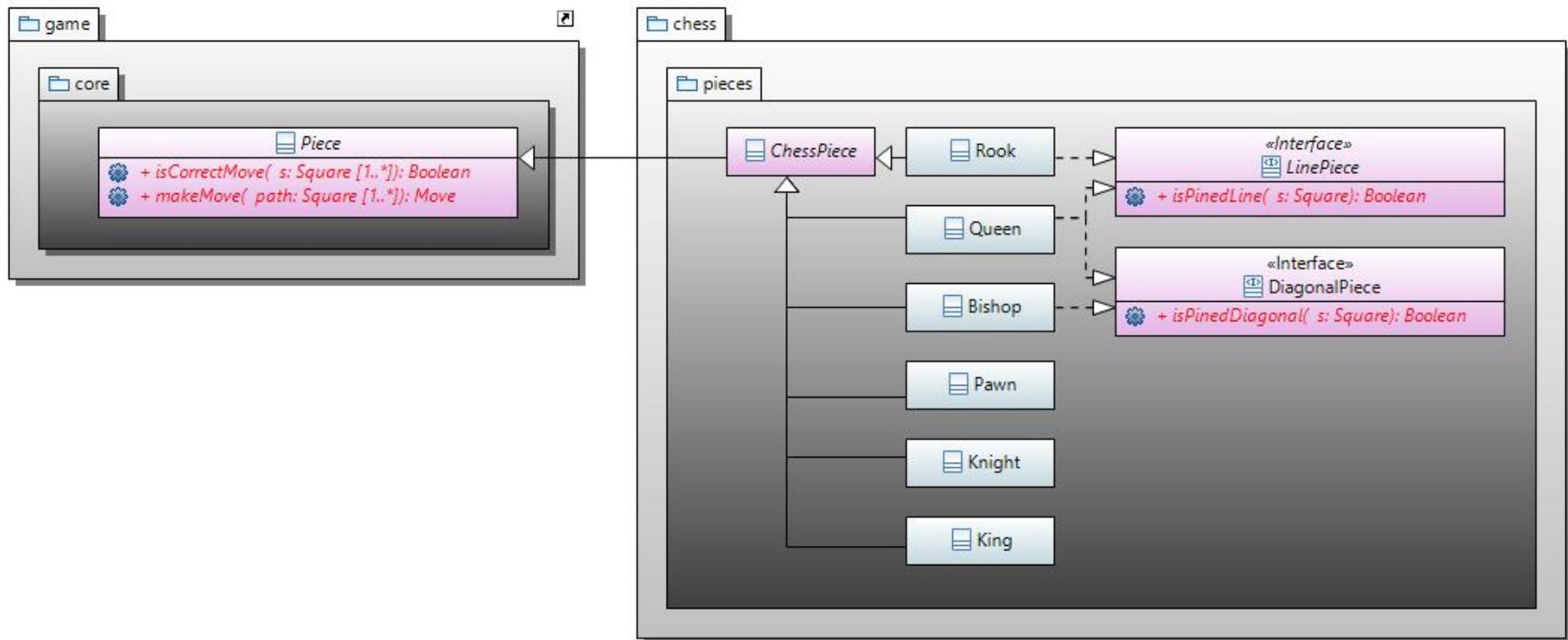


# Проектирование классов

## Классы-фигуры в шахматной программе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

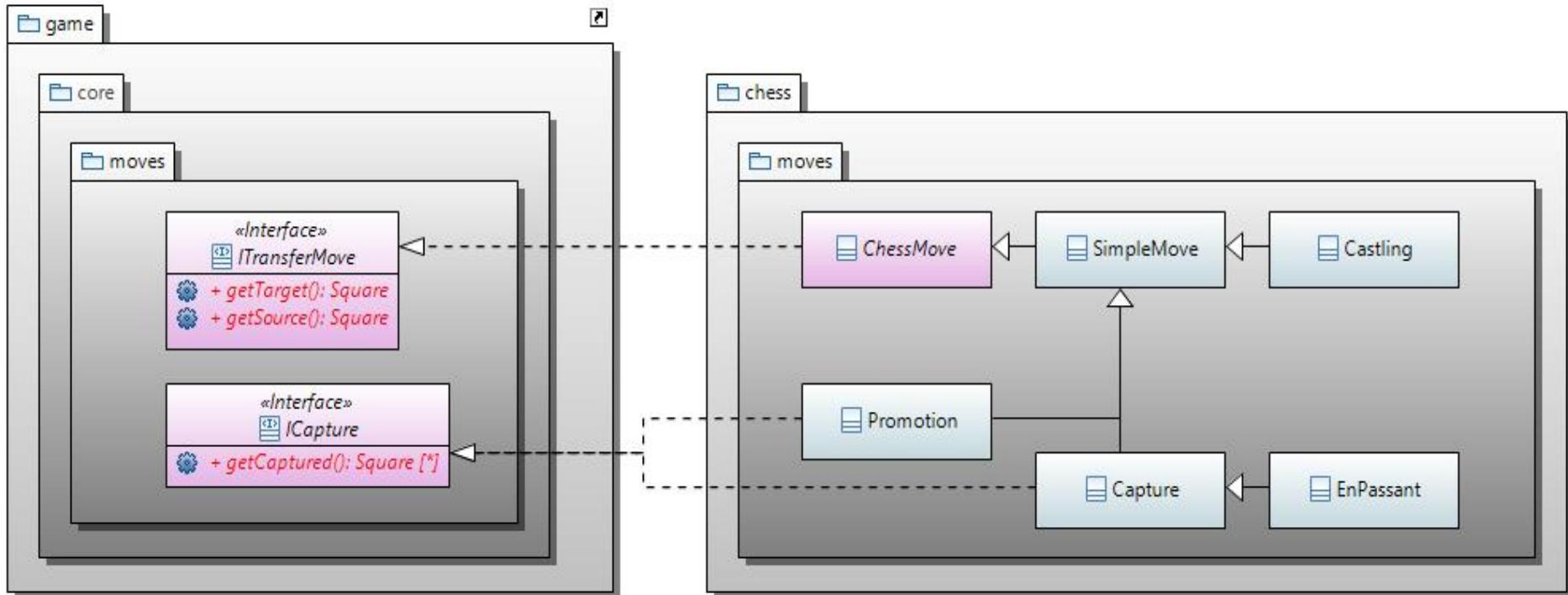


# Проектирование классов

## Классы-ходы в шахматной программе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

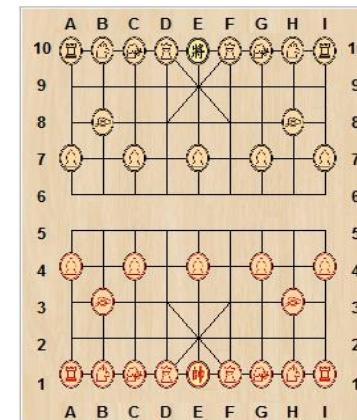
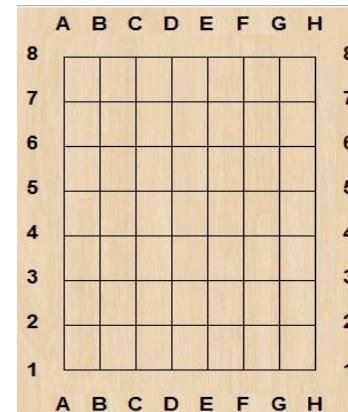
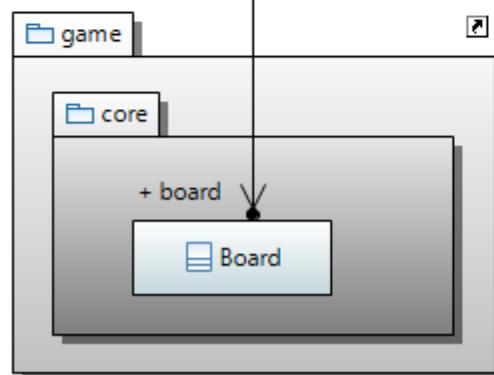
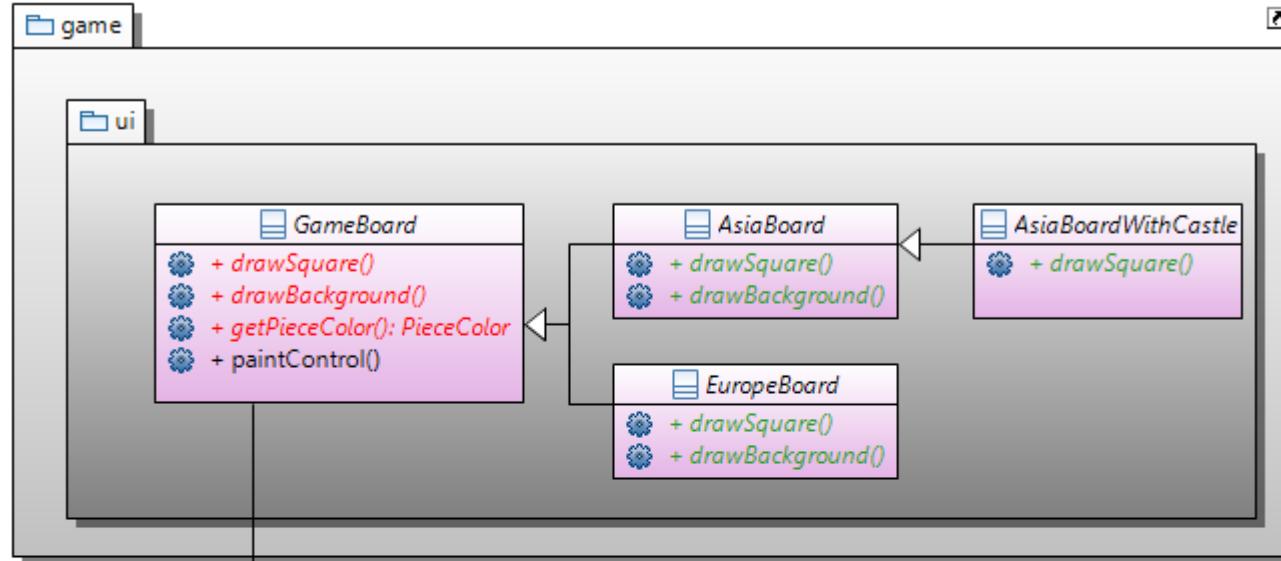


# Проектирование интерфейса пользователя

## Интерфейс пользователя для игры на доске.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

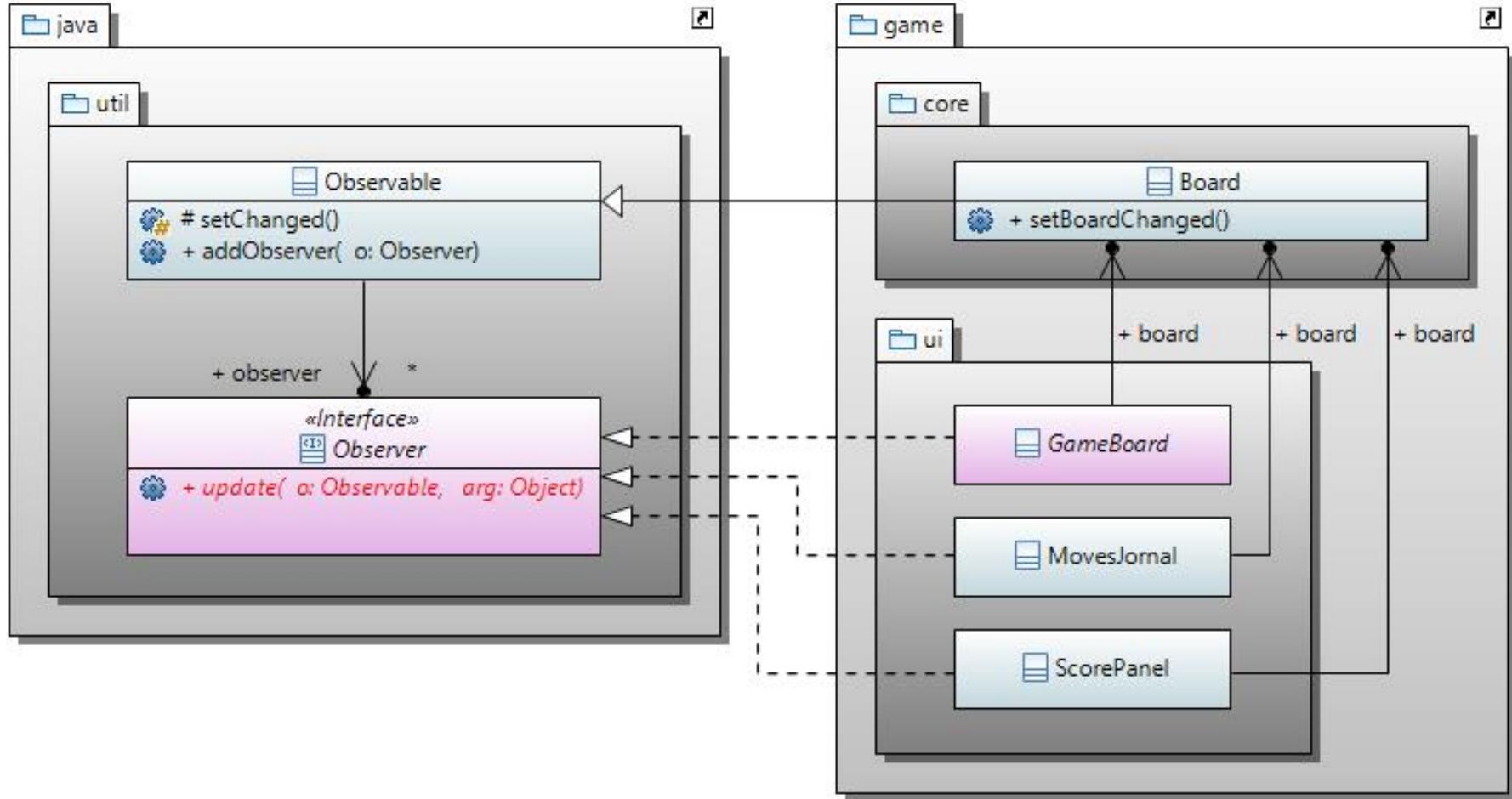


# Проектирование интерфейса пользователя

## Обозреваемый (доска) и обозреватели

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Проектирование интерфейса пользователя

## Обозреватели доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



ScorePanel

GameBoard

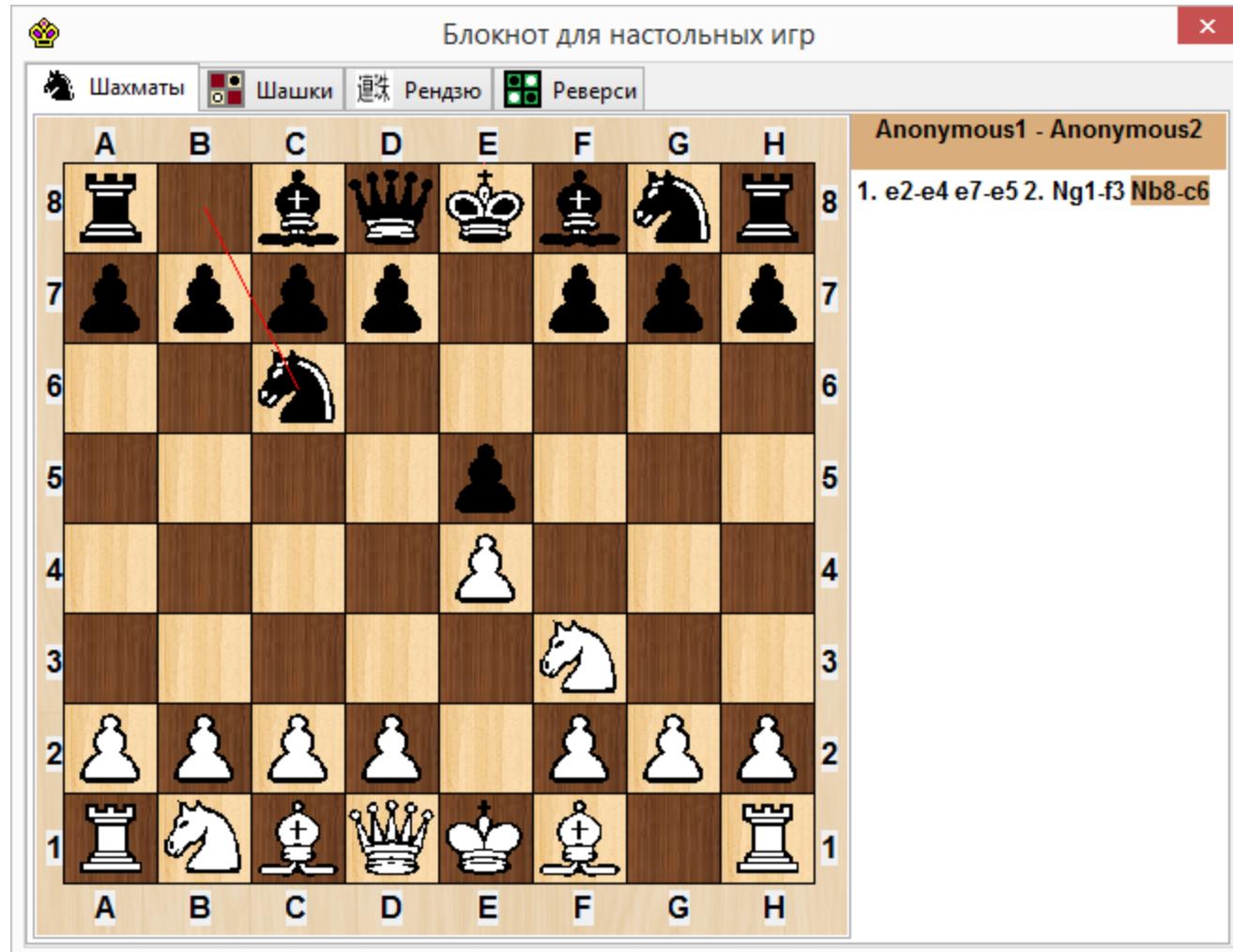
MovesJurnal

# Блокнот игр. Шахматы.

## Подсветка последнего хода

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

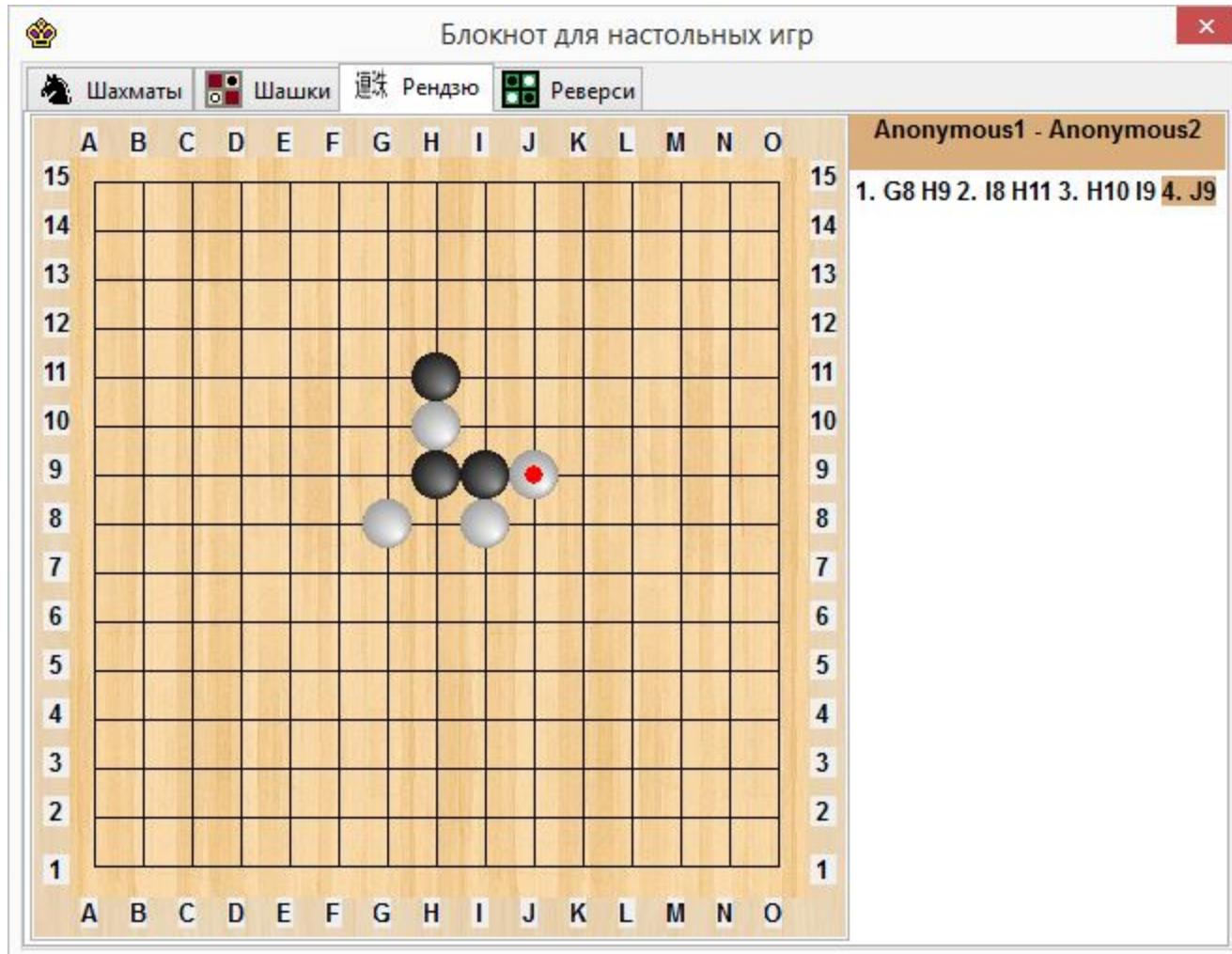


# Блокнот игр. Ренджю.

## Подсветка последнего хода

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

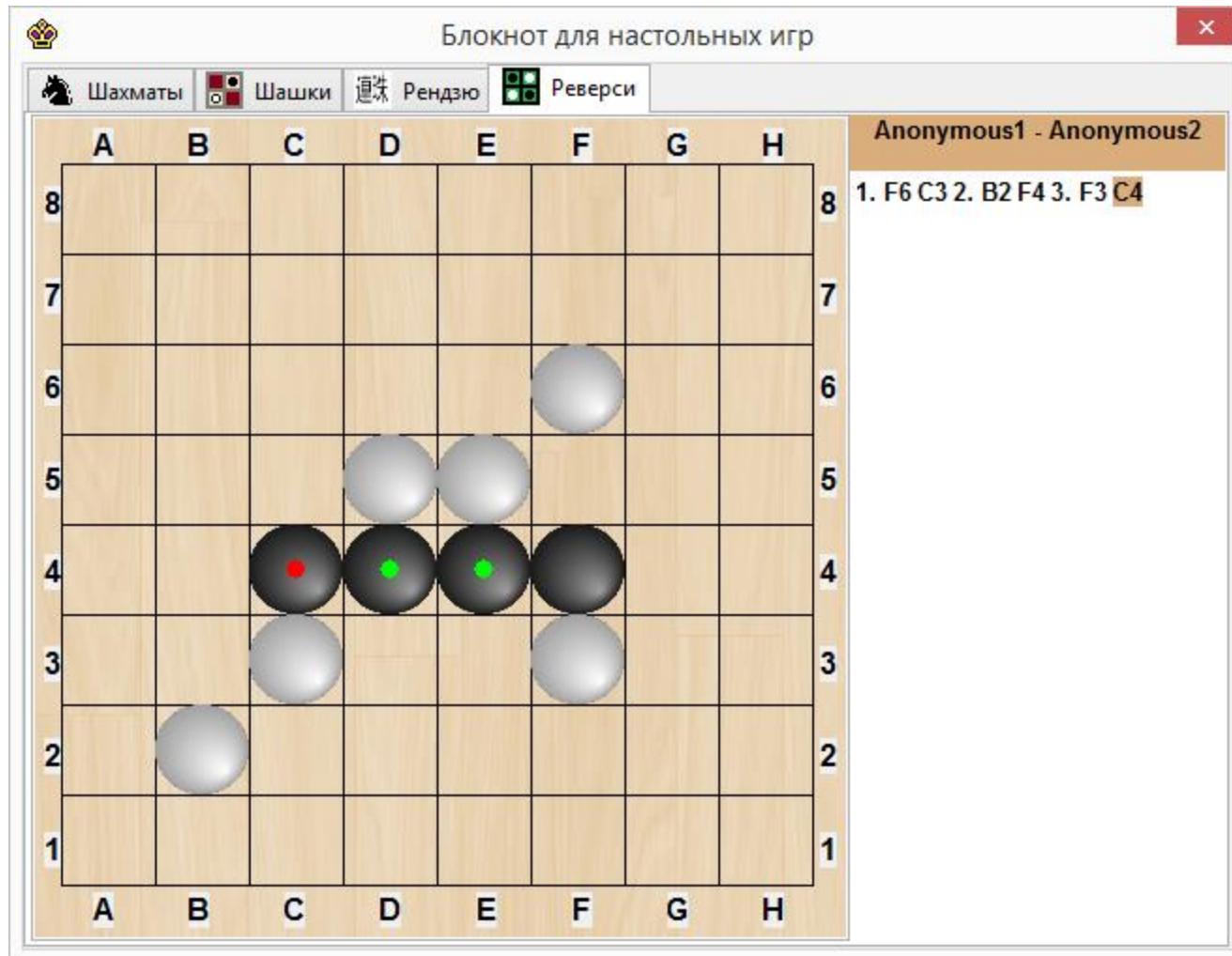


# Блокнот игр. Реверси.

## Подсветка последнего хода

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Назначение SWT

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- **Standard Widget Toolkit:** Инструмент для разработки интерфейса пользователя
  - Разработан в IBM
  - Используется в проектах [www.eclipse.org](http://www.eclipse.org)
  - Эффективный
  - Переносимый
  - Низкий уровень реализации: есть доступ к возможностям операционной системы
- Сайт проекта: [eclipse.org/swt](http://eclipse.org/swt)

# Элементы библиотеки SWT

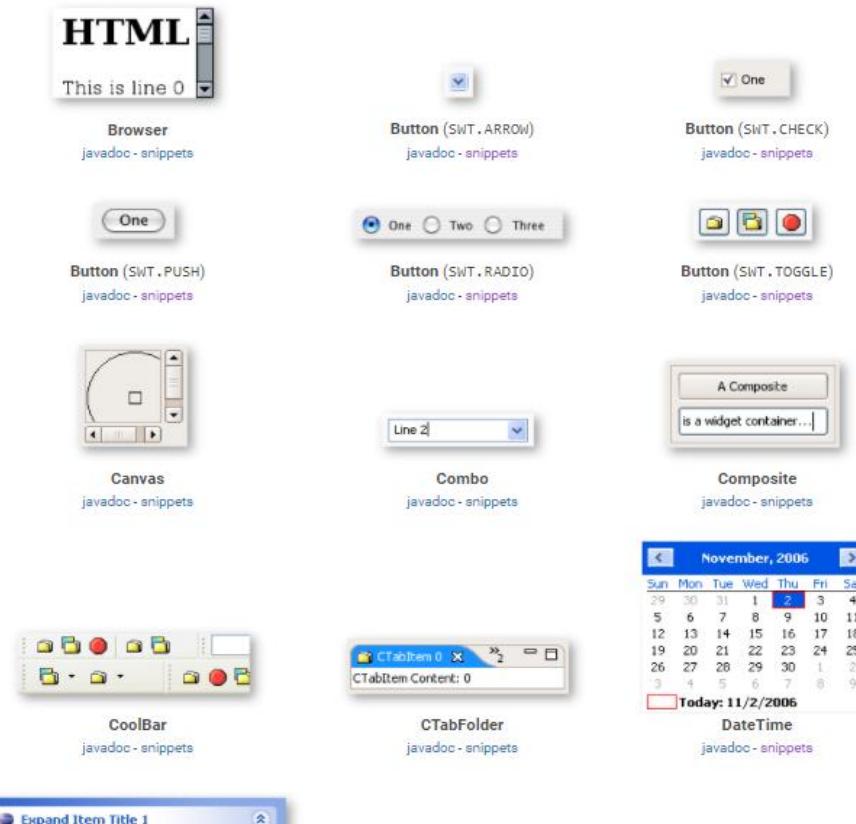
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Сайт проекта: <https://www.eclipse.org/swt/widgets/>

## SWT Widgets

Below are screenshots and links to documentation for many of the widgets included in SWT. For a complete list of classes including those that don't screenshot well, see the [SWT Javadoc](#).



# Фрагменты кода для использования элементов библиотеки SWT

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Сайт проекта: <https://www.eclipse.org/swt/snippets/>

## SWT Snippets

Snippets are minimal stand-alone programs that demonstrate specific techniques or functionality. Often a small example is the easiest way to understand how to use a particular feature. (If you are looking for large examples, like *ControlExample*, see the [SWT Examples](#), and if you are programming with JFace, you may find these [JFace Snippets](#) useful).

Snippets also help isolate problems. The best way to report an SWT bug is to write your own snippet showing the problem and paste it into the bug report. For a snippet template, see the "Hello World" example.

Note that the examples here are often edited for brevity rather than completeness. They are intended to guide the reader towards the correct solution, rather than be finished products. These snippets are tested against the HEAD stream and may sometimes reference new API or require bug fixes from there.

To run a snippet, simply import SWT into your Eclipse workspace, create a new Java project that depends on SWT, copy the desired snippet to the clipboard, and paste it into a new snippet class. (If you are using eclipse 3.2 M1 or earlier, you need to create the class using the New Class wizard before pasting; but since 3.2 M1 you can simply select your project and paste, and the class is created for you). Run by selecting the class and then selecting *Run > Run As > Java Application*.

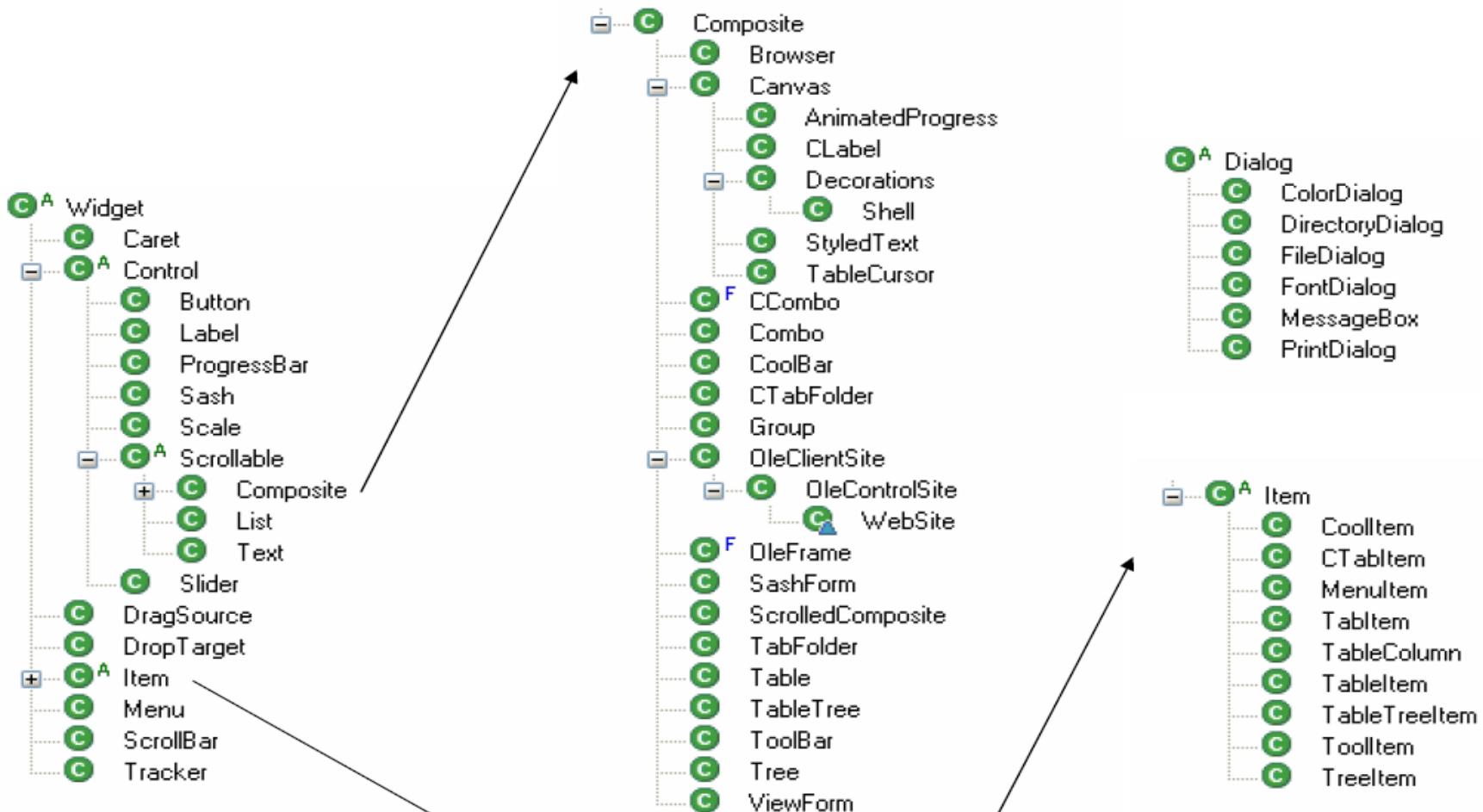
To contribute a new snippet, create a snippet contribution report in Bugzilla. Thanks in advance for your contribution!

- "Hello World"
  - "Hello World"
- Accessibility
  - using an accessible listener to provide state information – (preview)
  - provide text that will be spoken for an image button – (preview)
  - give accessible names to a tree and its tree items – (preview)
  - respond to text-based questions from an AT – (preview)
  - tell a screen reader about updates to a non-focused descriptive area – (preview)
  - use accessible relations to provide additional information to an AT – (preview)
  - provide a way for an AT to set text attributes in a StyledText – (preview)
  - declare a message area to be a "live region" – (preview)
- Browser
  - check if the browser is available or not – (preview)
  - bring up a browser (single window) – (preview)
  - bring up a browser with non-in blocker – (nreview)

# Иерархия управляющих элементов (widgets).

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# SWT. Основные классы библиотеки:

## Класс **Display**

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Представляет рабочее место (*workstation*) мониторы, клавиатуру, мышку
- Отвечает за распределение событий в цикле событий (*event loop*)
- Содержит список окон верхнего уровня (*Shells*)
- Содержит список мониторов (*Monitors*)

# SWT. Основные классы библиотеки:

## Класс **Shell**

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

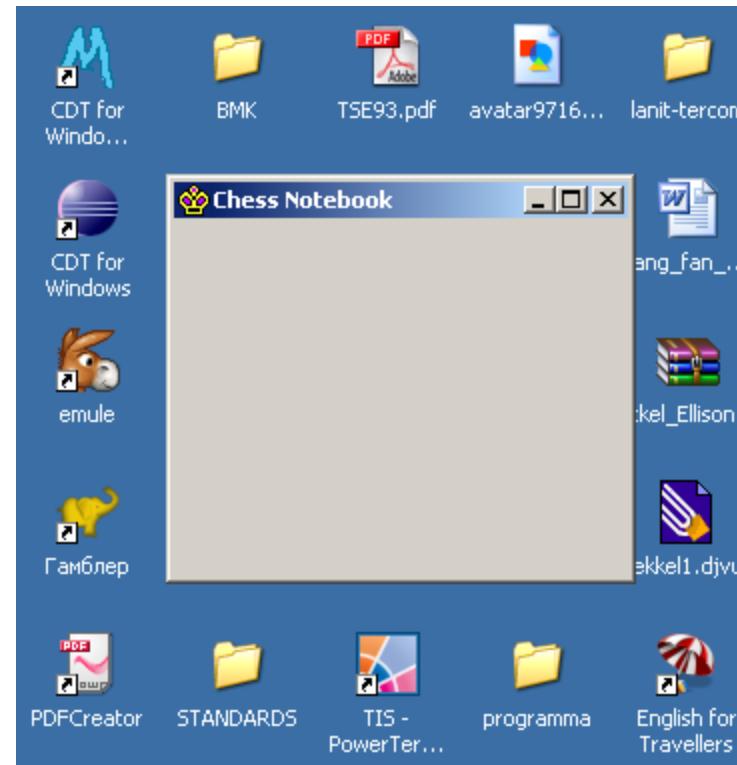
- Представляет окно на экране
- Это корень дерева состоящего из:
  - *Composites* (составной элемент)
  - *Controls* (управляющий элемент)
- **Shell** – потомок класса **Widget**

# SWT. Вид класса **Shell** на экране.

## Окно с заголовком *Chess Notebook*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# SWT. Основные классы библиотеки:

## Класс **Composite**

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- **Composite** (составной элемент) - управляющий элемент (control) который может состоять из других составных элементов и управляющий элементов
- **Composite** – потомок класса **Widget**

# SWT. Основные классы библиотеки:

## Класс **Control**

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- **Control** (управляющий элемент) – представляет легковесный (*heavyweight*) элемент операционной системы
- Примеры управляющих элементов: Button, Label, Text, Tree, Shell, Composite,  
...
- **Control** – потомок класса **Widget**

# SWT. Цикл событий (event loop)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- В **SWT**, цикл событий должен быть явно закодирован в приложении
- Цикл событий постоянно *читает и распределяет события* интерфейса пользователя поступающие из операционной системы и «отдает» СРУ когда событий нет.
- Цикл событий завершается когда завершается приложение. Обычно когда закрывается окно.

# SWT. Пример цикла событий

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
while( !shell.isDisposed() ) {  
    if( !display.readAndDispatch() )  
        display.sleep();  
}
```

# Шахматы. Минимальное «приложение»

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
import org.eclipse.swt.widgets.*;

public class Chess {
    public static void main(String[] args) {
        final Display display = new Display();

        final Shell shell = new Shell(display);
        shell.setSize(600, 500);
        shell.setText("Chess Notebook");
        shell.open();

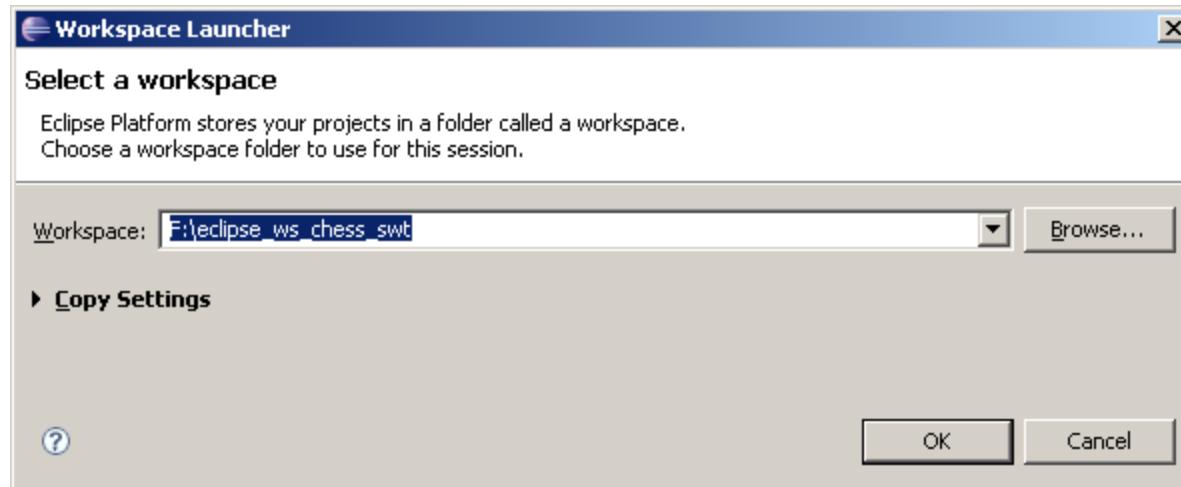
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        display.dispose();
    } // main
} // class Chess
```

# Шахматы. Создание рабочего места

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

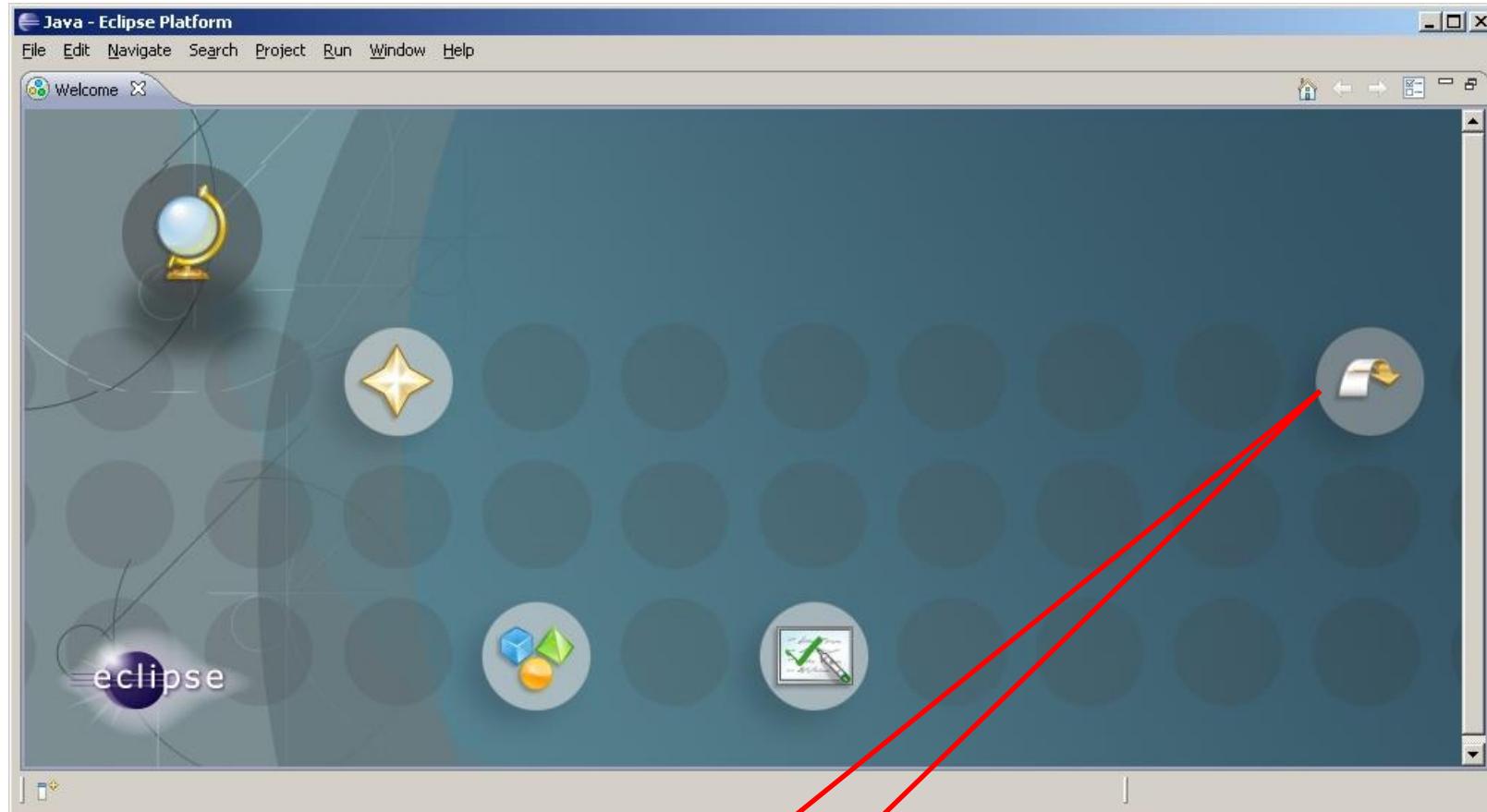
## ■ Files | Switch Workspace | Other ...



# Шахматы. Рабочее место. Wellcome

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

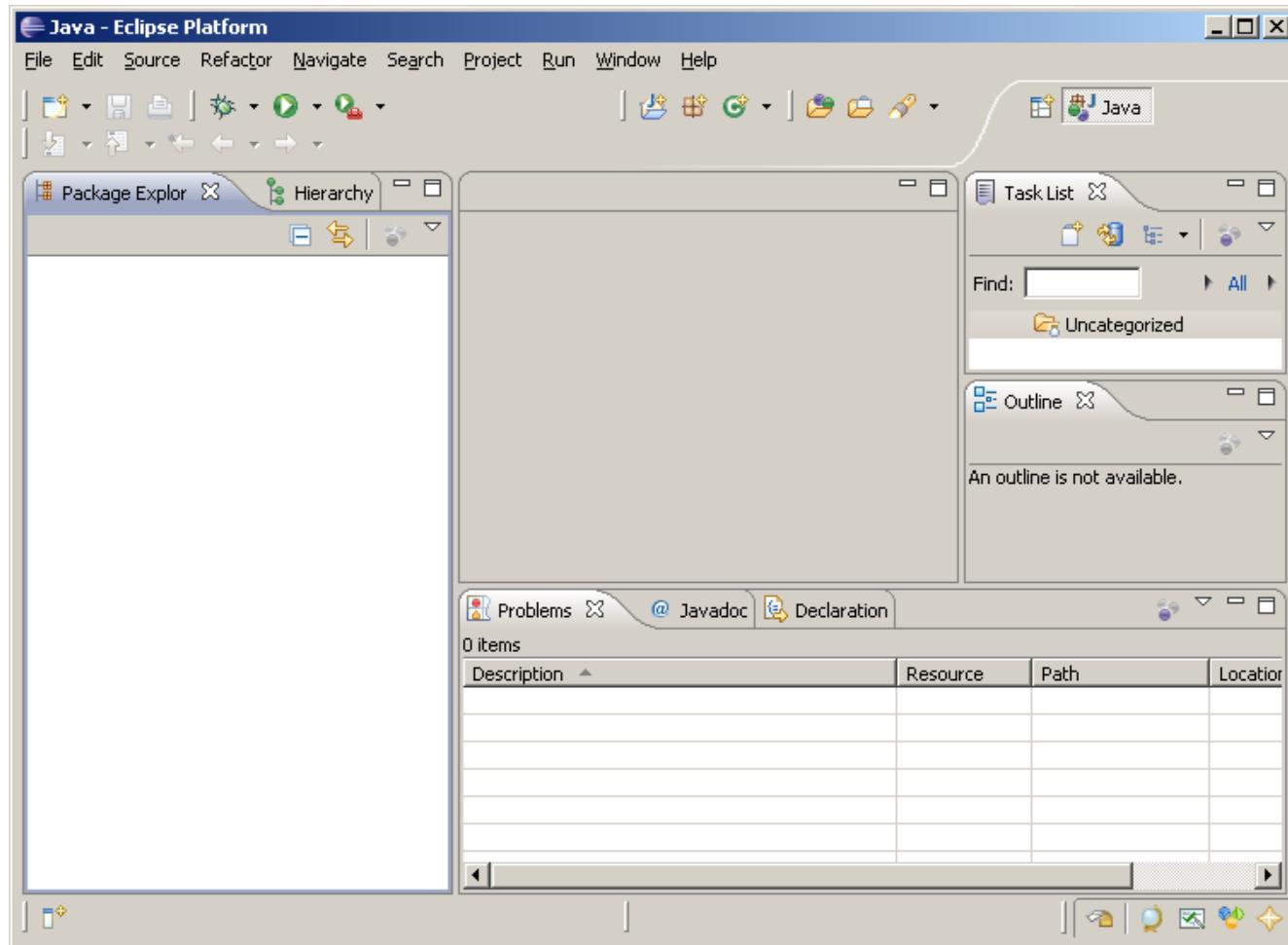


Нажать здесь

# Шахматы. Workbench (Станок ??)

МГУ им. М.В.Ломоносова. Факультет ВМК.

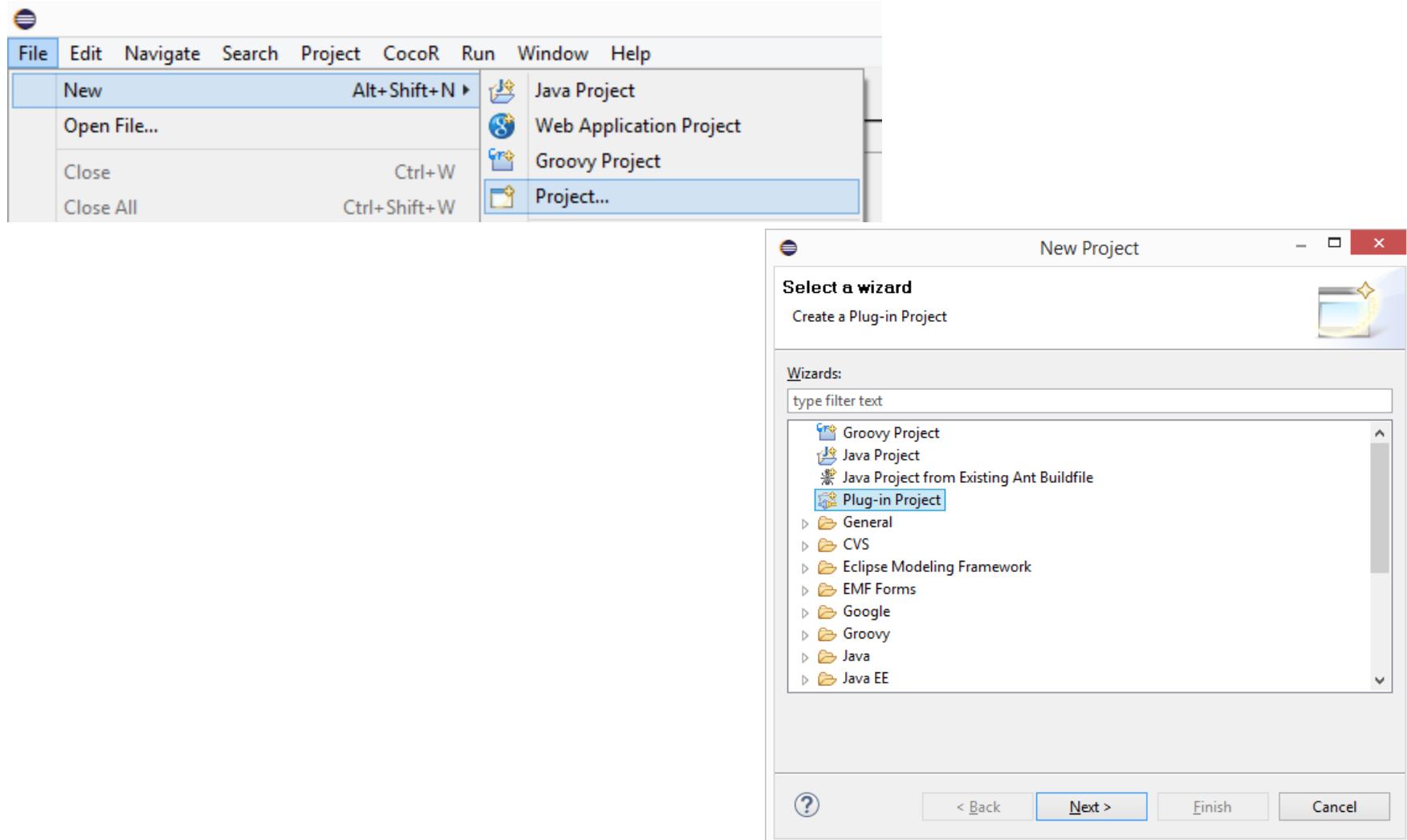
Романов Владимир Юрьевич ©2025



# Шахматы. Создание нового проекта

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

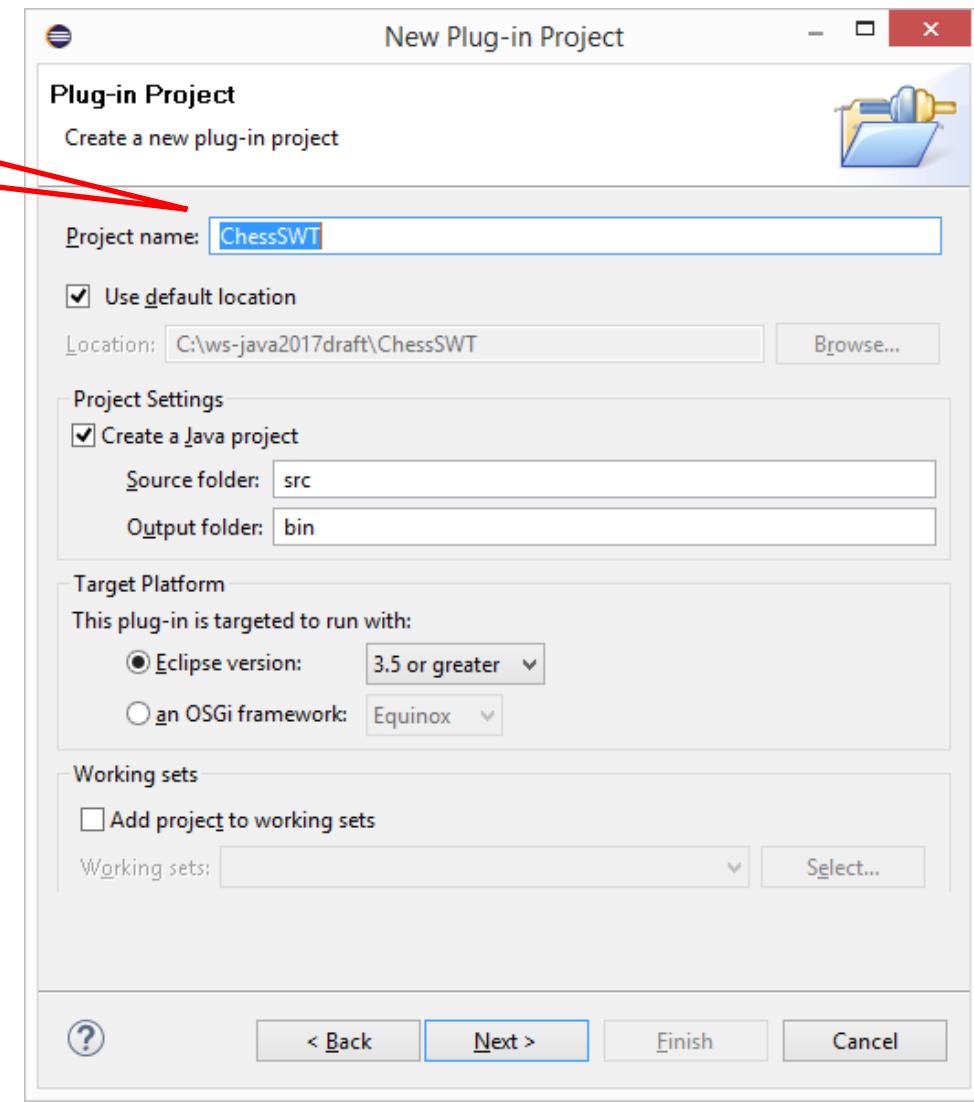


# Шахматы. Создание проекта

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

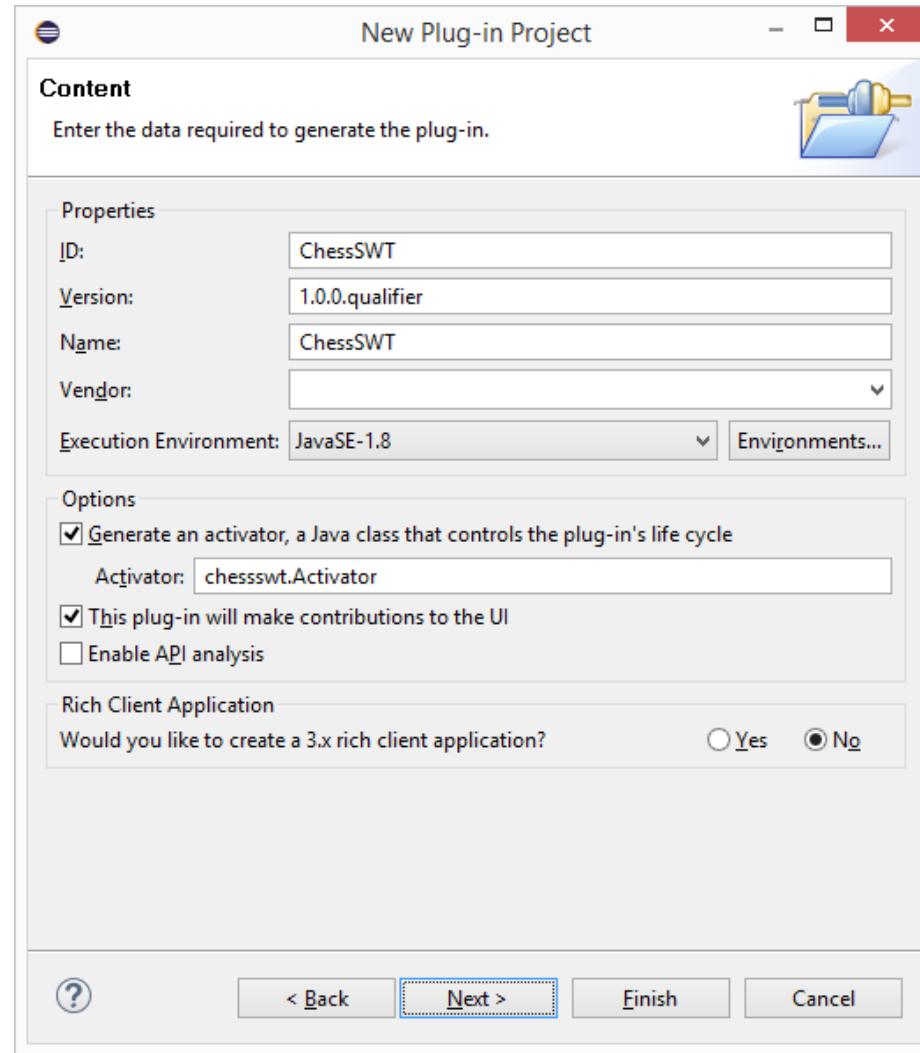
Ввести имя проекта  
**ChessSWT**



# Шахматы. Создание проекта

МГУ им. М.В.Ломоносова. Факультет ВМК.

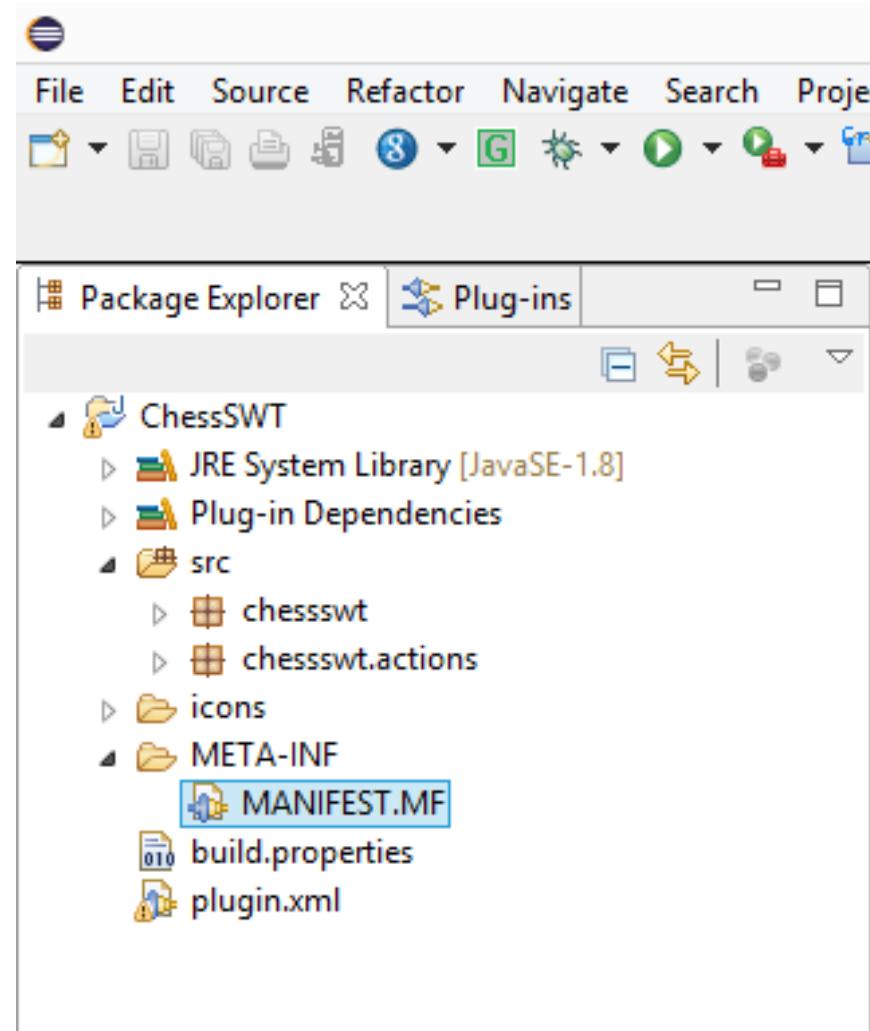
Романов Владимир Юрьевич ©2025



# Шахматы. Создание проекта

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Шахматы. Создание “главного” класса

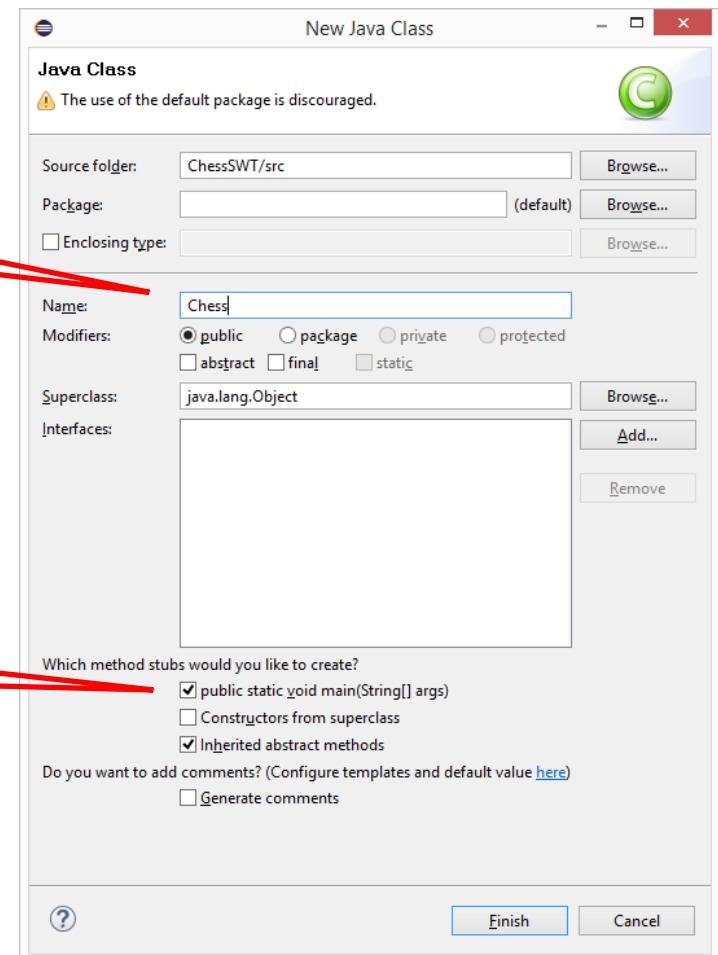
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

## ■ Files | New | Class ...

Ввести имя класса  
**Chess**

Попросить создать  
метод main()



# Шахматы. Сгенерированный «главный» класс **Chess**

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure for "ChessSWT". It includes:
  - JRE System Library [JavaSE-1.8]
  - Plug-in Dependencies
  - src
    - (default package)
      - Chess.java
      - chesswt
      - chesswt.actions
    - icons
  - META-INF
    - MANIFEST.MF
  - build.properties
  - plugin.xml
- Code Editor View:** Displays the content of the "Chess.java" file.

```
1 public class Chess {  
2  
3  
4     public static void main(String[] args) {  
5         // TODO Auto-generated method stub  
6  
7     }  
8  
9 }  
10
```
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and run.
- MenuBar:** Includes File, Edit, Source, Refactor, Navigate, Search, Project, CocoR, Run, Window, Help.
- Title Bar:** Shows "Java - ChessSWT/src/Chess.java - Eclipse".

# Шахматы. Минимальное приложение

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
import org.eclipse.swt.widgets.*;  
  
public class Chess {  
    public static void main(String[] args) {  
        final Display display = new Display();  
  
        final Shell shell = new Shell(display);  
        shell.setSize(600, 500);  
        shell.setText("Chess Notebook");  
        shell.open();  
        while (!shell.isDisposed()) {  
            if (!display.readAndDispatch())  
                display.sleep();  
        }  
        display.dispose();  
    } // main  
} // class Chess
```

# Шахматы. «Минимальные» шахматы в окне редактора

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - ChessSWT/src/Chess.java - Eclipse
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, CocoR, Run, Window, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and various project management tools.
- Package Explorer View:** Shows the project structure:
  - ChessSWT
  - JRE System Library [JavaSE]
  - Plug-in Dependencies
  - src
    - (default package)
      - Chess.java
      - chesswt
      - chesswt.actions
    - icons
  - META-INF
    - MANIFEST.MF
  - build.properties
  - plugin.xml
- Editor View:** Displays the Java code for the `Chess.java` file:

```
1 import org.eclipse.swt.widgets.*;
2
3 public class Chess {
4     public static void main(String[] args) {
5         final Display display = new Display();
6
7         final Shell shell = new Shell(display);
8         shell.setSize(600, 500);
9         shell.setText("Chess Notebook");
10        shell.open();
11        while (!shell.isDisposed()) {
12            if (!display.readAndDispatch())
13                display.sleep();
14        }
15        display.dispose();
16    } // main
17 } // class Chess
```
- Problems View:** Shows 0 errors, 3 warnings, and 0 others.

Description	Resource	Path	Location
⚠ Warnings (3 items)			
- Sign in to Google...** button at the bottom left.

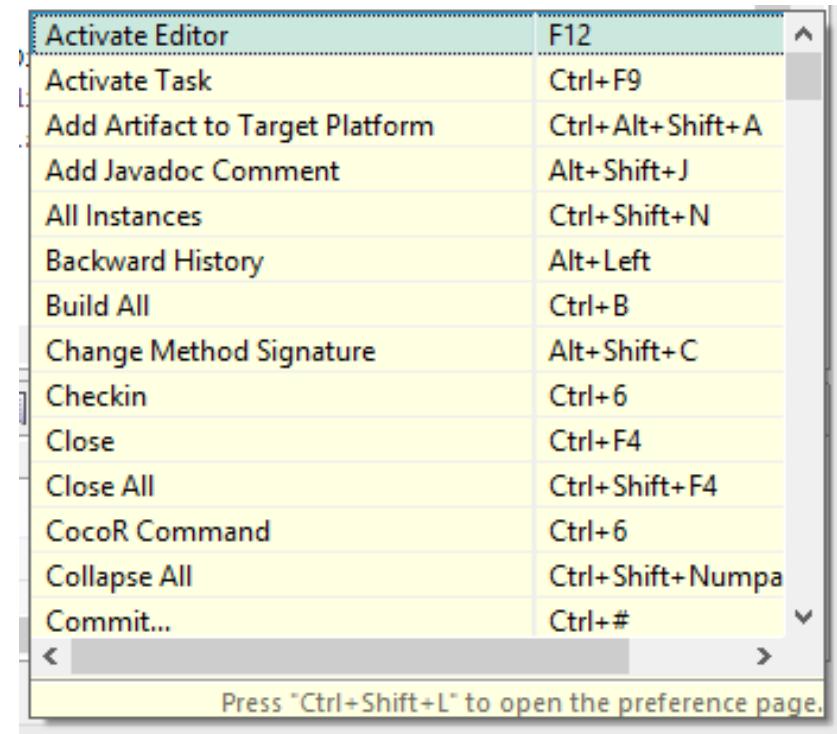
# Шахматы. Горячие клавиши редактора

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

**Ctrl + Shift + L**

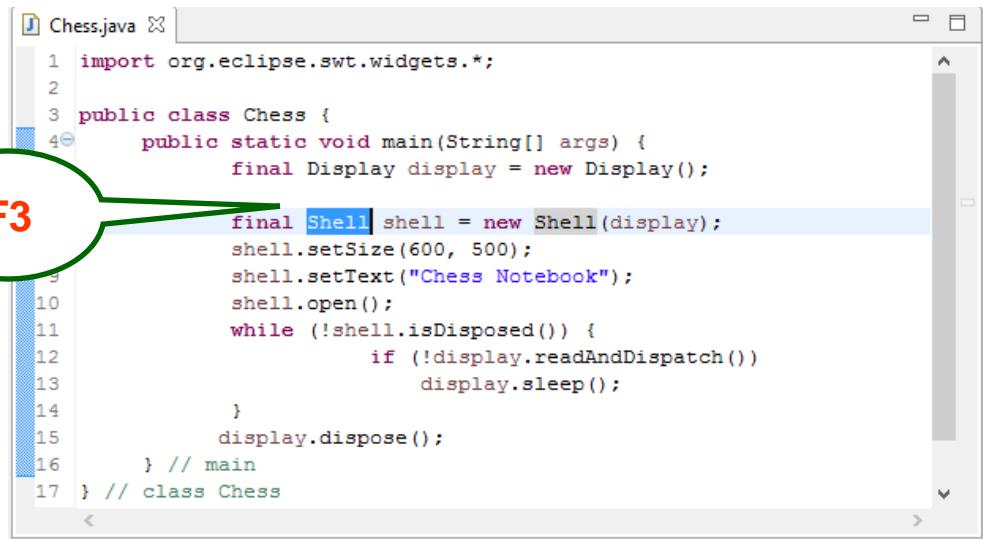
показать список всех «горячих клавиш»



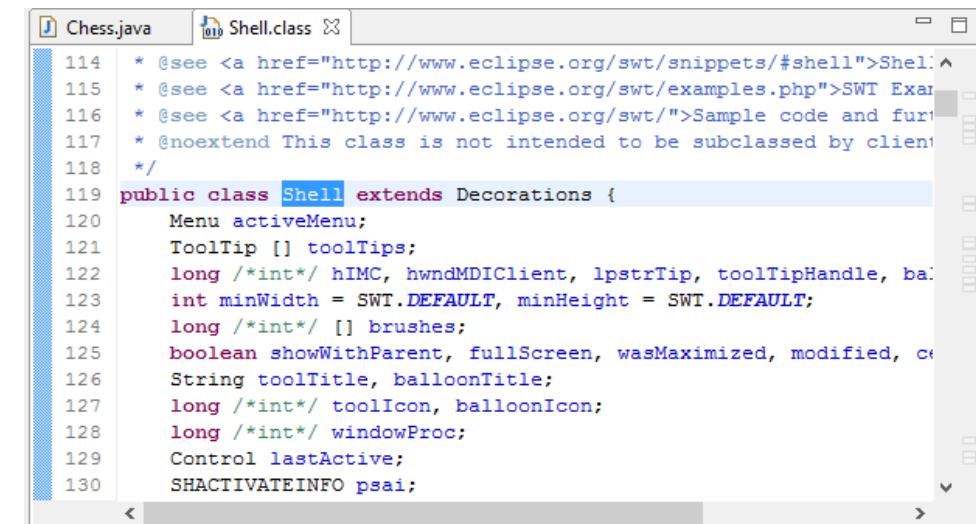
# Шахматы. Открытие исходных текстов горячей клавишей F3

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



```
Chess.java
1 import org.eclipse.swt.widgets.*;
2
3 public class Chess {
4     public static void main(String[] args) {
5         final Display display = new Display();
6
7         final Shell shell = new Shell(display);
8         shell.setSize(600, 500);
9         shell.setText("Chess Notebook");
10        shell.open();
11        while (!shell.isDisposed()) {
12            if (!display.readAndDispatch())
13                display.sleep();
14        }
15        display.dispose();
16    } // main
17 } // class Chess
```

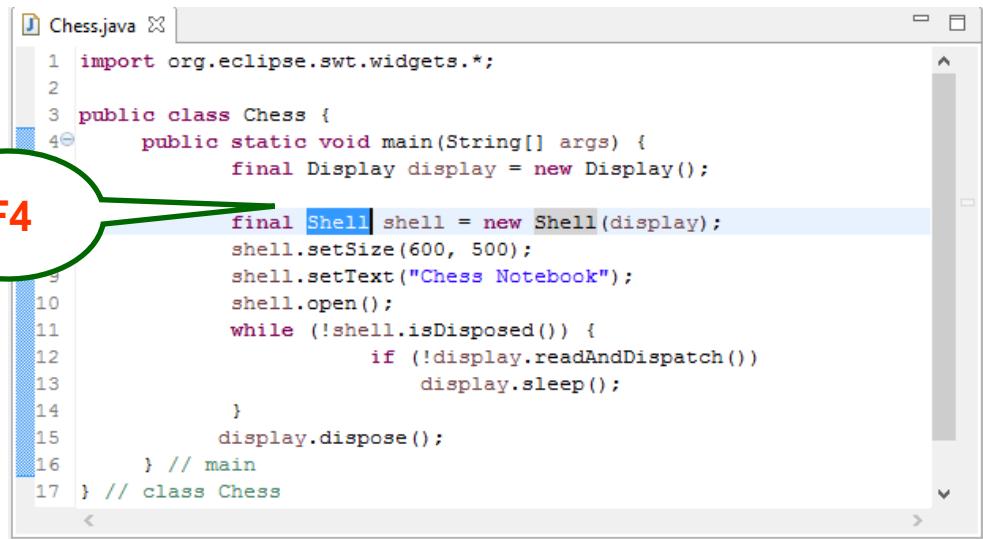


```
Shell.class
114 * @see <a href="http://www.eclipse.org/swt/snippets/#shell">Shell
115 * @see <a href="http://www.eclipse.org/swt/examples.php">SWT Examples
116 * @see <a href="http://www.eclipse.org/swt/">Sample code and further resources
117 * @noextend This class is not intended to be subclassed by clients
118 */
119 public class Shell extends Decorations {
120     Menu activeMenu;
121     ToolTip [] toolTips;
122     long /*int*/ hIMC, hwndMDIClient, lpstrTip, toolTipHandle, balloonHandle;
123     int minWidth = SWT.DEFAULT, minHeight = SWT.DEFAULT;
124     long /*int*/ [] brushes;
125     boolean showWithParent, fullScreen, wasMaximized, modified, canTrayIcon;
126     String toolTitle, balloonTitle;
127     long /*int*/ toolIcon, balloonIcon;
128     long /*int*/ windowProc;
129     Control lastActive;
130     SHACTIVATEINFO psai;
```

# Шахматы. Открытие иерархии наследования горячей клавишей F4

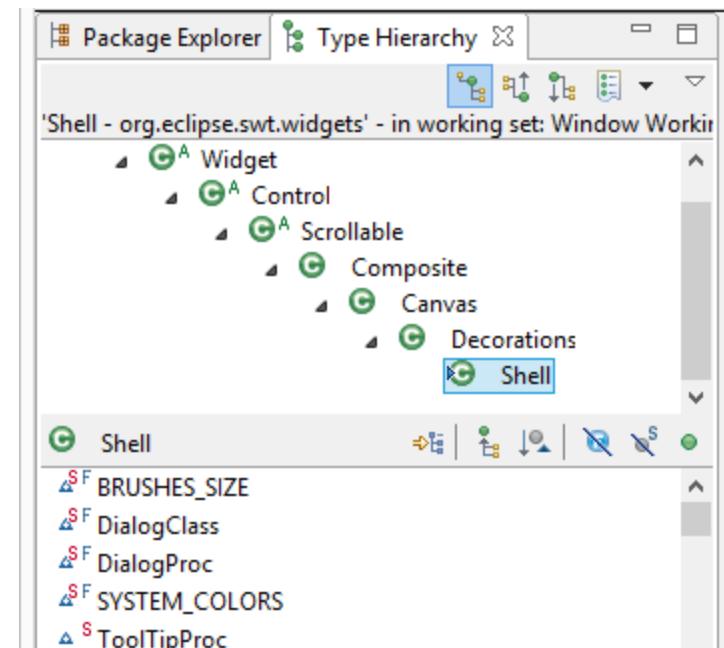
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



```
Chess.java X
1 import org.eclipse.swt.widgets.*;
2
3 public class Chess {
4     public static void main(String[] args) {
5         final Display display = new Display();
6
7         final Shell shell = new Shell(display);
8         shell.setSize(600, 500);
9         shell.setText("Chess Notebook");
10        shell.open();
11        while (!shell.isDisposed()) {
12            if (!display.readAndDispatch())
13                display.sleep();
14        }
15        display.dispose();
16    } // main
17 } // class Chess
```

F4

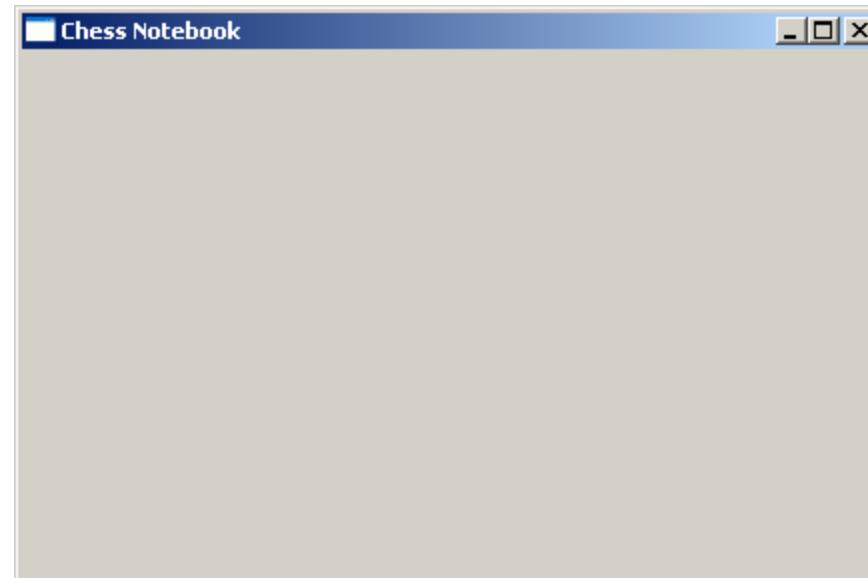


# Шахматы. «Минимальные» шахматы на экране

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Нажать “запуск в отладочном режиме” F11



# Пиктограмма приложения

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

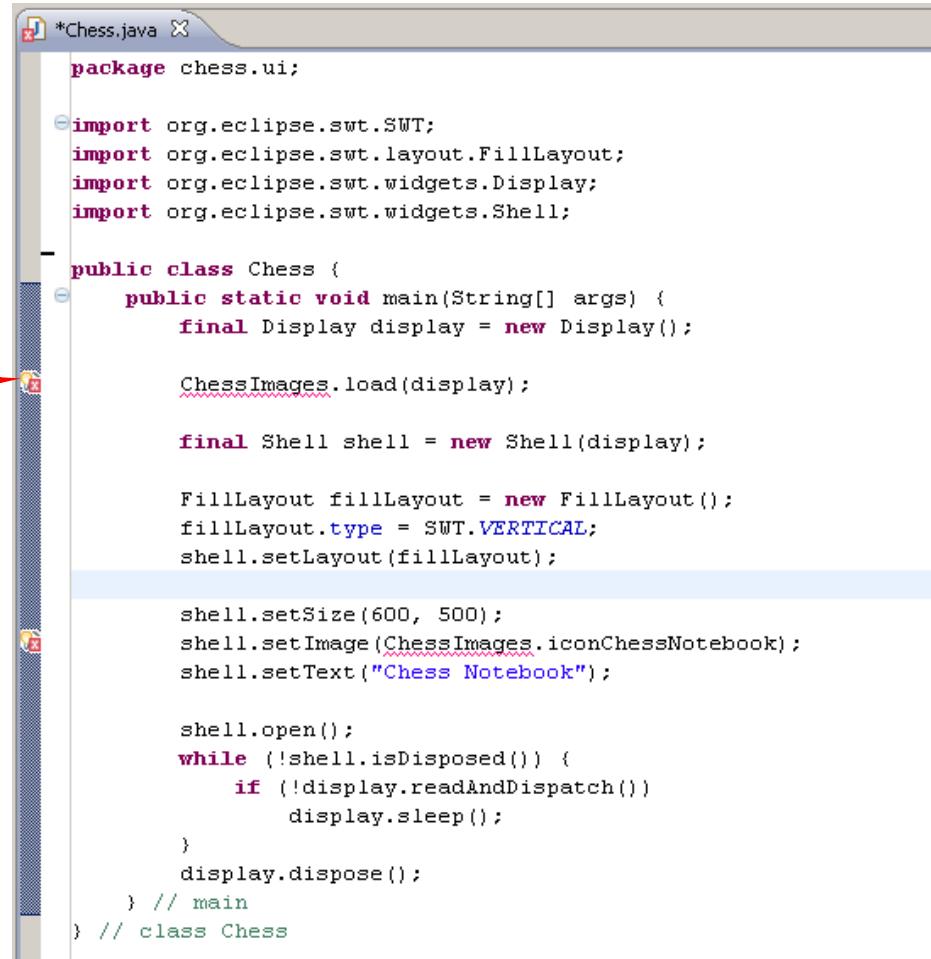
```
import org.eclipse.swt.widgets.*;  
  
public class Chess {  
    public static void main(String[] args) {  
        final Display display = new Display();  
        ChessImages.load(display);  
        final Shell shell = new Shell(display);  
        shell.setSize(600, 500);  
        shell.setText("Chess Notebook");  
        shell.setImage(ChessImages.iconChessNotebook);  
        shell.open();  
        while (!shell.isDisposed()) {  
            if (!display.readAndDispatch())  
                display.sleep();  
        }  
        display.dispose();  
    } // main  
} // class Chess
```

# Шахматы. Создание класса *ChessImages* в режиме «исправления ошибок»

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Нажатие мышью  
или  
**CTRL+1**



The screenshot shows the Eclipse IDE interface with a Java code editor window titled "Chess.java". The code implements a simple application for a chess notebook. It imports necessary SWT components and defines a main class "Chess" with a static main method. The method creates a display, loads chess images, creates a shell with a vertical fill layout, sets its size and text, and opens it in a loop until disposal. A red callout bubble points from the text "Нажатие мышью или CTRL+1" to the word "ChessImages" in the code, which is underlined by the IDE's error highlighting.

```
package chess.ui;

import org.eclipse.swt.SWT;
import org.eclipse.swt.layout.FillLayout;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Shell;

public class Chess {
    public static void main(String[] args) {
        final Display display = new Display();

        ChessImages.load(display);

        final Shell shell = new Shell(display);

        FillLayout fillLayout = new FillLayout();
        fillLayout.type = SWT.VERTICAL;
        shell.setLayout(fillLayout);

        shell.setSize(600, 500);
        shell.setImage(ChessImages.iconChessNotebook);
        shell.setText("Chess Notebook");

        shell.open();
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        display.dispose();
    } // main
} // class Chess
```

# Шахматы. Варианты «исправления ошибки» предлагаемые в среде Eclipse

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with a Java file named \*Chess.java open. The code defines a class Chess with a main method that creates a Display object and calls load() on a ChessImages object. A code completion tooltip is displayed over the call to load(). The tooltip lists several options for creating a new class 'ChessImages': 'Create class 'ChessImages'', 'Create constant 'ChessImages'', 'Create local variable 'ChessImages'', 'Create field 'ChessImages'', 'Create parameter 'ChessImages'', 'Rename in file (Ctrl+2, R direct access)', and 'Fix project setup...'. It also shows the package declaration and the start of the class definition. The tooltip includes a note: 'Press 'Tab' from proposal table or click for focus'.

```
package chess.ui;

import org.eclipse.swt.SWT;
import org.eclipse.swt.layout.FillLayout;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Shell;

public class Chess {
    public static void main(String[] args) {
        final Display display = new Display();

        ChessImages.load(display);
    }

    shell.open();
    while (!shell.isDisposed()) {
        if (!display.readAndDispatch())
            display.sleep();
    }
    display.dispose();
} // main
} // class Chess
```

# Шахматы. Создание метода *load*

## в классе *ChessImages*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with two tabs open: 'Chess.java' and 'ChessImages.java'. The code editor displays Java code for a chess application. In the 'Chess.java' tab, at the bottom of the main class definition, there is a call to 'ChessImages.load(display);'. A code completion tooltip is displayed over this line, listing two options:

- Create method 'load(Display)' in type 'ChessImage...
- Rename in file (Ctrl+2, R direct access)

The tooltip also contains a preview of the code for the new 'load' method:

```
...  
package chess.ui;  
  
import org.eclipse.swt.widgets.Display;  
public class ChessImages {  
    public static void load(Display display) {  
        // TODO Auto-generated method stub  
    }  
}
```

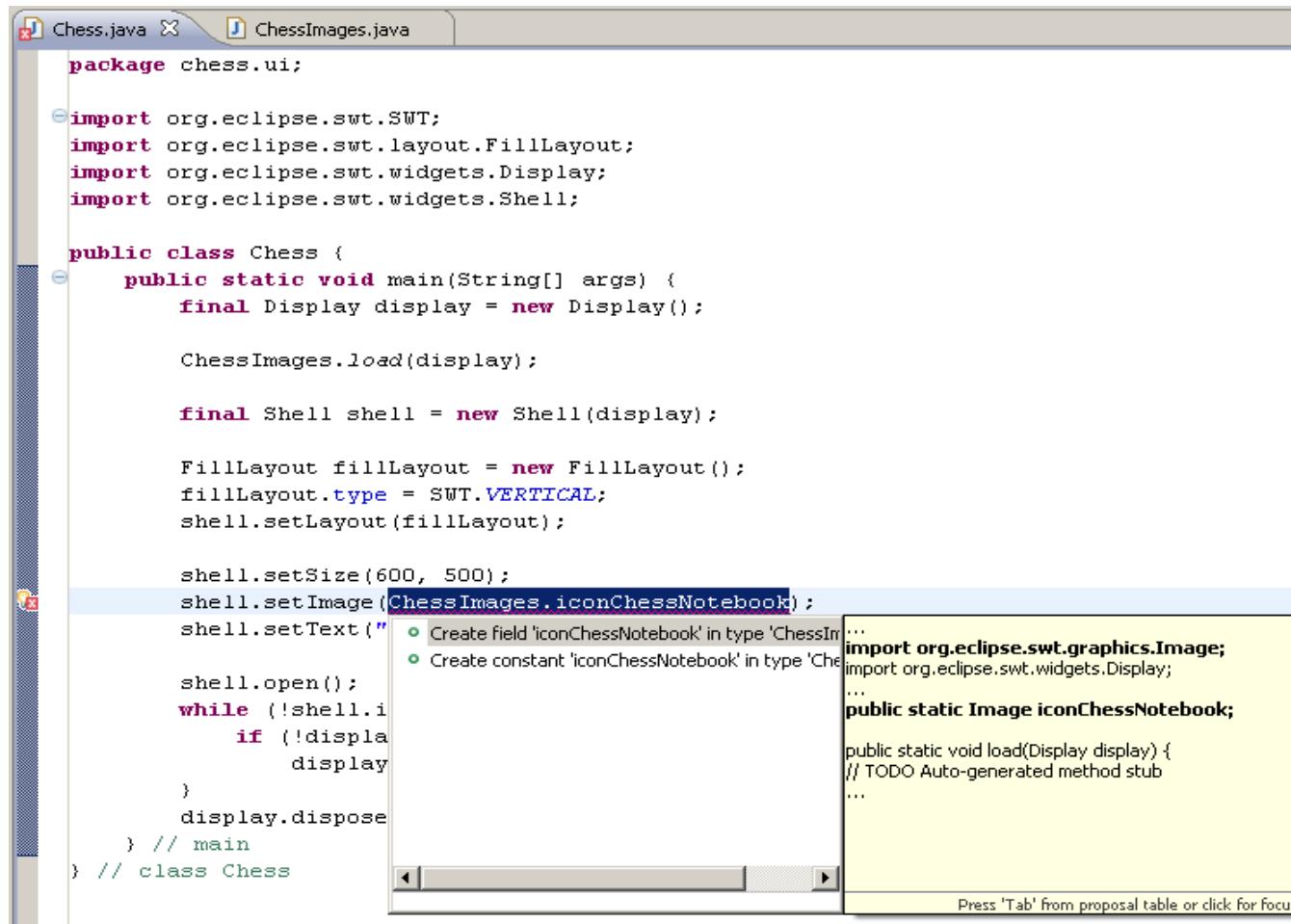
At the bottom right of the tooltip, there is a note: "Press 'Tab' from proposal table or click for focus".

```
package chess.ui;  
  
import org.eclipse.swt.SWT;  
import org.eclipse.swt.layout.FillLayout;  
import org.eclipse.swt.widgets.Display;  
import org.eclipse.swt.widgets.Shell;  
  
public class Chess {  
    public static void main(String[] args) {  
        final Display display = new Display();  
  
        ChessImages.load(display);  
        final Shell shell = new Shell(display);  
        FillLayout fillLayout = new FillLayout();  
        fillLayout.setLayout(shell);  
        shell.setLayout(fillLayout);  
        shell.setSize(400, 300);  
        shell.setImage(null);  
        shell.setText("Chess");  
        shell.open();  
        while (!shell.isDisposed()) {  
            if (!display.readAndDispatch())  
                display.sleep();  
        }  
        display.dispose();  
    } // main  
} // class Chess
```

# Создание поля *iconChessNotebook* в классе *ChessImages*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



```
Chess.java X  ChessImages.java
package chess.ui;

import org.eclipse.swt.SWT;
import org.eclipse.swt.layout.FillLayout;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Shell;

public class Chess {
    public static void main(String[] args) {
        final Display display = new Display();

        ChessImages.load(display);

        final Shell shell = new Shell(display);

        FillLayout fillLayout = new FillLayout();
        fillLayout.type = SWT.VERTICAL;
        shell.setLayout(fillLayout);

        shell.setSize(600, 500);
        shell.setImage(ChessImages.iconChessNotebook);
        shell.setText("Chess");
        shell.open();
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        display.dispose();
    } // main
} // class Chess
```

The screenshot shows an IDE interface with two tabs: "Chess.java" and "ChessImages.java". The "Chess.java" tab is active, displaying Java code for a "Chess" class. The "ChessImages.java" tab is visible in the background. A code completion tooltip is displayed over the line "shell.setImage(ChessImages.iconChessNotebook);". The tooltip contains two suggestions:

- Create field 'iconChessNotebook' in type 'ChessImages'
- Create constant 'iconChessNotebook' in type 'ChessImages'

Below the suggestions, the beginning of the "ChessImages.java" code is shown:

```
import org.eclipse.swt.graphics.Image;
import org.eclipse.swt.widgets.Display;
...
public static Image iconChessNotebook;
```

At the bottom of the tooltip, there is a note: "Press 'Tab' from proposal table or click for focus".

# Сгенерированный класс *ChessImages*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
package chess.ui;

import org.eclipse.swt.graphics.Image;
import org.eclipse.swt.widgets.Display;

public class ChessImages {

    public static Image iconChessNotebook;

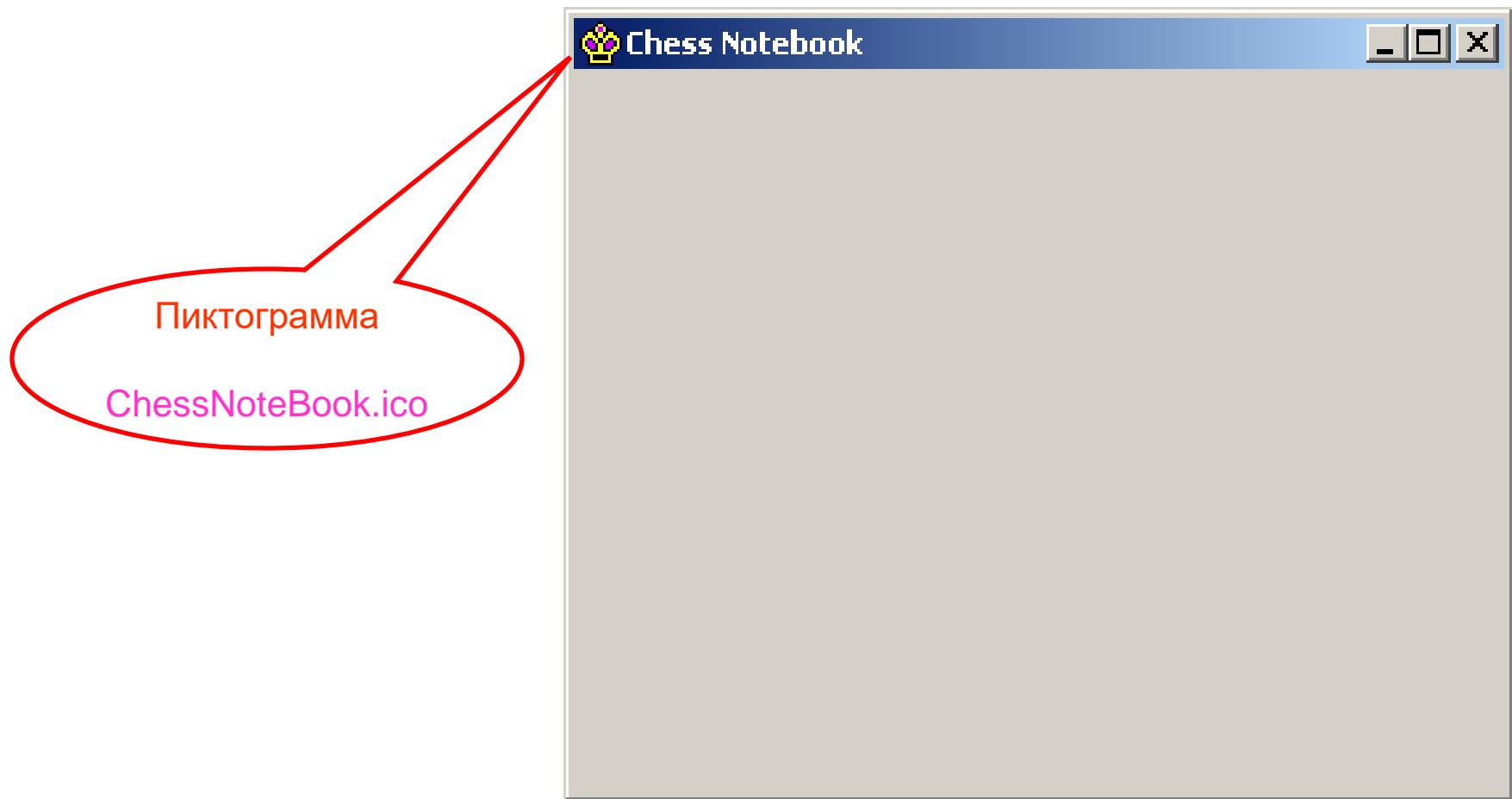
    public static void load(Display display) {
        iconChessNotebook = new Image(display,
            ChessImages.class.getResourceAsStream("ChessNoteBook.ico"));
    }
}
```

- **Файл *ChessNoteBook.ico* должен находиться в той же папке, что и файл *ChessImages.class***

# Шахматы. Блокнот с пиктограммой

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# SWT. Пакеты библиотеки

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

## ■ Пакеты библиотеки SWT на языке Java



- ❑ org.eclipse.swt
- ❑ org.eclipse.swt.widgets
- ❑ org.eclipse.swt.graphics
- ❑ org.eclipse.swt.events
- ❑ org.eclipse.swt.layout
- ❑ org.eclipse.swt.dnd
- ❑ org.eclipse.swt.printing
- ❑ org.eclipse.swt.program
- ❑ org.eclipse.swt.accessibility
- ❑ org.eclipse.swt.custom
- ❑ org.eclipse.swt.browser
- ❑ org.eclipse.swt.awt
- ❑ org.eclipse.swt.internal

# Класс SWT.

МГУ им. М.В.Ломоносова. Факультет ВМК.

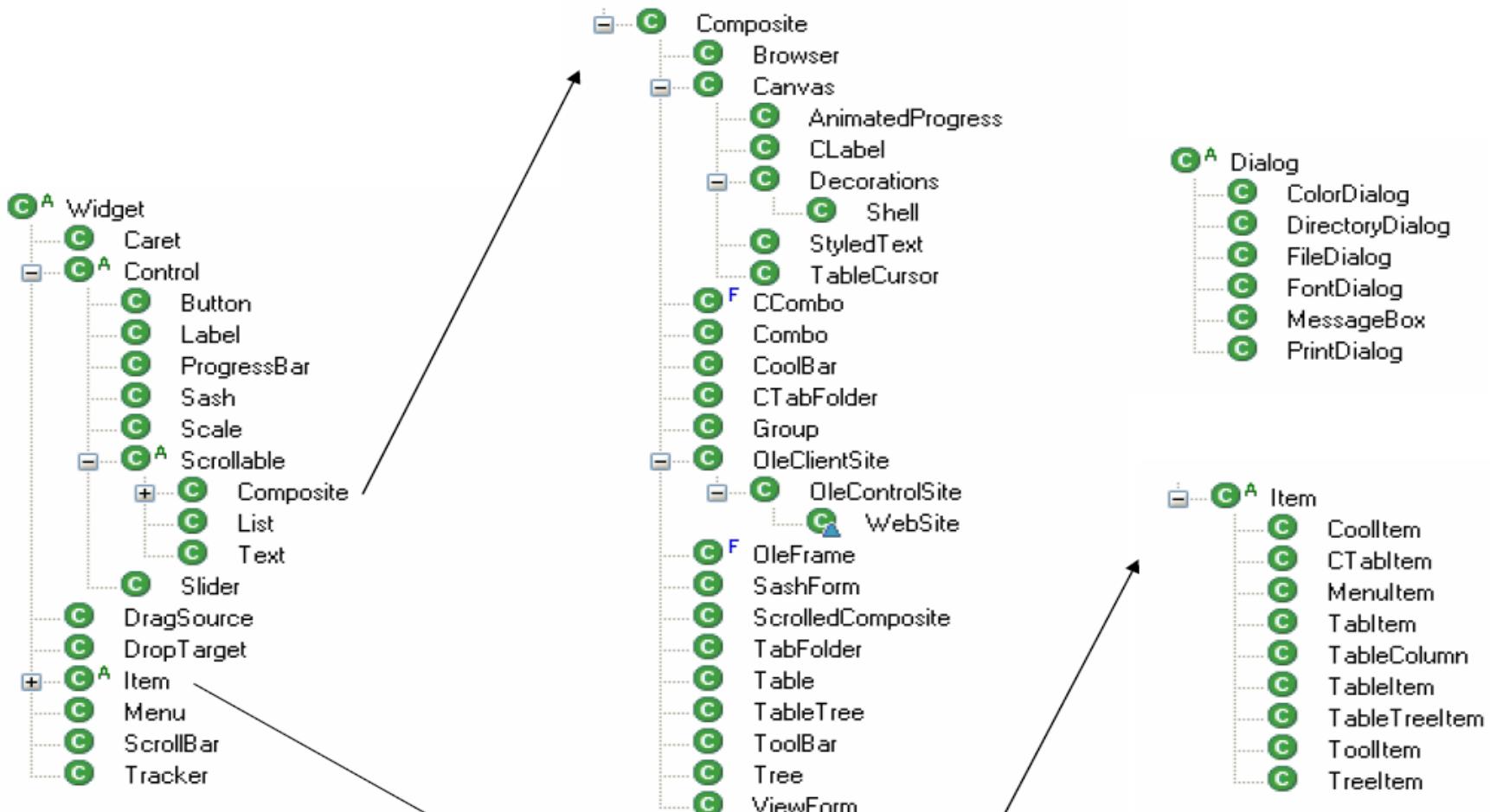
Романов Владимир Юрьевич ©2025

- Содержит все константы библиотеки
  - **SWT.PUSH, SWT.RADIO**
  - **SWT.Selection**
- Содержит универсальные методы
  - **getPlatform()**
  - **getVersion()**
  - **error()**

# Иерархия управляющих элементов (widgets).

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Конструкторы и стили управляющих элементов.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Управляющий элемент (*widget*) всегда имеет предка
- Вид типичного конструктора:  
*Widget(Composite parent, int style)*
- Стили задаются с помощью констант из класса *SWT*
- Примеры:
  - **new Label(shell, SWT.NONE);**
  - Button push = **new Button(shell, SWT.PUSH);**
  - Button radio = **new Button(parent, SWT.RADIO);**
  - Text text = **new Text(group, SWT.SINGLE | SWT.BORDER);**
- Исключение. Класс *Shell* всегда имеет предка *Shell* или *Display*
  - Shell shell = **new Shell(display, SWT.SHELL\_TRIM);**
  - Shell dialog = **new Shell(shell, SWT.DIALOG\_TRIM);**

# Класс Widget.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Абстрактный суперкласс для всех элементов интерфейса пользователя
- Создается с помощью конструкторов (без фабрик)
- При создании занимает ресурсы операционной системы
- Ресурсы освобождаются программно с помощью метода `dispose()`
- Уведомляет слушателей когда происходят с этим управляющим элементом происходят события
- Позволяет хранить специфичные для приложения данные
  - `setData(Object)`
  - `setData(String, Object)`
- Событие
  - `Dispose`

# Освобождение ресурсов графики и управляющих элементов.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Вы должны явно освобождать объекты потребляющие ресурсы:
  - Класс **Widget** и его подклассов
  - Классы **Color**, **Cursor**, **Font**, **GC**, **Image**, **Region**,
  - Класс **Device** и его подклассы (**Display**, **Printer**)
- **Правило 1: “Если вы создали его, вы его освобождаете”**
  - Программист должен освободить шрифт:

```
Font font = new Font (display, "Courier", 10, SWT.NORMAL);
font.dispose ();
```

- Программист не должен освободить шрифт:

```
Font font = control.getFont();
```

# Освобождение ресурсов управляющих элементов их предками.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- **Правило 2: “Освобождение предка освобождает его потомков “**
  - `shell.dispose();` // Освобождает всех потомков окна
  - `menu.dispose();` // Освобождает все элементы меню
  - `tree.dispose();` // Освобождает все элементы дерева
- Заметим, что:
  - `control.dispose();`
  - `menultem.dispose();`
  - Освобождает элемент меню созданный с помощью `setMenu(menu);`

# Класс Control

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Абстрактный суперкласс для всех легковесных (heavyweight) элементов интерфейса пользователя
- Стили
  - BORDER, LEFT\_TO\_RIGHT, RIGHT\_TO\_LEFT
- События
  - FocusIn, FocusOut
  - KeyDown, KeyUp
  - Traverse
  - MouseDown, MouseUp, MouseDoubleClick
  - MouseEnter, MouseExit, MouseMove, MouseHover
  - Move, Resize
  - Paint
  - Help

# Класс Shell

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

## ■ Конструкторы

- `new Shell(display, SWT.SHELL_TRIM);`
- `new Shell(shell, SWT.DIALOG_TRIM);`

## ■ Стили

- `BORDER, CLOSE, MIN, MAX,`  
`NO_TRIM, RESIZE, TITLE`
- `APPLICATION_MODAL, MODELESS,`  
`PRIMARY_MODAL, SYSTEM_MODAL`

## ■ События

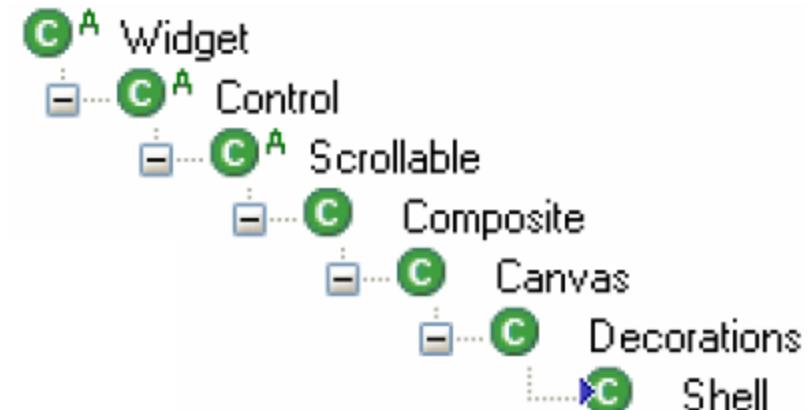
- `Close, Activate, Deactivate, Iconify, Deiconify`

## ■ Характерные методы

- `open(), close(), setActive()`

## ■ Замечания

- Предок для `shell` верхнего уровня всегда `Display`
- Предок для `shell` - диалогового окна всегда `Shell` верхнего уровня

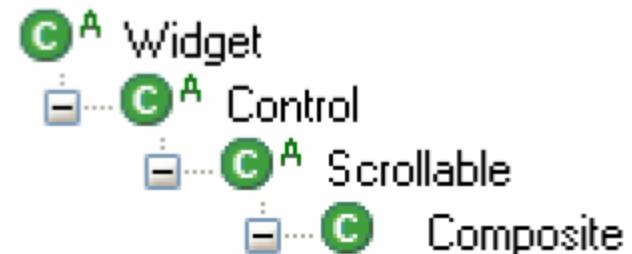


# Класс Composite

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Конструкторы
  - `new Composite(parent, SWT.NONE);`
- Стили
  - `NO_BACKGROUND, NO_FOCUS,`
  - `NO_MERGE_PAINTS, NO_REDRAW_RESIZE,`
  - `NO_RADIO_GROUP`
- События
  - `Close, Activate, Deactivate, Iconify, Deiconify`
- Характерные методы
  - `getChildren()`
  - `setLayout(Layout), layout(boolean)`
  - `setTabList(Control[])`
- Замечания
  - может иметь потомков – управляющие элементы (*controls*)
  - может использовать класс **Layout** для задания положения потомков
  - используется как суперкласс для создания управляющих элементов (*widgets*) пользователя

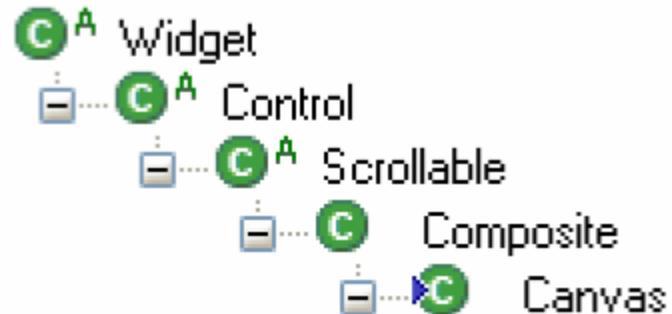


# Класс Canvas

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Конструкторы
  - `new Canvas(parent, SWT.NONE);`
- Стили
  - `NO_BACKGROUND, NO_FOCUS,`
  - `NO_MERGE_PAINTS, NO_REDRAW_RESIZE`
- Характерные методы
  - `scroll(int, int, int, int, int, boolean)`
  - `setCaret(Caret)`
- Замечания
  - Обычно используется как «чистый лист бумаги» для рисования графики
  - используется как суперкласс для создания управляющих элементов (*widgets*) пользователя

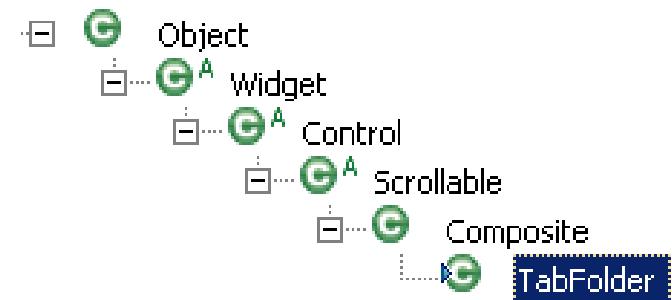


# Класс TabFolder

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Конструкторы
  - `new TabFolder(parent, SWT.TOP);`
- Стили
  - `SWT.TOP , SWT.BOTTOM, SWT.LEFT, SWT.RIGHT`
- Характерные методы
  - `setSelection(int, boolean)`
  - `setSelection(TabItem)`
  - `TabItem getSelection()`
  - `int getSelectionIndex()`
- Замечания
  - используется как «книга» с закладками

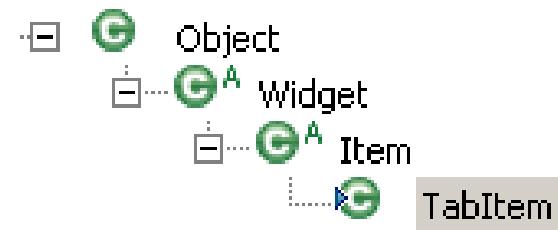


# Класс TabItem

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Конструкторы
  - `new TabItem(parent, SWT.NONE);`
- Стили
  - Унаследованные от родителей
- Характерные методы
  - `setControl(Control)`
  - `setImage(Image)`
  - `setText(String)`
  - `setToolTipText(String)`
- Замечания
  - используется как закладка в «книге»



# Шахматы. Шахматная доска – теперь как закладка в «книге» игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public static void main(String[] args) {  
    final Display display = new Display();  
    ChessImages.load(display);  
  
    final Shell shell = new Shell(display);  
    shell.setSize(600, 500);  
    shell.setText("Chess Notebook");  
    shell.setImage(ChessImages.iconChessNotebook);  
    setLayout(new FillLayout()); // Чтобы растянуть gamesFolder на все окно.  
  
    final TabFolder gamesFolder = new TabFolder(shell, SWT.TOP);  
    TabItem chessItem = new TabItem(gamesFolder, SWT.NULL);  
    chessItem.setText("Шахматы");  
    chessItem.setControl( new ChessBoard(gamesFolder, SWT.NONE) );  
  
    // ...  
} // main
```

Вставка шахматной  
доски в закладку

# Шахматы. Шахматы – пиктограмма для закладки в блокноте игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public static void main(String[] args) {  
    // ...  
  
    final TabFolder gamesFolder = new TabFolder(shell, SWT.TOP);  
  
    final Image chessTabImage = new Image(display,  
        ChessImages.imageKnightBlack.getImageData().scaledTo(20, 20) );  
  
    TabItem chessItem = new TabItem(gamesFolder, SWT.NULL);  
    chessItem.setText("Шахматы");  
    chessItem.setImage(chessTabImage);  
    chessItem.setControl( new ChessBoard(gamesFolder, SWT.NONE) );  
  
    // ...  
} // main
```

Масштабирование  
рисунка

Пиктограмма для  
закладки шахмат в  
блокноте клеточных игр

# Шахматы. Создание класса для шахматной доски с помощью wizard

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

**Ctrl + 1** когда «каретка» в тексте *ChessBoard*

The screenshot shows the Eclipse IDE interface with a Java code editor window titled "Chess.java". The code is as follows:

```
28
29     FillLayout layout = new FillLayout();
30     shell.setLayout(layout);
31
32     TabFolder gamesFolder = new TabFolder(shell, SWT.TOP);
33
34     Image chessImage = new Image(display, ChessImages.imageKnightBlack
35         .getImageData().scaledTo(20, 20));
36
37     TabItem chessItem = new TabItem(gamesFolder, SWT.NONE);
38     chessItem.setText("Шахматы");
39     chessItem.setControl(new ChessBoard(gamesFolder, SWT.NONE));
```

At line 39, the cursor is positioned after "ChessBoard" in the "setControl" method call. A context menu is open, listing several options:

- Export the 'chess.ui' package from the 'GameNotebook' plug
- Create class 'ChessBoard'** (highlighted)
- Change to 'Control' (org.eclipse.swt.widgets)
- Change to 'Chess' (chess.ui)
- Change to 'Clipboard' (java.awt.datatransfer)
- Change to 'Clipboard' (org.eclipse.swt.dnd)
- Rename in file (Ctrl+2, R)
- Fix project setup...

At the bottom of the code editor, there is a message: "Press 'Tab' from proposal table or click for focus".

# Шахматы. Создание класса для шахматной доски с помощью wizard

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Всплывающая подсказка когда мышка над текстом *ChessBoard*

```
28
29     FillLayout layout = new FillLayout();
30     shell.setLayout(layout);
31
32     TabFolder gamesFolder = new TabFolder(shell, SWT.TOP);
33
34     Image chessImage = new Image(display, ChessImages.imageKnightBlack
35         .getImageData().scaledTo(20, 20));
36
37     TabItem chessItem = new TabItem(gamesFolder, SWT.NONE);
38     chessItem.setText("Шахматы");
39     chessItem.setControl(new ChessBoard(gamesFolder, SWT.NONE));
40     chessItem.setImage(chessImage);
41
42     shell.open();
43     while (!shell.isDisposed())
44         if (!display.readAndDispatch())
45             display.sleep();
46     display.dispose();
47
48 }
49 }
```

ChessBoard cannot be resolved to a type

7 quick fixes available:

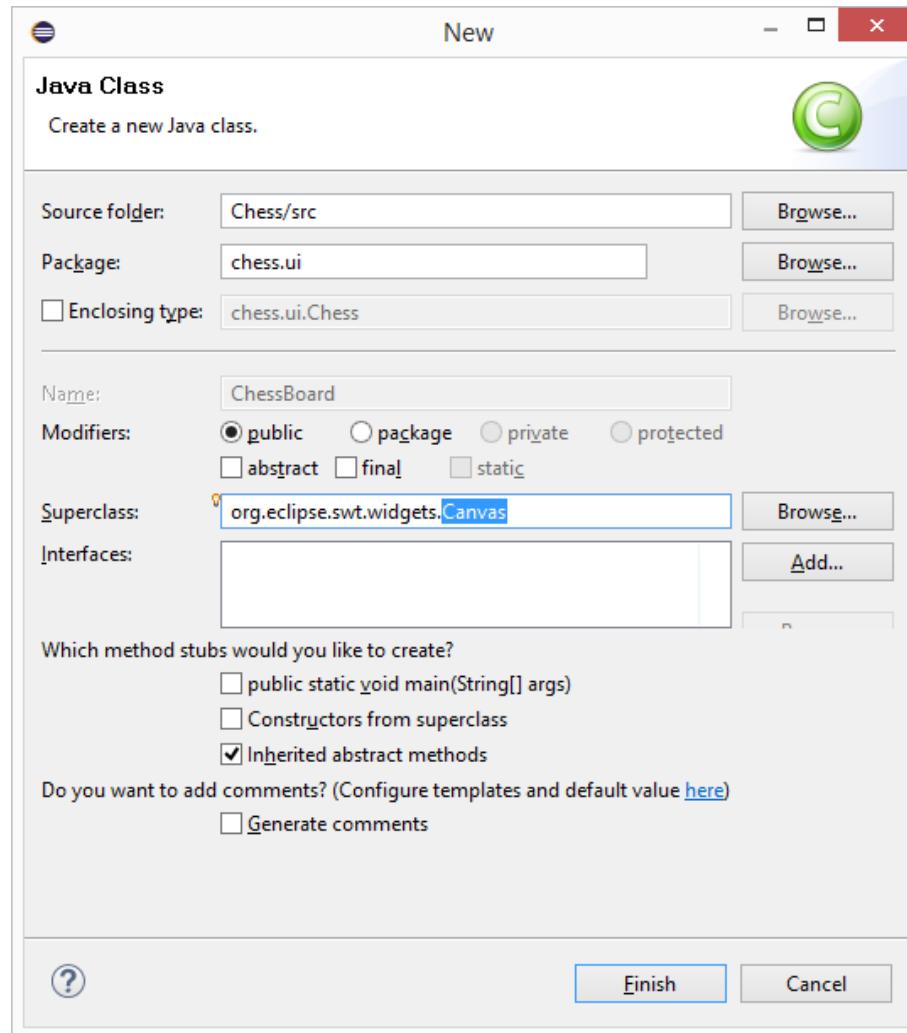
- Export the 'chess.ui' package from the 'GameNotebook' plug-in
- Create class 'ChessBoard'
- Change to 'Control' (org.eclipse.swt.widgets)
- Change to 'Chess' (chess.ui)
- Change to 'Clipboard' (java.awt.datatransfer)
- Change to 'Clipboard' (org.eclipse.swt.dnd)
- Fix project setup...

Press 'F2' for focus

# Шахматы. Создание класса для шахматной доски с помощью wizard

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Шахматы. Класс *ChessBoard* создан

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Java - Chess/src/chess/ui/ChessBoard.java - E
- Menu Bar:** File Edit Source Refactor Navigate Search Project CocoR Run Window Help
- Toolbar:** Includes icons for file operations, search, and run.
- Package Explorer:** Shows the project structure:
  - Chess [chess2017 games2018\_10\_19]
  - JRE System Library [JavaSE-1.8]
  - Plug-in Dependencies
  - src
    - chess
      - Board.java
      - Chess.java
      - ChessBoard.java
    - chess2014
    - chess2014.handlers
- Editor Area:** Displays the code for **ChessBoard.java**:

```
1 package chess.ui;
2
3 import org.eclipse.swt.widgets.Canvas;
4
5 public class ChessBoard extends Canvas {
6
7 }
```

# Шахматы. Создание конструктора класса *ChessBoard* с помощью wizard

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with two tabs open: 'Chess.java' and 'ChessBoard.java'. The 'ChessBoard.java' tab is active, displaying the following Java code:

```
1 package chess.ui;
2
3 import org.eclipse.swt.widgets.Canvas;
4
5 public class ChessBoard extends Canvas {
6
7 }
```

A code completion dropdown menu is open at the cursor position (line 5, character 15) over the word 'ChessBoard'. The menu contains the following options:

- Add constructor 'ChessBoard()'
- Add constructor 'ChessBoard(Composite,int)' (highlighted)
- Rename in file (Ctrl+2, R)
- Rename in workspace (Alt+Shift+R)

To the right of the code editor, a preview window shows the generated code for the selected constructor:

```
...
public class ChessBoard extends Canvas {
    ChessBoard() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```

At the bottom of the preview window, there is a message: "Press 'Tab' from proposal table or click for focus".

**Ctrl + 1** когда «каретка» в тексте *ChessBoard*

# Шахматы. Конструктор класса *ChessBoard*

## создан

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with the following details:

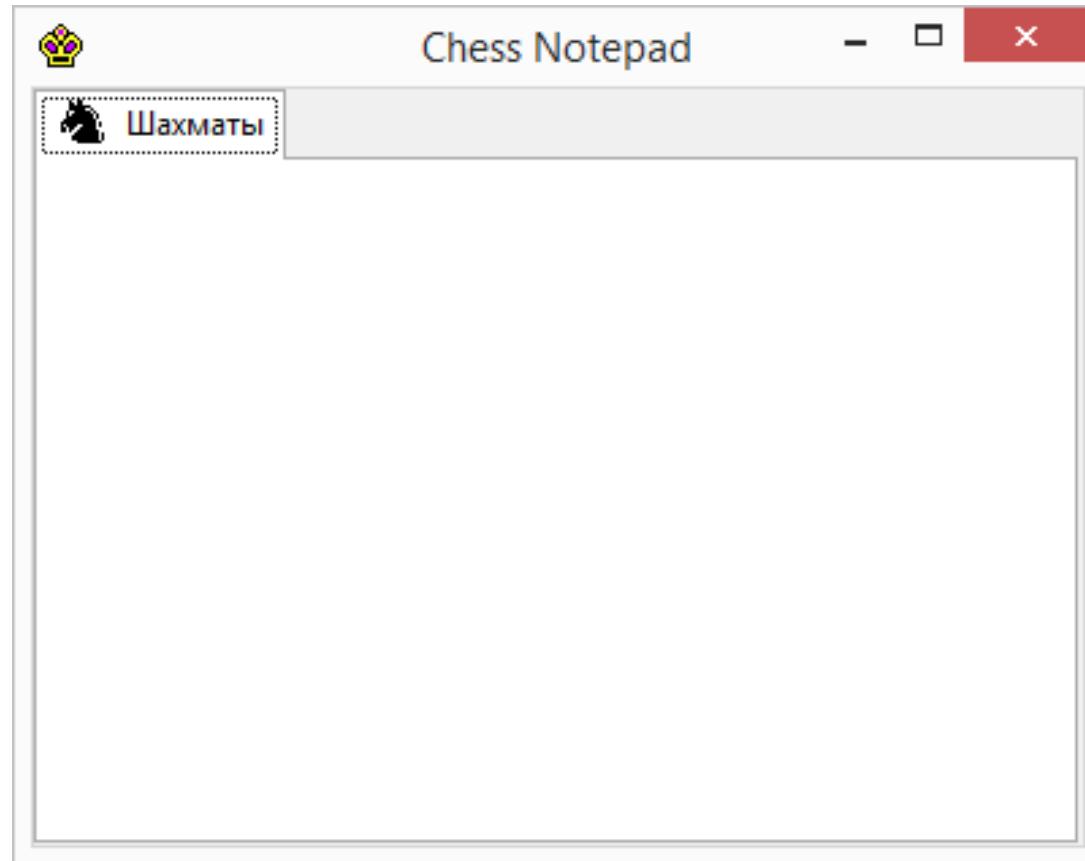
- Project Explorer View:** Shows the project structure under the "Chess" project.
  - Java Build Path:
    - JRE System Library [JavaSE-1.8]
    - Plug-in Dependencies
  - src:
    - chess
      - Board.java
      - Chess.java
      - ChessBoard.java (selected)
    - chess2014
    - chess2014.handlers
    - readme.txt
  - icons
- Code Editor View:** Displays the content of the `ChessBoard.java` file.

```
1 package chess.ui;
2
3 import org.eclipse.swt.widgets.Canvas;
4 import org.eclipse.swt.widgets.Composite;
5
6 public class ChessBoard extends Canvas {
7
8     public ChessBoard(Composite parent, int style) {
9         super(parent, style);
10        // TODO Auto-generated constructor stub
11    }
12
13 }
14
```

# Шахматы. Показывается только пустое поле для рисования доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Шахматы. Рисование рамки доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

The screenshot shows the Eclipse IDE interface with the title "Java - Chess/src/chess/ui/ChessBoard.java - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, CocoR, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, and Cut. The left sidebar is the Package Explorer showing the project structure:

- Chess [chess2017 games2018\_10\_19]
  - JRE System Library [JavaSE-1.8]
  - Plug-in Dependencies
- src
  - chess
    - Board.java
    - Chess.java
    - ChessBoard.java
  - chess2014
  - chess2014 handlers

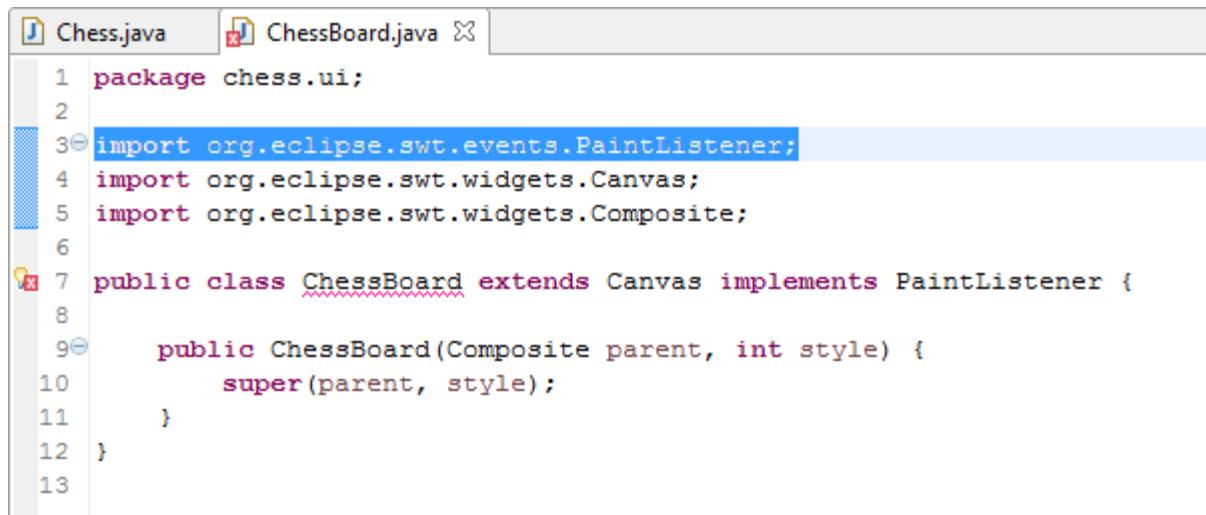
The right side shows two tabs: "Chess.java" and "ChessBoard.java". The "ChessBoard.java" tab contains the following Java code:

```
1 package chess.ui;
2
3 import org.eclipse.swt.widgets.Canvas;
4 import org.eclipse.swt.widgets.Composite;
5
6 public class ChessBoard extends Canvas implements PaintListener {
7
8     public ChessBoard(Composite parent, int style) {
9         super(parent, style);
10    }
11 }
```

# Шахматы. Вставка импорта неизвестного класса или интерфейса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



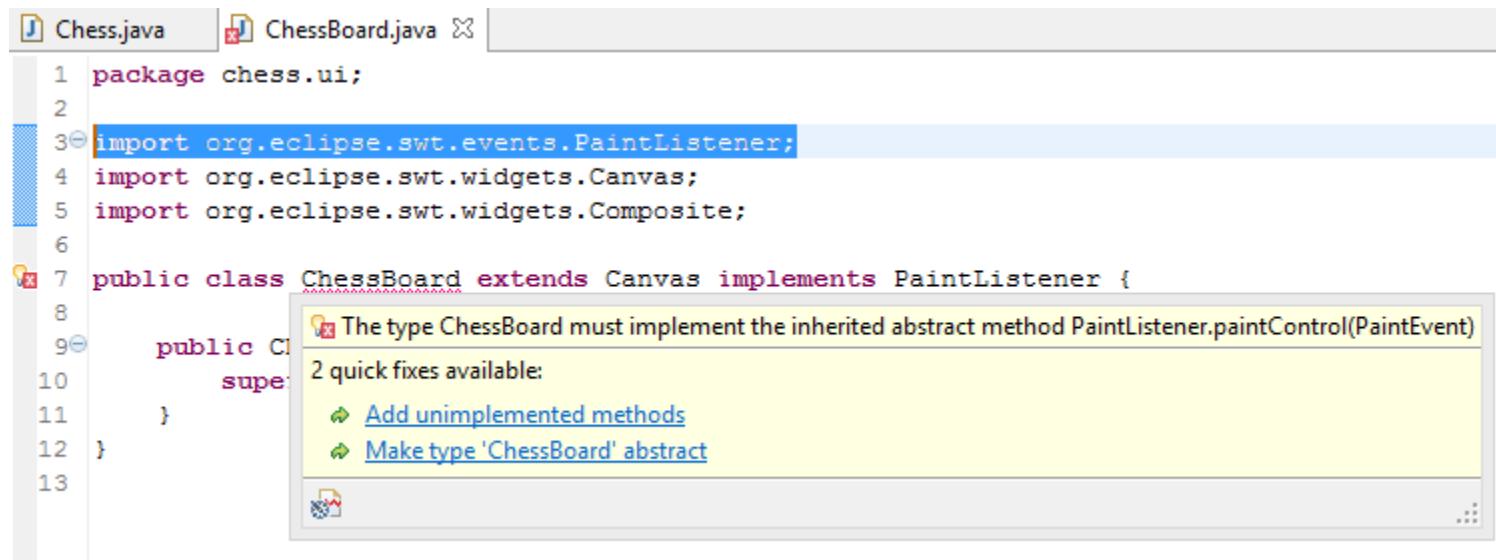
```
Chess.java  ChessBoard.java X
1 package chess.ui;
2
3 import org.eclipse.swt.events.PaintListener;
4 import org.eclipse.swt.widgets.Canvas;
5 import org.eclipse.swt.widgets.Composite;
6
7 public class ChessBoard extends Canvas implements PaintListener {
8
9     public ChessBoard(Composite parent, int style) {
10         super(parent, style);
11     }
12 }
13
```

**Ctrl + Shift + O** для вставки импорта с помощью wizard

# Шахматы. Класс *ChessBoard* реализует интерфейс *PaintListener*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



The screenshot shows the Eclipse IDE interface with two tabs: 'Chess.java' and 'ChessBoard.java'. The 'ChessBoard.java' tab is active, displaying the following code:

```
1 package chess.ui;
2
3 import org.eclipse.swt.events.PaintListener;
4 import org.eclipse.swt.widgets.Canvas;
5 import org.eclipse.swt.widgets.Composite;
6
7 public class ChessBoard extends Canvas implements PaintListener {
8
9     public ChessBoard(Composite parent) {
10         super(parent);
11     }
12 }
```

A code completion tooltip is open over the 'implements' keyword in line 7, showing the following message:

The type ChessBoard must implement the inherited abstract method PaintListener.paintControl(PaintEvent)  
2 quick fixes available:  
Add unimplemented methods  
Make type 'ChessBoard' abstract

**Ctrl + Shift + O** добавление нереализованных методов с помощью wizard

# Шахматы. Добавлена реализация методов интерфейса *PaintListener*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
Chess.java ChessBoard.java X

1 package chess.ui;
2
3 import org.eclipse.swt.events.PaintEvent;
4 import org.eclipse.swt.events.PaintListener;
5 import org.eclipse.swt.widgets.Canvas;
6 import org.eclipse.swt.widgets.Composite;
7
8 public class ChessBoard extends Canvas implements PaintListener {
9
10    public ChessBoard(Composite parent, int style) {
11        super(parent, style);
12    }
13
14    @Override
15    public void paintControl(PaintEvent e) {
16        // TODO Auto-generated method stub
17
18    }
19
20 }
```

# Шахматы. Рисование рамки по границам

## ДОСКИ

МГУ им. М.В.Ломоносова. Факультет ВМК.

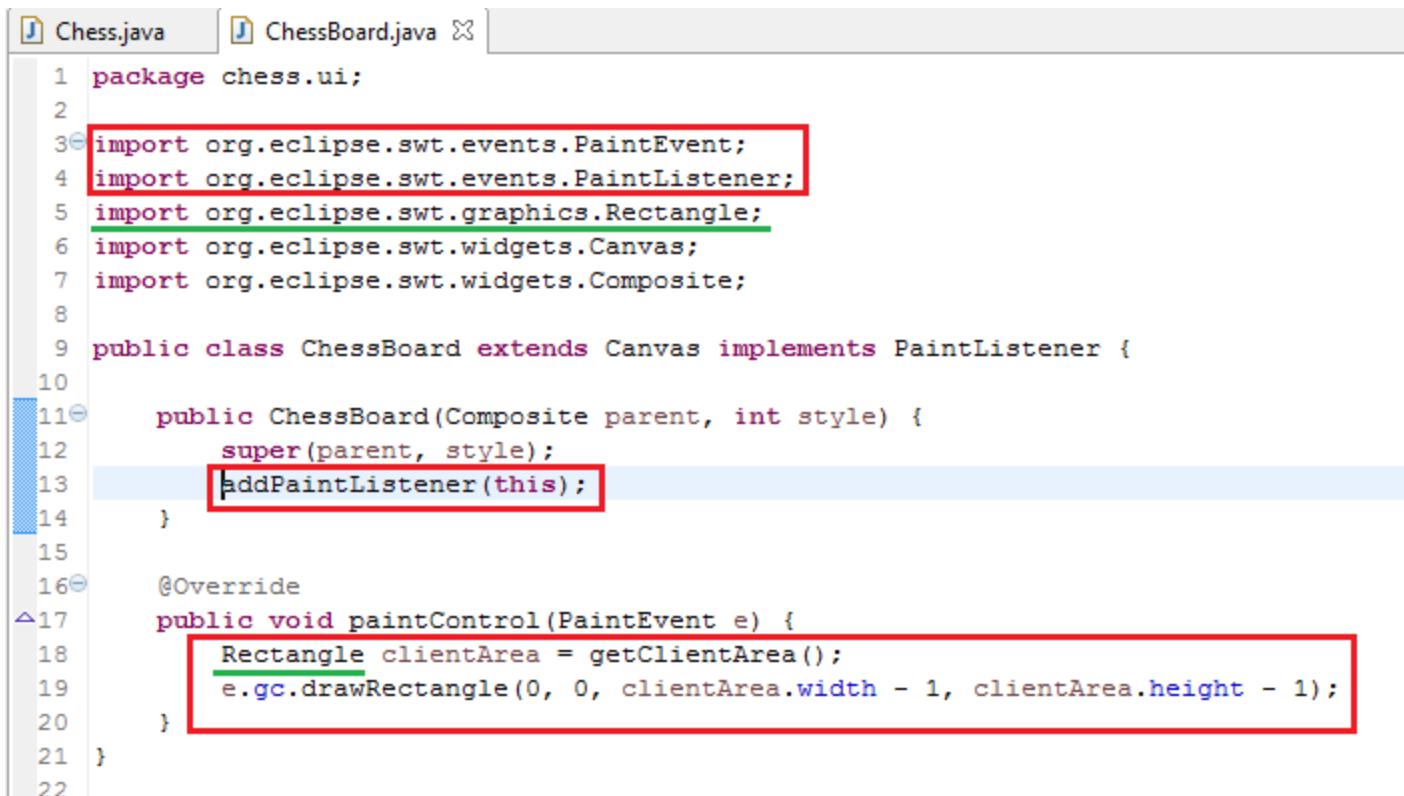
Романов Владимир Юрьевич ©2025

```
public void paintControl(PaintEvent e) {  
    Rectangle clientArea = getClientArea();  
    e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);  
}
```

# Шахматы. Рисование рамки по границам доски слушателем события *Paint*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



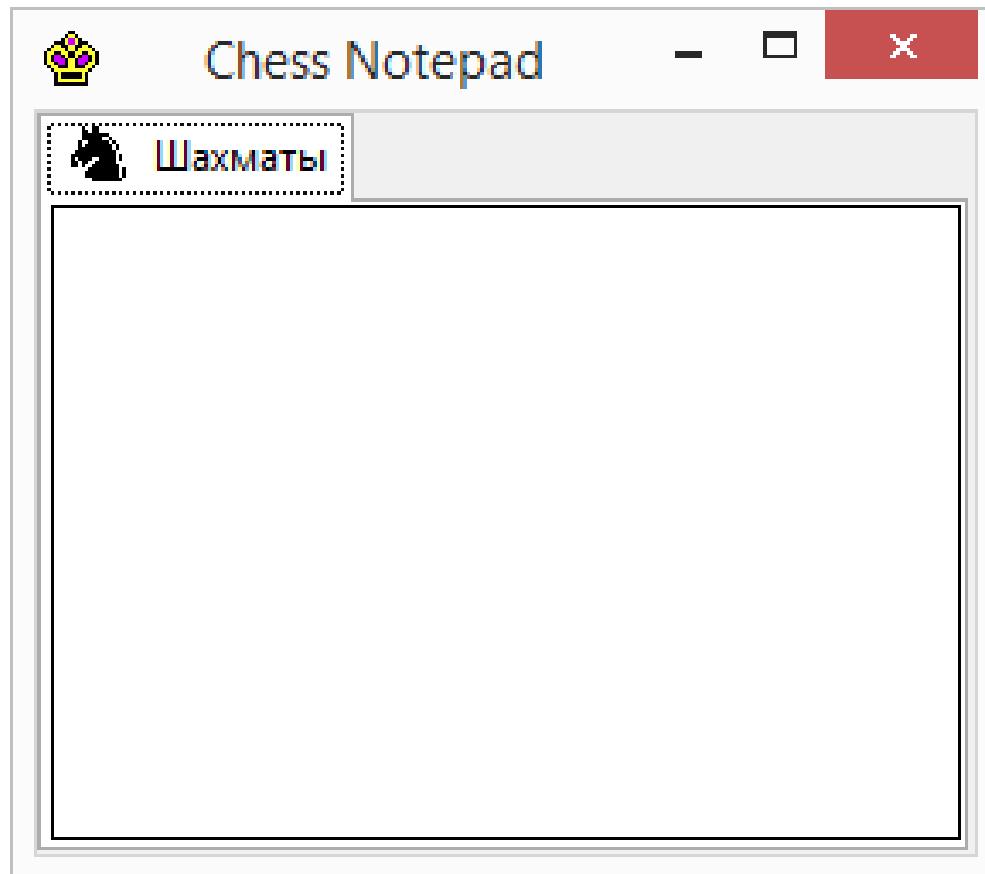
```
Chess.java  ChessBoard.java X

1 package chess.ui;
2
3 import org.eclipse.swt.events.PaintEvent;
4 import org.eclipse.swt.events.PaintListener;
5 import org.eclipse.swt.graphics.Rectangle;
6 import org.eclipse.swt.widgets.Canvas;
7 import org.eclipse.swt.widgets.Composite;
8
9 public class ChessBoard extends Canvas implements PaintListener {
10
11     public ChessBoard(Composite parent, int style) {
12         super(parent, style);
13         addPaintListener(this);
14     }
15
16     @Override
17     public void paintControl(PaintEvent e) {
18         Rectangle clientArea = getClientArea();
19         e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);
20     }
21 }
22
```

# Шахматы. Рамка доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Шахматы. Отрисовка клеток доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public void paintControl(PaintEvent e) {  
    Rectangle clientArea = getClientArea();  
    GC gc = e.gc;  
    for (int v = 0; v < nV; v++)  
        for (int h = 0; h < nH; h++) {  
            int squareWidth = getClientArea().width / nV;  
            int squareHeight = getClientArea().height / nH;  
  
            boolean isWhiteSquare = ((v + h) % 2 == 0);  
            Color squareColor = isWhiteSquare ? new Color(null, 255, 255, 255) : new Color(null, 0, 192, 0);  
  
            gc.setBackground(squareColor);  
            gc.fillRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
  
            gc.setForeground(new Color(null, 0, 0, 0));  
            gc.drawRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
        } // for  
    e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);  
} // paintControl
```

# Шахматы. Создание поля (field) для необъявленных nV и nH

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
16    }
17
18    @Override
19    public void paintControl(PaintEvent e) {
20        Rectangle clientArea = getClientArea();
21        GC gc = e.gc;
22        for (int v = 0; v < nV; v++)
23            for (int h = 0; h < nH; h++) {
24                int squareWidth
25                int squareHeight
26
27                boolean isWhiteS
28                Color squareColo
29
30                gc.setBackground
31                gc.fillRectangle
32
33                gc.setForeground
34                gc.drawRectangle
35            } // for
36        e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);

```

nV;  
nH;

6 quick fixes available:

- ① [Create local variable 'nH'](#)
- ② [Create field 'nH'](#)
- ③ [Change to 'h'](#)
- ④ [Create parameter 'nH'](#)
- ⑤ [Create constant 'nH'](#)
- ⑥ [Change to 'v'](#)

r(null, 255, 255, 255) : new Color(null, 0, 192, 0

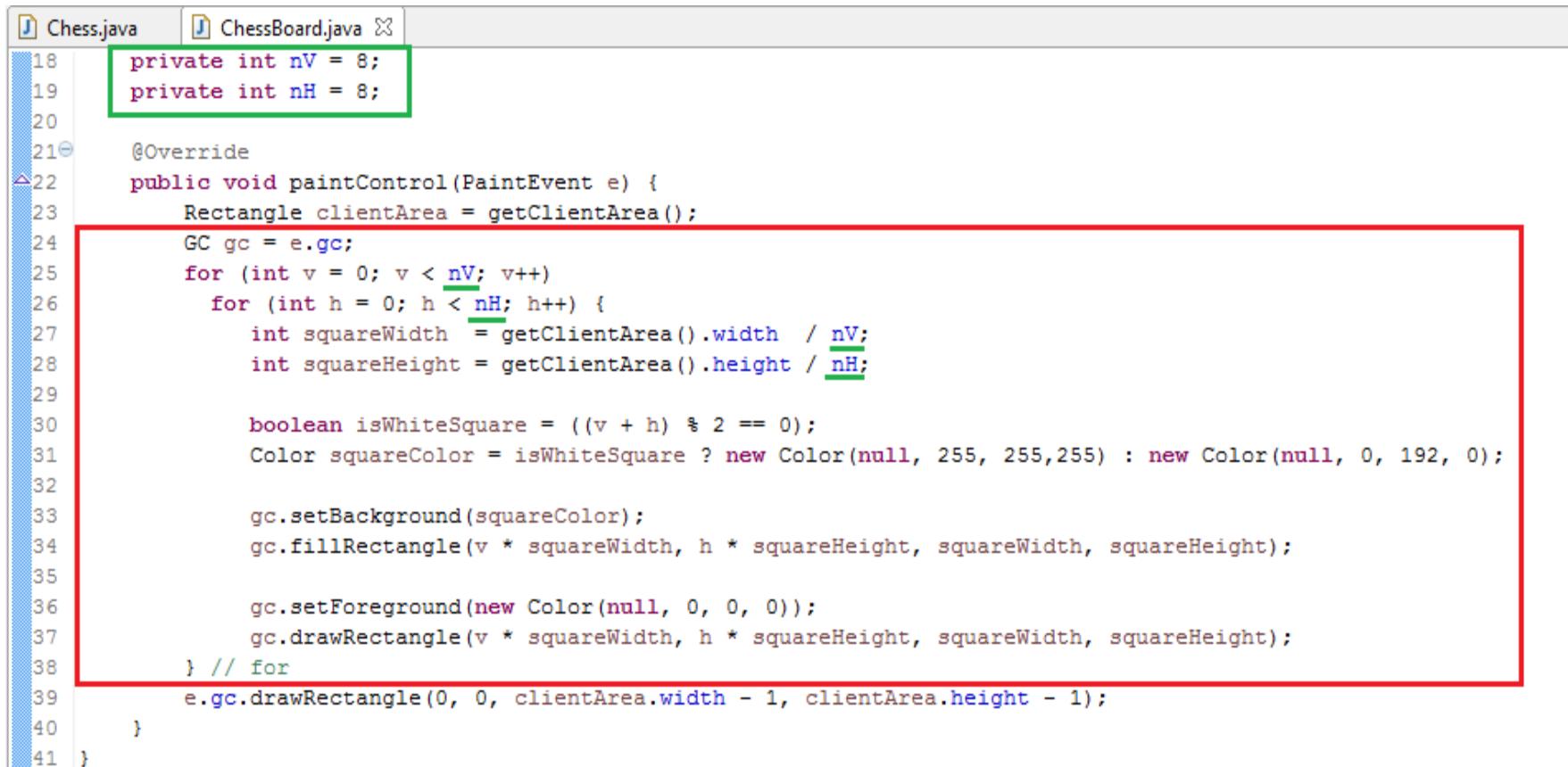
Height, squareWidth, squareHeight);

Height, squareWidth, squareHeight);

# Шахматы. Отрисовка клеток доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

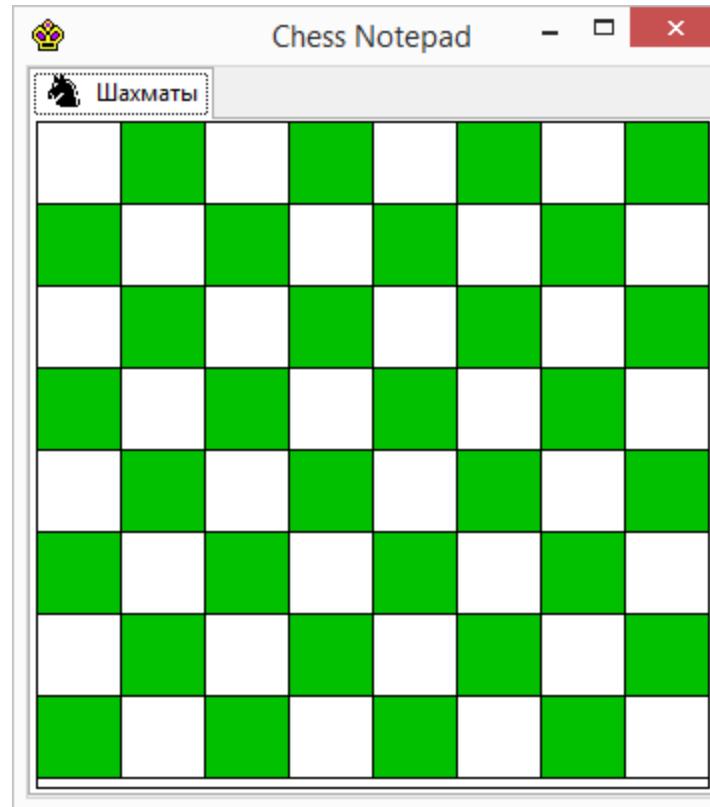


```
Chess.java  ChessBoard.java X
18  private int nV = 8;
19  private int nH = 8;
20
21  @Override
22  public void paintControl(PaintEvent e) {
23      Rectangle clientArea = getClientArea();
24      GC gc = e.gc;
25      for (int v = 0; v < nV; v++) {
26          for (int h = 0; h < nH; h++) {
27              int squareWidth = getClientArea().width / nV;
28              int squareHeight = getClientArea().height / nH;
29
30              boolean isWhiteSquare = ((v + h) % 2 == 0);
31              Color squareColor = isWhiteSquare ? new Color(null, 255, 255, 255) : new Color(null, 0, 192, 0);
32
33              gc.setBackground(squareColor);
34              gc.fillRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);
35
36              gc.setForeground(new Color(null, 0, 0, 0));
37              gc.drawRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);
38          } // for
39          e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);
40      }
41 }
```

# Шахматы. Клетки шахматной доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Рефакторинг программы. Разновидности досок для игр

# Рефакторинг. Вынос универсальных полей и методов в базовый класс

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

## ❖ Предпосылки

- Размерность доски - поля `nH` и `nV` общие для всех клеточных игр
- Алгоритм рисования в цикле всех клеток доски универсален

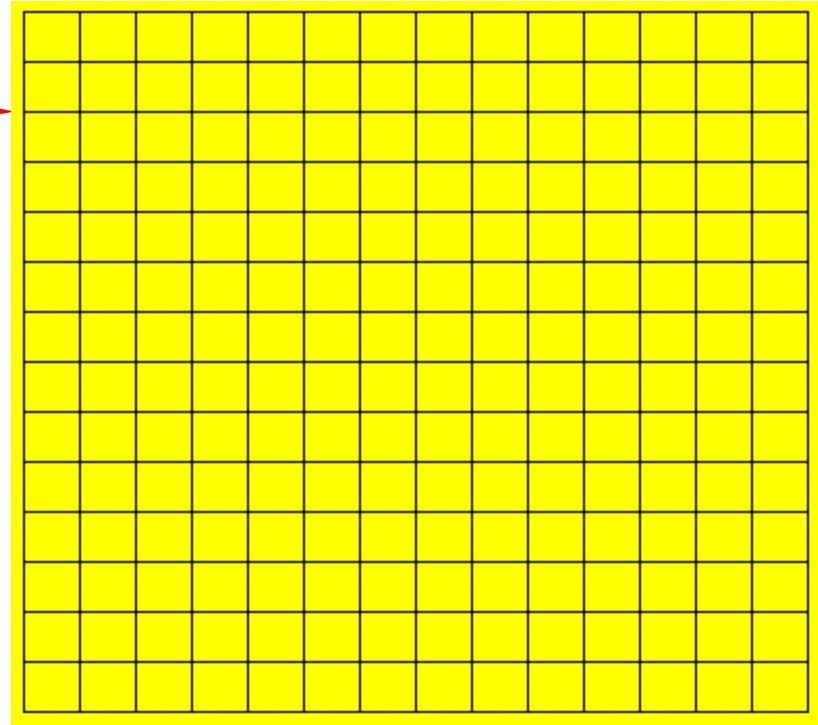
# Доска для азиатских игр.

## Клетки доски для игр го и ренджю

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Фигуры  
располагаются на  
пересечении линий



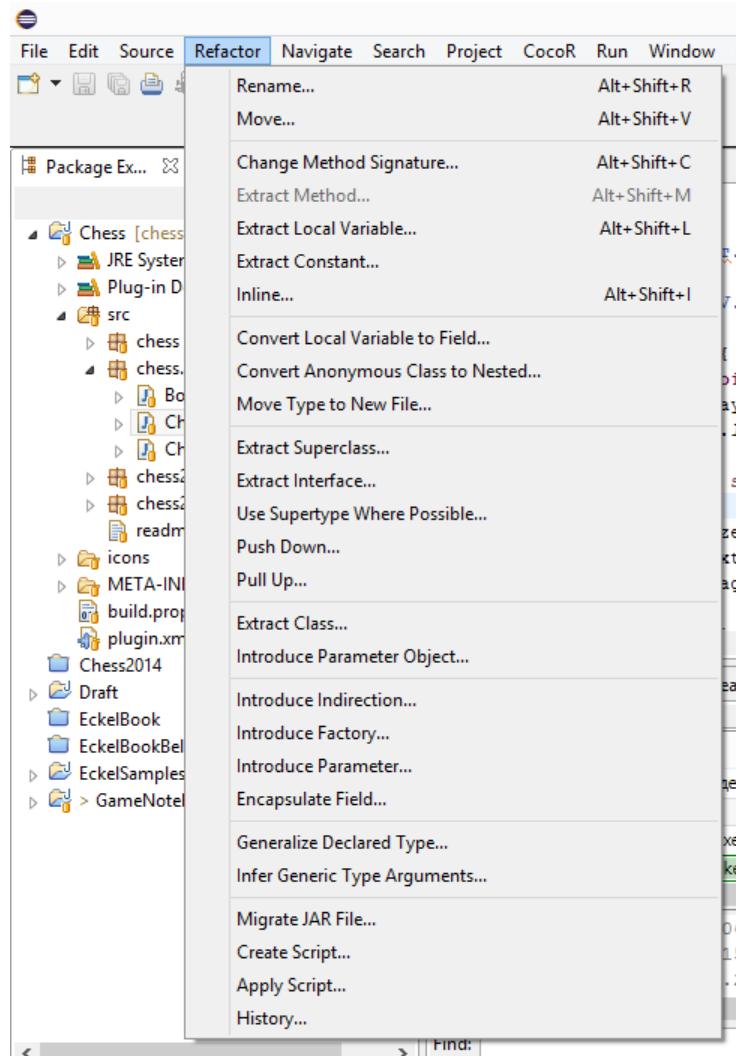
- Цель – преобразовать программу для рисования досок различных настольных игр

# Рефакторинг.

## Инструменты рефакторинга

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Контексто-зависимый рефакторинг.

## Клавиши Alt + Shift + T (Eclipse)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

### Преобразования применимые к классу ChessBoard

```
1 package chess.ui;
2
3+ import org.eclipse.swt.events.PaintEvent;
10
11 public class ChessBoard extends Canvas implements PaintListener{
12
13     public Ch
14         super
15             addPa
16         }
17
18     private i
19     private i
20
21     @Override
22     public vo
23         Recta
--
```

Rename... Alt+Shift+R  
Move... Alt+Shift+V  
Extract Interface...  
Extract Superclass...  
Use Supertype Where Possible...  
Pull Up...  
Push Down...  
Extract Class...  
Generalize Declared Type...

# Контексто-зависимый рефакторинг.

## Клавиши Alt + Shift + T

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

### Преобразования применимые к методу paintControl

```
ChessBoard.java
@Override
public void paintControl(PaintEvent e) {
    Rectangle1
    GC gc =
    for (int
        for (i
            int
            int
            boolean
            Col
            gc.
            gc.
            gc.
            gc.
            gc.
            gc.
    } // for
    e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea
}
```

Context menu for the `paintControl` method:

- Rename... Alt+Shift+R
- Move... Alt+Shift+V
- Change Method Signature... Alt+Shift+C
- Inline... Alt+Shift+I
- Extract Interface...
- Extract Superclass...
- Use Supertype Where Possible...
- Pull Up...
- Push Down...
- Extract Class...
- Introduce Parameter Object...
- Introduce Indirection...

Annotations in the code:

- `nV;` and `nH;` are placed after the first two `int` declarations.
- `or(null,` and `eHeight,` are placed after the `Col` declaration.
- `eHeight,` is placed after the `clientArea` declaration.

# Контексто-зависимый рефакторинг.

## Клавиши Alt + Shift + T

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

### Преобразования применимые к выделенному тексту

The screenshot shows a Java code editor window for a file named `ChessBoard.java`. The code implements a `paintControl` method. A context menu is open over the following line of code:

```
int squareWidth = getClientArea().width / nV;
```

The menu items visible in the context menu are:

- Move... (Alt+Shift+V)
- Change Method Signature... (Alt+Shift+C)
- Extract Method... (Alt+Shift+M)** (highlighted)
- Extract Interface...
- Extract Superclass...
- Use Supertype Where Possible...
- Pull Up...
- Push Down...
- Extract Class...
- Introduce Parameter Object...

# Шахматы. Отрисовка клеток доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public void paintControl(PaintEvent e) {  
    Rectangle clientArea = getClientArea();  
    GC gc = e.gc;  
    for (int v = 0; v < nV; v++)  
        for (int h = 0; h < nH; h++) {  
            int squareWidth = getClientArea().width / nV;  
            int squareHeight = getClientArea().height / nH;  
  
            boolean isWhiteSquare = ((v + h) % 2 == 0);  
            Color squareColor = isWhiteSquare ? new Color(null, 255, 255, 255) : new Color(null, 0, 192, 0);  
  
            gc.setBackground(squareColor);  
            gc.fillRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
  
            gc.setForeground(new Color(null, 0, 0, 0));  
            gc.drawRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
        } // for  
    e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);  
} // paintControl
```

Алгоритм рисования клетки шахматной доски вынесем в отдельный метод

# Рефакторинг. Выделение метода.

## Клавиши **ALT + Shift + M**

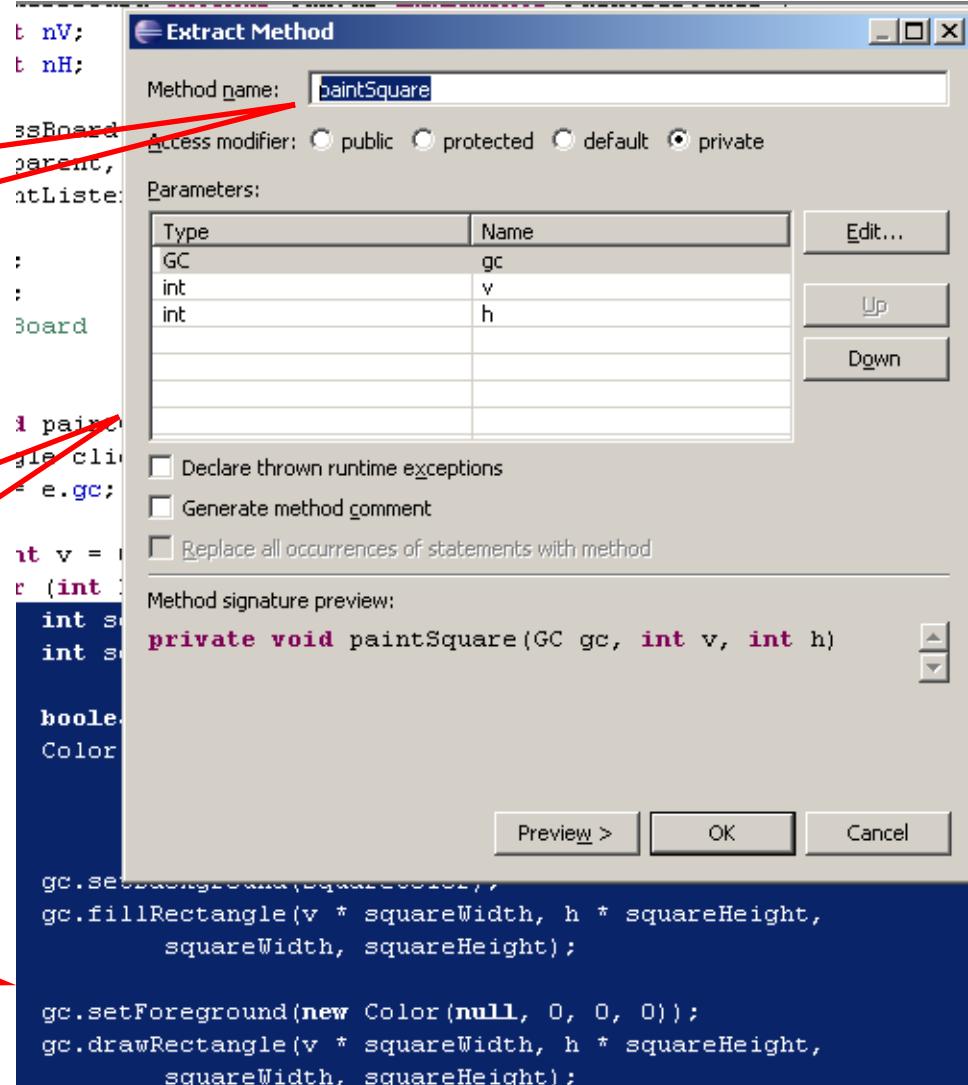
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

3. напечатать  
имя метода

2. Потом открыть  
диалоговое окно  
**ALT + Shift + M**

1. Сначала  
выделение  
текста



# Шахматы. Выделенный метод рисования клетки шахмат (двуцветная доска)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
private void drawSquare(GC gc, int v, int h) {  
    int squareWidth = getClientArea().width / nV;  
    int squareHeight = getClientArea().height / nH;  
  
    boolean isWhiteSquare = ((v + h) % 2 == 0);  
    Color squareColor = isWhiteSquare  
        ? new Color(null, 255, 255, 255) : new Color(null, 0, 192, 0);  
  
    gc.setBackground(squareColor);  
    gc.fillRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
  
    gc.setForeground(new Color(null, 0, 0, 0));  
    gc.drawRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
}
```

# Шахматы. Вызов выделенного метода в универсальном методе рисования доски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public void paintControl(PaintEvent e) {  
    Rectangle clientArea = getClientArea();  
  
    GC gc = e.gc;  
  
    for (int v = 0; v < nV; v++)  
        for (int h = 0; h < nH; h++) {  
            drawSquare(gc, v, h);  
        }  
  
    gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);  
}
```

# Рефакторинг. Вынос универсальных полей и методов в базовый класс

МГУ им. М.В.Ломоносова. Факультет ВМК.

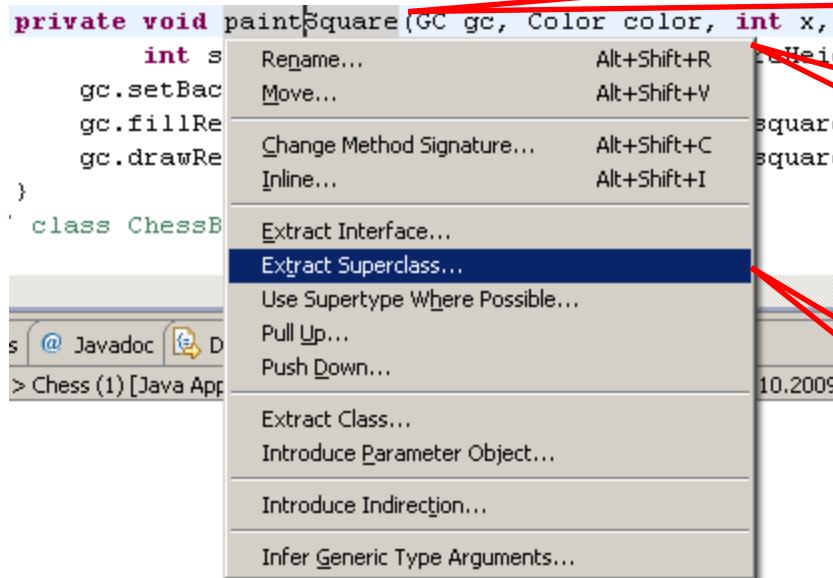
Романов Владимир Юрьевич ©2025

- Предпосылки
  - Размерность доски - поля  $nH$  и  $nV$  общие для всех клеточных игр
  - Алгоритм рисования в цикле всех клеток доски универсален

# Рефакторинг. Создание абстрактного класса *GameBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



1. Сначала  
выделение имени

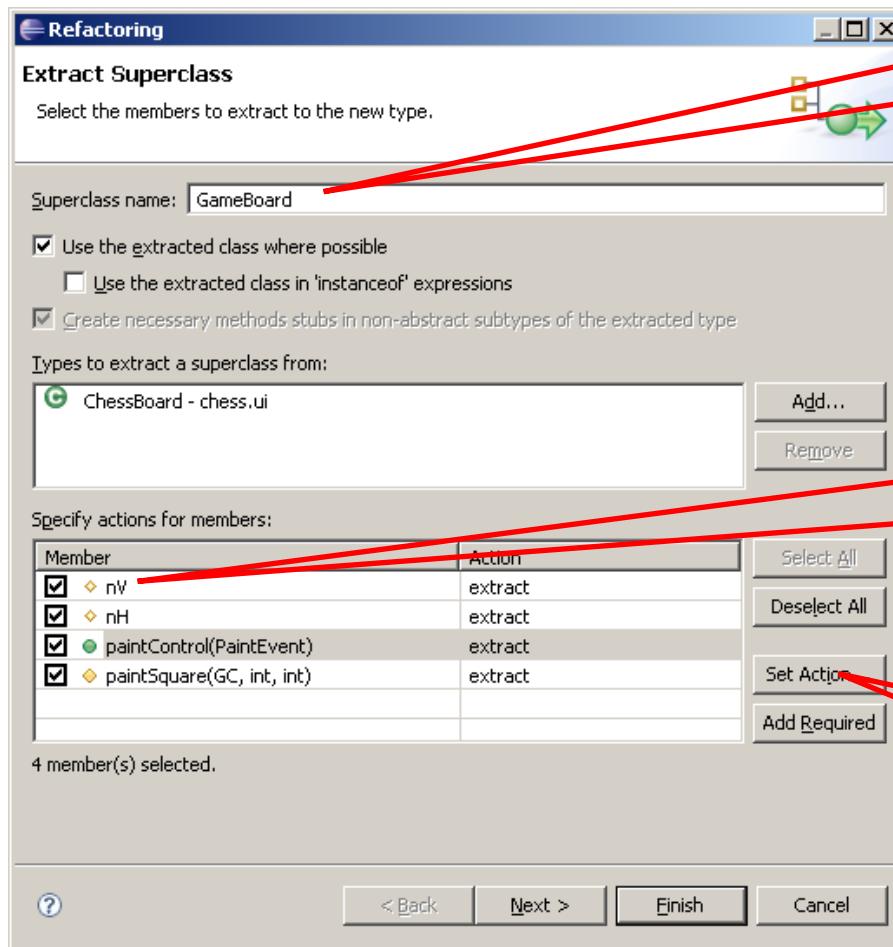
2. Переход к меню  
допустимых рефакторингов  
**ALT + Shift + T**

3. Выбор  
рефакторинга  
**Extract Superclass**

# Рефакторинг. Создание абстрактного класса *GameBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



1. Ввод имени суперкласса

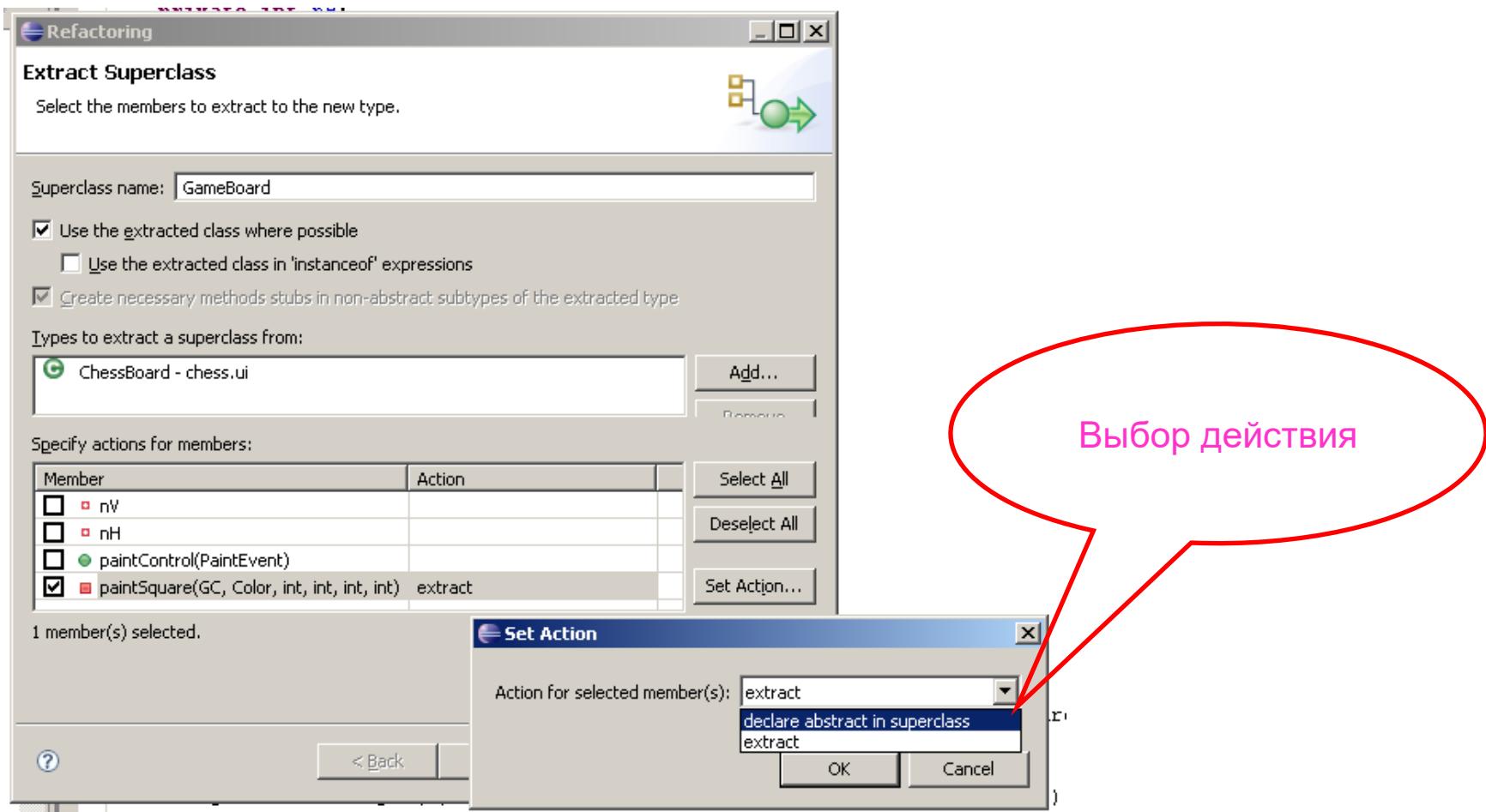
2. Выбор переносимых в суперкласс полей и методов

3. Выбор действия умалчиваемое действие extract

# Рефакторинг. Создание абстрактного метода в классе *GameBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Шахматы. Выделенный суперкласс

## *GameBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class GameBoard extends Canvas {  
    protected int nV;  
    protected int nH;  
    protected abstract void paintSquare(GC gc, int v, int h);  
  
    public GameBoard(Composite parent, int style)  
    { super(parent, style); }  
  
    public void paintControl(PaintEvent e) {  
        Rectangle clientArea = getClientArea();  
        GC gc = e.gc;  
        for (int v = 0; v < nV; v++)  
            for (int h = 0; h < nH; h++)  
                paintSquare(gc, v, h);  
        e.gc.drawRectangle(0, 0, clientArea.width - 1, clientArea.height - 1);  
    } // paintControl  
}
```

Предок  
выделенного  
суперкласса

Переопределяемый  
метод *paintSquare*

# Шахматы. *ChessBoard* – потомок

## выделенного суперкласса *GameBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
import org.eclipse.swt.events.MouseEvent;
import org.eclipse.swt.events.MouseListener;
import org.eclipse.swt.graphics.Color;
import org.eclipse.swt.graphics.Cursor;
import org.eclipse.swt.graphics.GC;
import org.eclipse.swt.graphics.Image;
import org.eclipse.swt.graphics.ImageData;
import org.eclipse.swt.widgets.Composite;

import chess.ChessImages;

public class ChessBoard extends GameBoard {

    // ...
}
```

Потомок  
выделенного  
суперкласса



# Перенос реализации интерфейса PaintListener в базовый класс

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

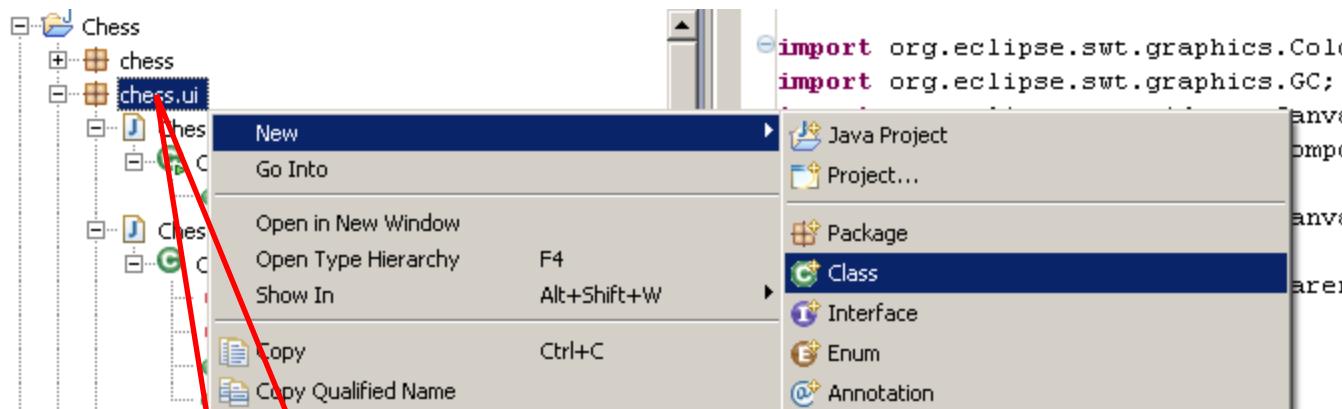
```
ChessBoard.java GameBoard.java X
1 package chess.ui;
2
3+ import org.eclipse.swt.events.PaintEvent;□
4
5 public abstract class GameBoard extends Canvas implements PaintListener {
6     public GameBoard(Composite parent, int style) {
7         super(parent, style);
8         addPaintListener(this);
9     }
10 }
```

```
ChessBoard.java X GameBoard.java
4 import org.eclipse.swt.graphics.Color;
5 import org.eclipse.swt.graphics.GC;
6 import org.eclipse.swt.widgets.Composite;
7
8 public class ChessBoard extends GameBoard implements PaintListener{
9
10     public ChessBoard(Composite parent, int style) {
11         super(parent, style);
12         addPaintListener(this);
13     }
14 }
```

# Создание класса AsiaBoard - потомка класса *GameBoard* (wizard)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Создание класса в  
пакете chess.ui

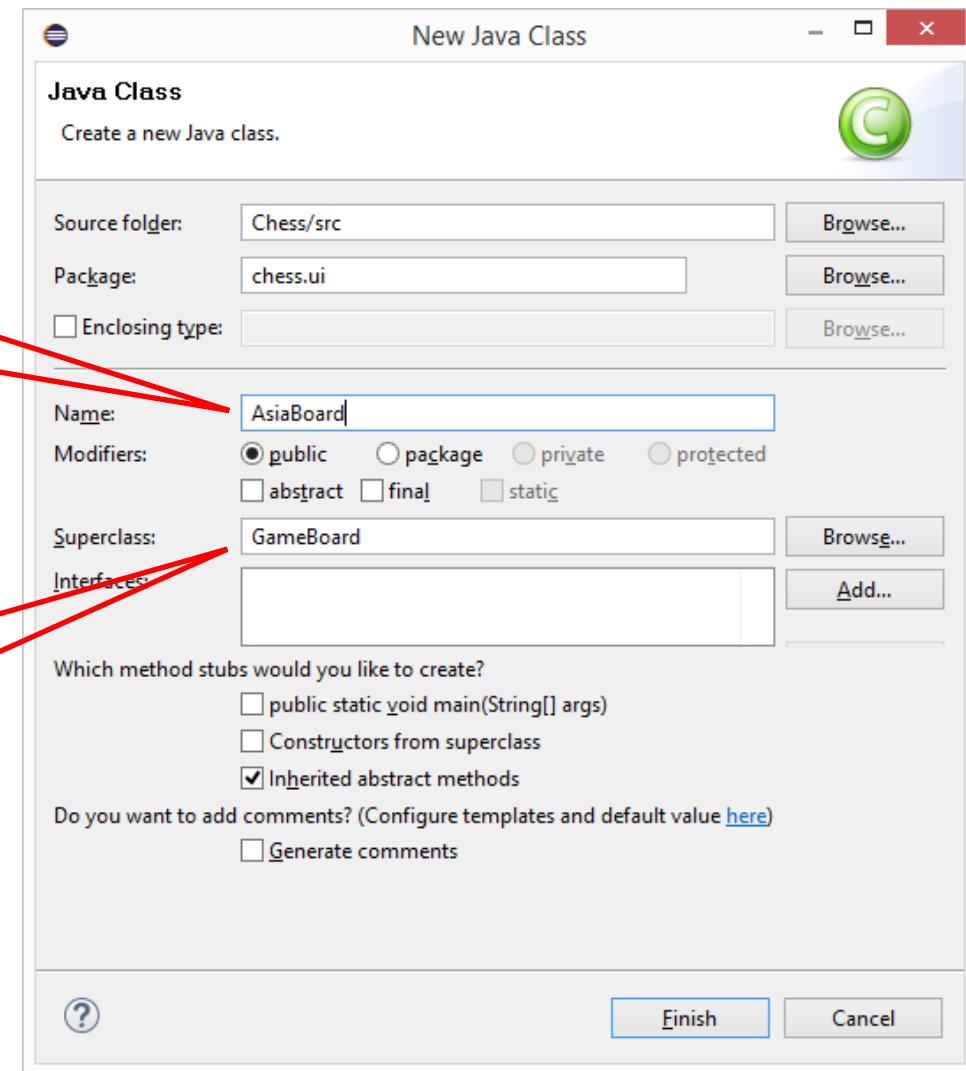
# Создание класса *AsiaBoard* - потомка класса *GameBoard* (wizard)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Задание имени  
класса

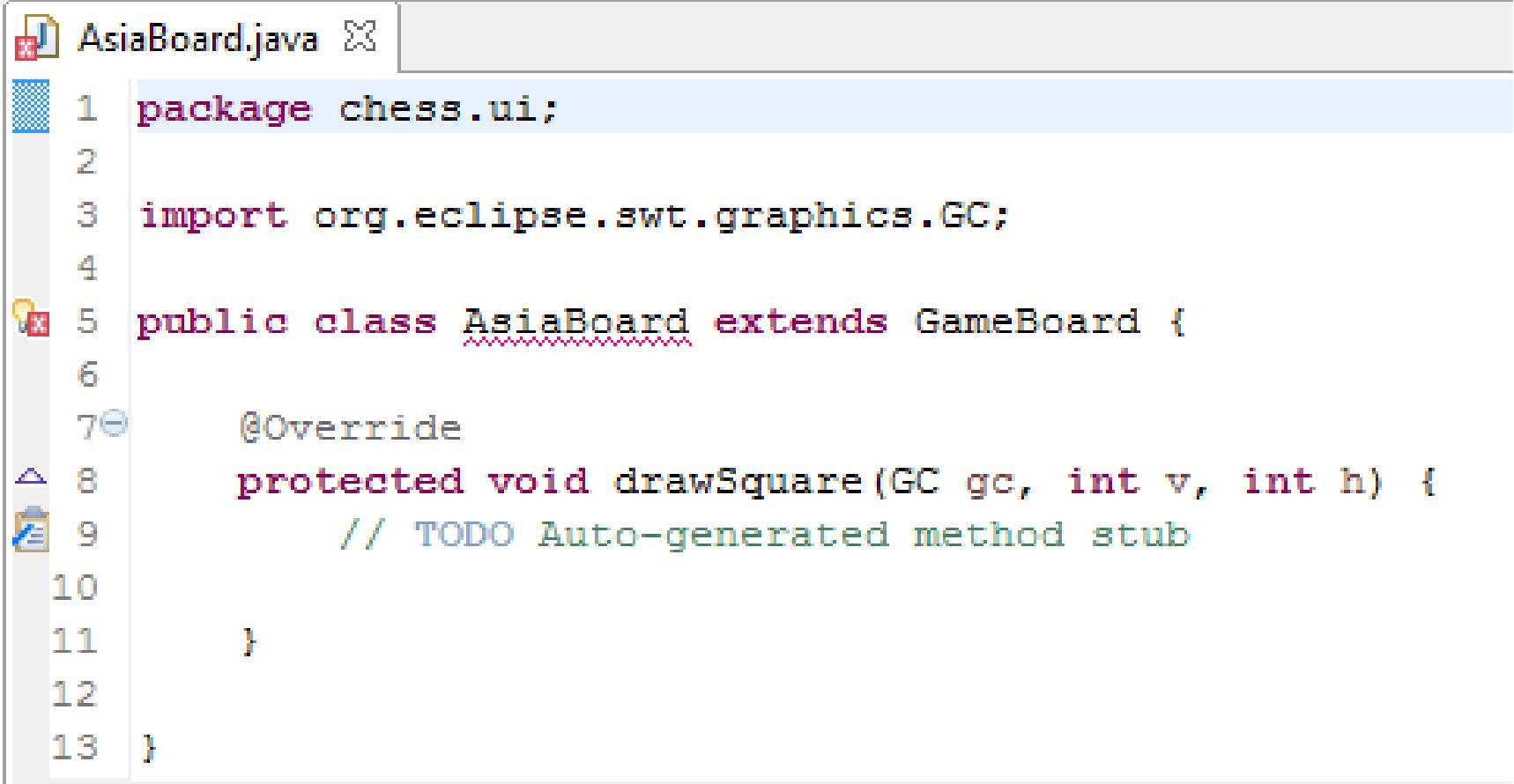
Задание имени  
базового класса



# Создан класс *AsiaBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



The screenshot shows a Java code editor window with the file `AsiaBoard.java` open. The code defines a class `AsiaBoard` that extends `GameBoard`. The `drawSquare` method is overridden and contains a TODO comment. The code editor interface includes a toolbar on the left with icons for file operations like new, open, save, and cut/paste, and a status bar at the bottom.

```
1 package chess.ui;
2
3 import org.eclipse.swt.graphics.GC;
4
5 public class AsiaBoard extends GameBoard {
6
7     @Override
8     protected void drawSquare(GC gc, int v, int h) {
9         // TODO Auto-generated method stub
10
11     }
12
13 }
```

# Создание конструктора класса *AsiaBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
1 package chess.ui;
2
3 import org.eclipse.swt.graphics.GC;
4
5 public class AsiaBoard extends GameBoard {
6
7     @Override
8     protected // TO
9
10 }
11
12
13
```

Implicit super constructor GameBoard() is undefined for default constructor. Must define an explicit constructor

1 quick fix available:

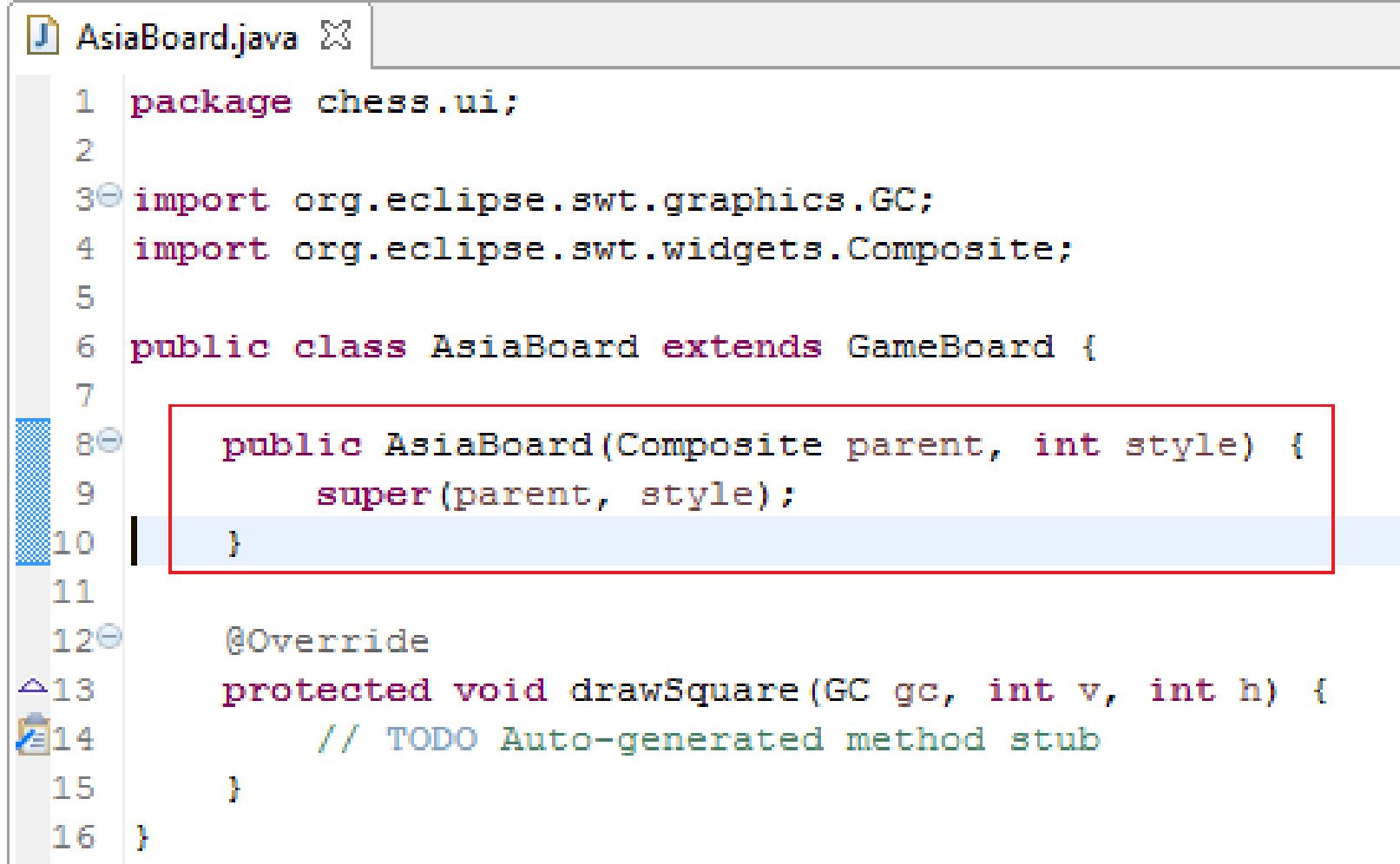
Add constructor 'AsiaBoard(Composite,int)'

Press 'F2' for focus

# Конструктор класса *AsiaBoard* создан

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



```
1 package chess.ui;
2
3 import org.eclipse.swt.graphics.GC;
4 import org.eclipse.swt.widgets.Composite;
5
6 public class AsiaBoard extends GameBoard {
7
8     public AsiaBoard(Composite parent, int style) {
9         super(parent, style);
10    }
11
12     @Override
13     protected void drawSquare(GC gc, int v, int h) {
14         // TODO Auto-generated method stub
15     }
16 }
```

# Шахматы. *AsiaBoard* – потомок суперкласса *GameBoard* (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
package chess.ui;

import org.eclipse.swt.graphics.GC;
import org.eclipse.swt.widgets.Composite;

public class AsiaBoard extends GameBoard {

    public AsiaBoard(Composite parent, int style) {
        super(parent, style);
    }

    @Override
    protected void drawSquare(GC gc, int v, int h) {
        // TODO Auto-generated method stub
    }
}
```

Потомок  
суперкласса  
*GameBoard*

# Шахматы. *AsiaBoard* – потомок суперкласса *GameBoard* (2)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

@Override

```
protected void drawSquare(GC gc, int v, int h) {
    int squareWidth = getClientArea().width / nV;
    int squareHeight = getClientArea().height / nH;

    gc.setBackground(new Color(null, 255, 255, 0));
    gc.fillRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);

    int x = v * squareWidth + squareWidth/2;
    int y = h * squareHeight + squareHeight/2;

    if (v != 0) gc.drawLine(x, y, x - squareWidth/2, y);
    if (v != nV-1) gc.drawLine(x, y, x + squareWidth/2, y);

    if (h != 0) gc.drawLine(x, y, x, y - squareHeight/2);
    if (h != nH-1) gc.drawLine(x, y, x, y + squareHeight/2);
}
```

Реализация  
абстрактного метода  
*GameBoard.drawSquare*

# Блокнот игр. Доска для игр:

## *Шахматы, Ренджю*

МГУ им. М.В.Ломоносова. Факультет ВМК.

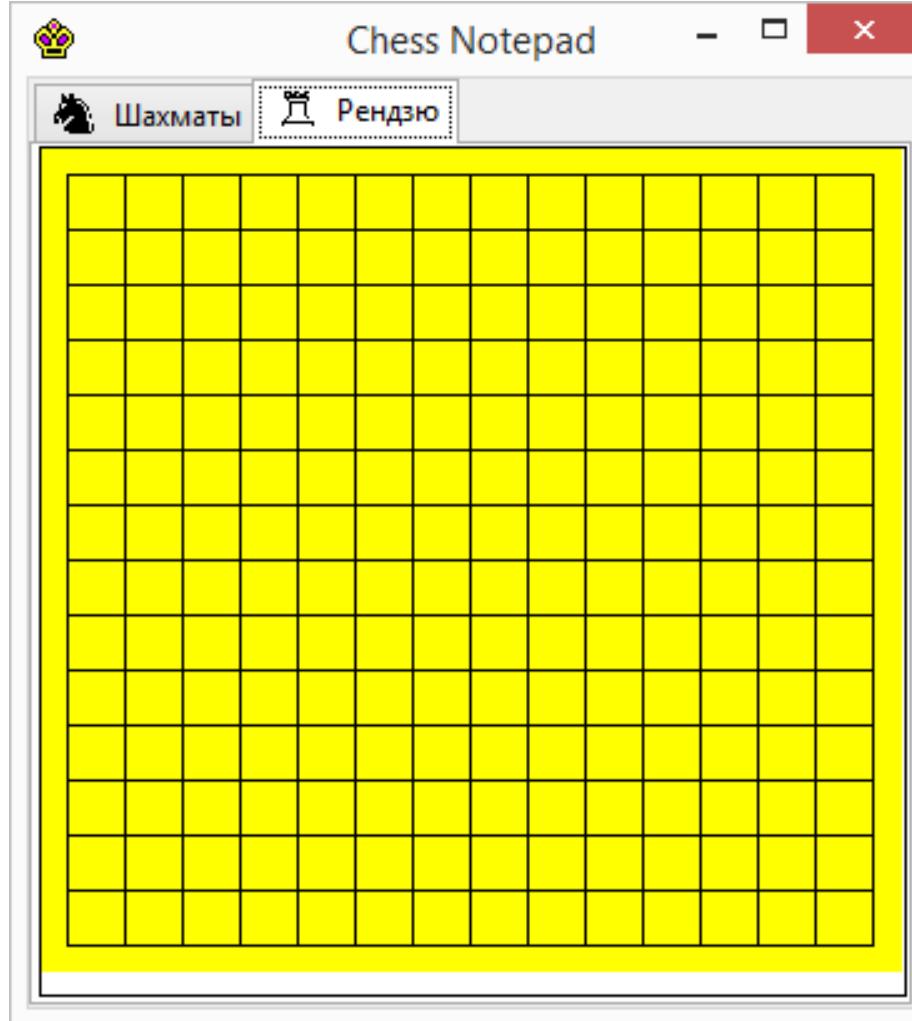
Романов Владимир Юрьевич ©2025

```
Java AsiaBoard.java Chess.java X
29         FillLayout layout = new FillLayout();
30         shell.setLayout(layout);
31
32         TabFolder gamesFolder = new TabFolder(shell, SWT.TOP);
33
34         Image chessImage = new Image(display, ChessImages.imageKnightBlack
35             .getImageData().scaledTo(20, 20));
36
37         TabItem chessItem = new TabItem(gamesFolder, SWT.NONE);
38         chessItem.setText("Шахматы");
39         chessItem.setControl(new ChessBoard(gamesFolder, SWT.NONE));
40         chessItem.setImage(chessImage);
41
42         final Image renjuTabImage = new Image(display,
43             ChessImages.imageRookWhite.getImageData().scaledTo(20, 20));
44
45         TabItem renjuItem = new TabItem(gamesFolder, SWT.NULL);
46         renjuItem.setText("Ренджю");
47         renjuItem.setImage(renjuTabImage);
48         renjuItem.setControl(new AsiaBoard(gamesFolder, SWT.NONE));
49
```

# Блокнот игр. Доска для игр: *Шахматы, Ренджю*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# БЛОКНОТ игр. Игры на шахматной доске

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
final TabFolder gamesFolder = new TabFolder(shell, SWT.TOP);
```

```
final Image chessTabImage = new Image(display,  
ChessImages.imageKnightBlack.getImageData().scaledTo(20, 20));
```

```
TabItem chessItem = new TabItem(gamesFolder, SWT.NULL);  
chessItem.setText("Шахматы");  
chessItem.setImage(chessTabImage);  
chessItem.setControl(new ChessBoard(gamesFolder, SWT.NONE));
```

```
final Image checkersTabImage = new Image(display,  
ChessImages.imageCheckersNotebook.getImageData().scaledTo(20, 20));
```

```
TabItem checkersItem = new TabItem(gamesFolder, SWT.NULL);  
checkersItem.setText("Шашки");  
checkersItem.setImage(checkersTabImage);  
checkersItem.setControl(new ChessBoard(gamesFolder, SWT.NONE));
```

Блокнот

# БЛОКНОТ ИГР. ИГРЫ НА «ЯПОНСКОЙ» ДОСКЕ

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
final Image renjuTabImage = new Image(display,  
    ChessImages.imageRenjuNotebook.getImageData().scaledTo(20, 20));
```

```
TabItem renjulItem = new TabItem(gamesFolder, SWT.NULL);  
renjulItem.setText("Рэндзю");  
renjulItem.setImage(renjuTabImage);  
renjulItem.setControl(new AsiaBoard(gamesFolder, SWT.NONE));
```

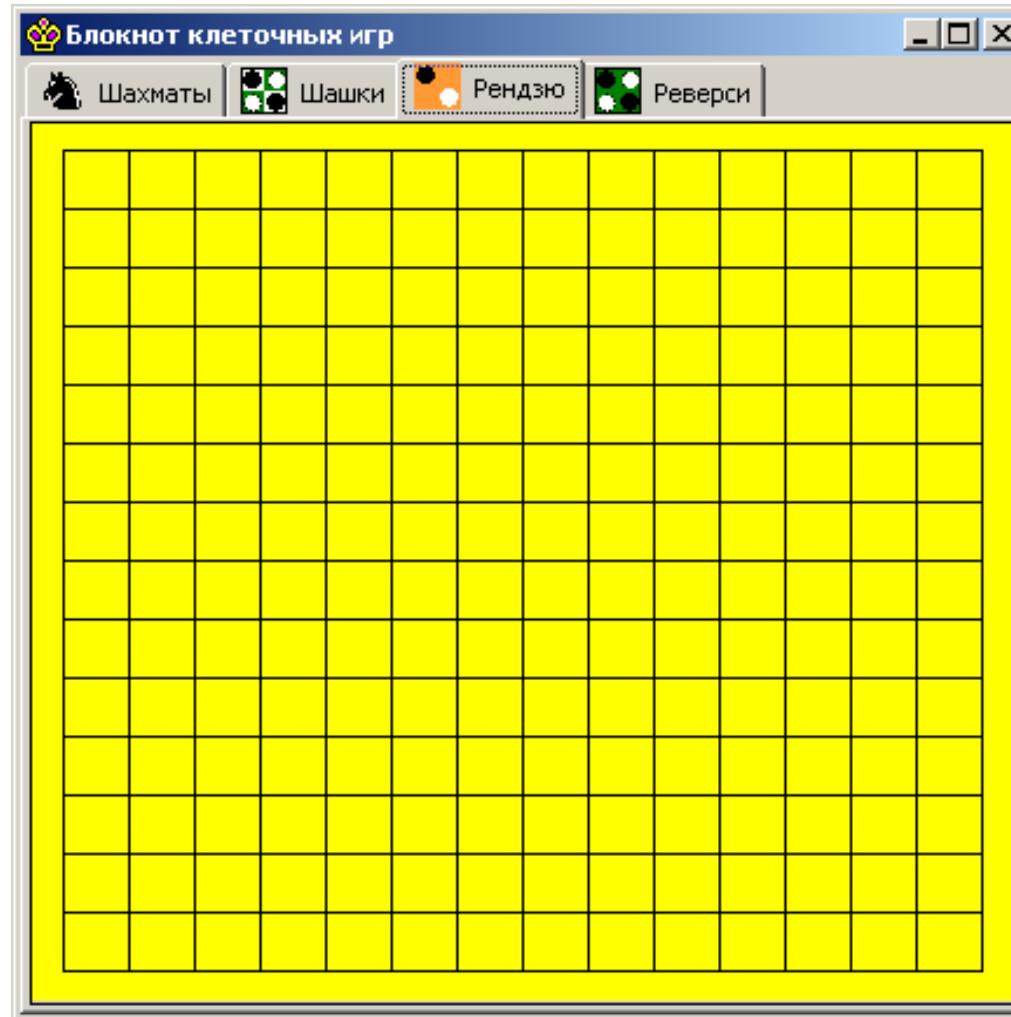
```
final Image reversiTabImage = new Image(display,  
    ChessImages.imageReversiNotebook.getImageData().scaledTo(20, 20));
```

```
TabItem reversilItem = new TabItem(gamesFolder, SWT.NULL);  
reversilItem.setText("Реверси");  
reversilItem.setImage(reversiTabImage);  
reversilItem.setControl(new AsiaBoard(gamesFolder, SWT.NONE));
```

# Блокнот игр. Доска для игры Ренджю, Го, Сянци (китайские шахматы)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



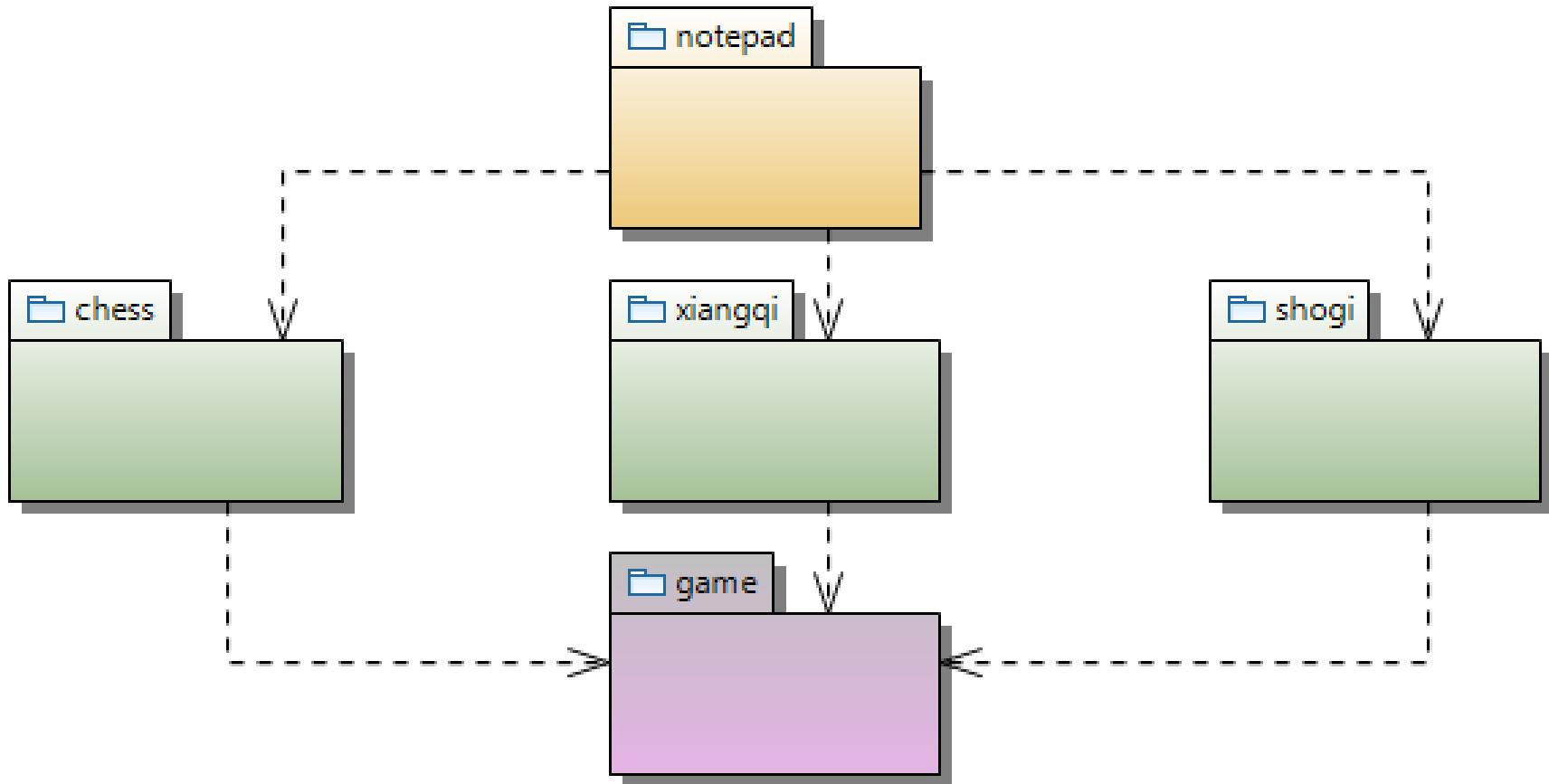
# Проектирование архитектуры.

## Выделение уровней в настольных играх

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

### Идентификация пакетов верхнего уровня

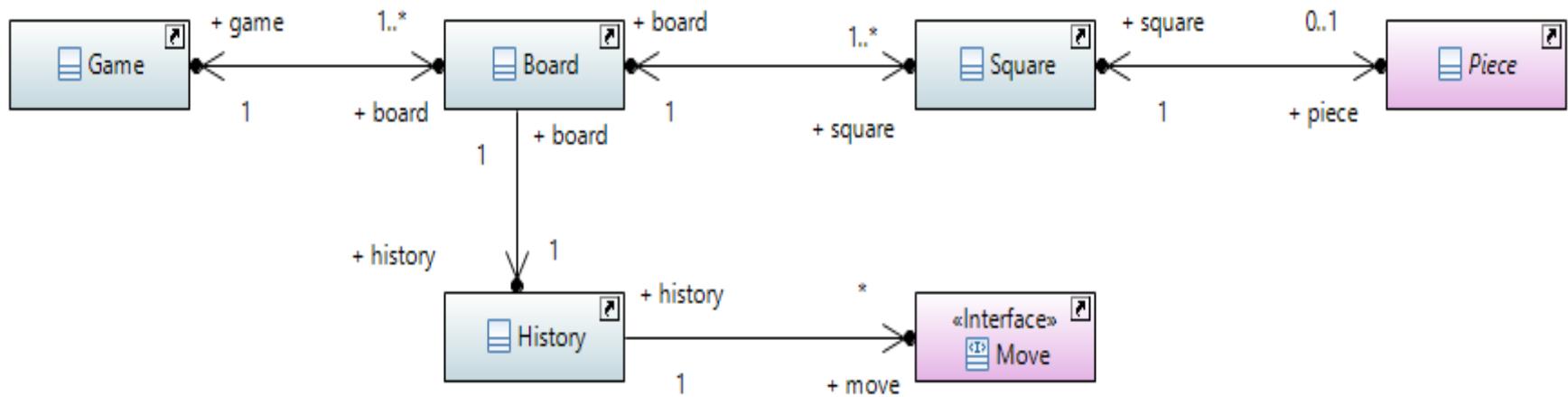


# Проектирование архитектуры.

## Классы пакета game.core

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

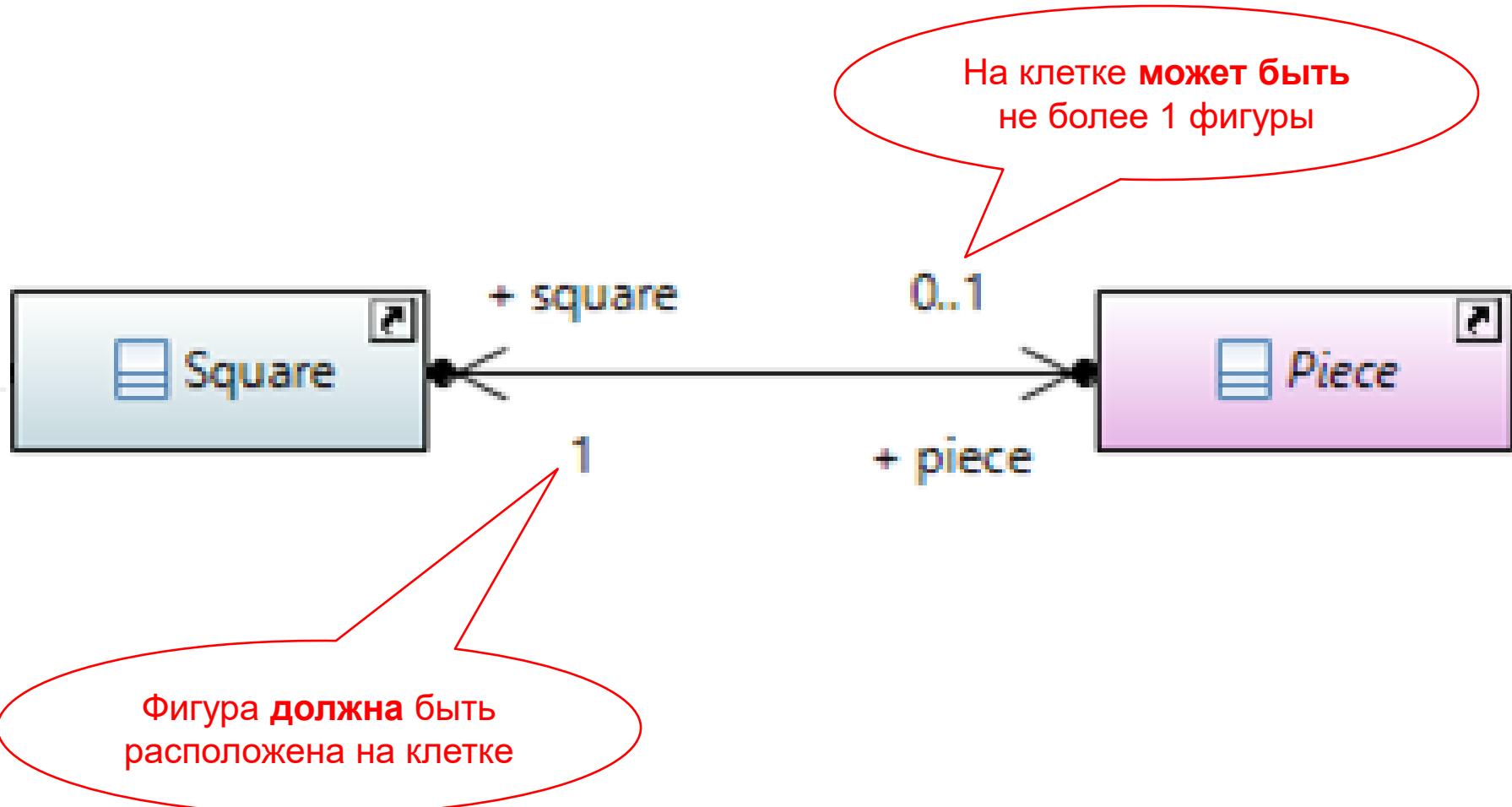


# Отношение ассоциации между классам

## Piece и Square

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Перечисление *PieceColor* - цвет фигуры

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
package game.core;

public enum PieceColor {
    WHITE,
    BLACK,
    GREEN,
    BLUE
}
```

# Класс Piece (фигура)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
abstract public class Piece {  
    private Square square;  
    PieceColor color;  
  
    public Piece(Square square, PieceColor color) {  
        setSquare(square);  
        this.color = color;  
    }  
  
    public Square getSquare() {  
        return square;  
    }  
  
    public void setSquare(Square square) {  
        this.square = square;  
        square.piece = this;  
    }  
    // ...  
}
```

Фигура **должна** быть  
расположена на клетке

# Класс Square (клетка)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public class Square {  
    public Piece piece;  
  
    public Board board;  
    public int v;  
    public int h;  
  
    public Square(Board board, int v, int h) {  
        this.board = board;  
        this.v = v;  
        this.h = h;  
    }  
  
    public Piece getPiece() {  
        return piece;  
    }  
  
    public void putPiece(Piece piece) {  
        this.piece = piece;  
        piece.setSquare(this);  
    }  
}
```

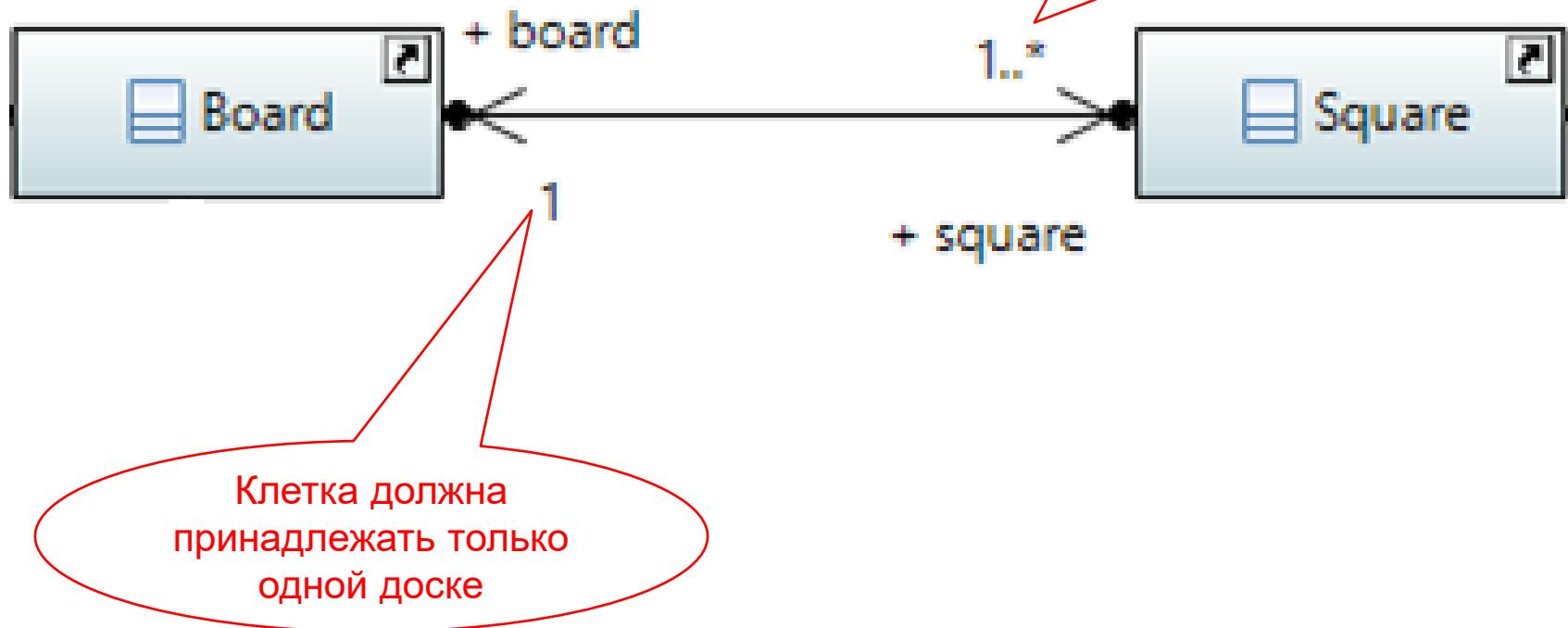
На клетке **может быть**  
не более 1 фигуры

# Отношение ассоциации между классам *Board* и *Square*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

На доске может быть  
1 и более клеток



# Класс Square (клетка)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public class Square {  
    public Piece piece;  
  
    public Board board;  
    public int v;  
    public int h;  
  
    public Square (Board board, int v, int h) {  
        this.board = board;  
        this.v = v;  
        this.h = h;  
    }  
  
    public Piece getPiece() {  
        return piece;  
    }  
  
    public Board getBoard() {  
        return board;  
    }  
}
```

Клетка **должна**  
принадлежать только  
одной доске

# Класс Board (доска)

МГУ им. М.В.Ломоносова. Факультет ВМК.

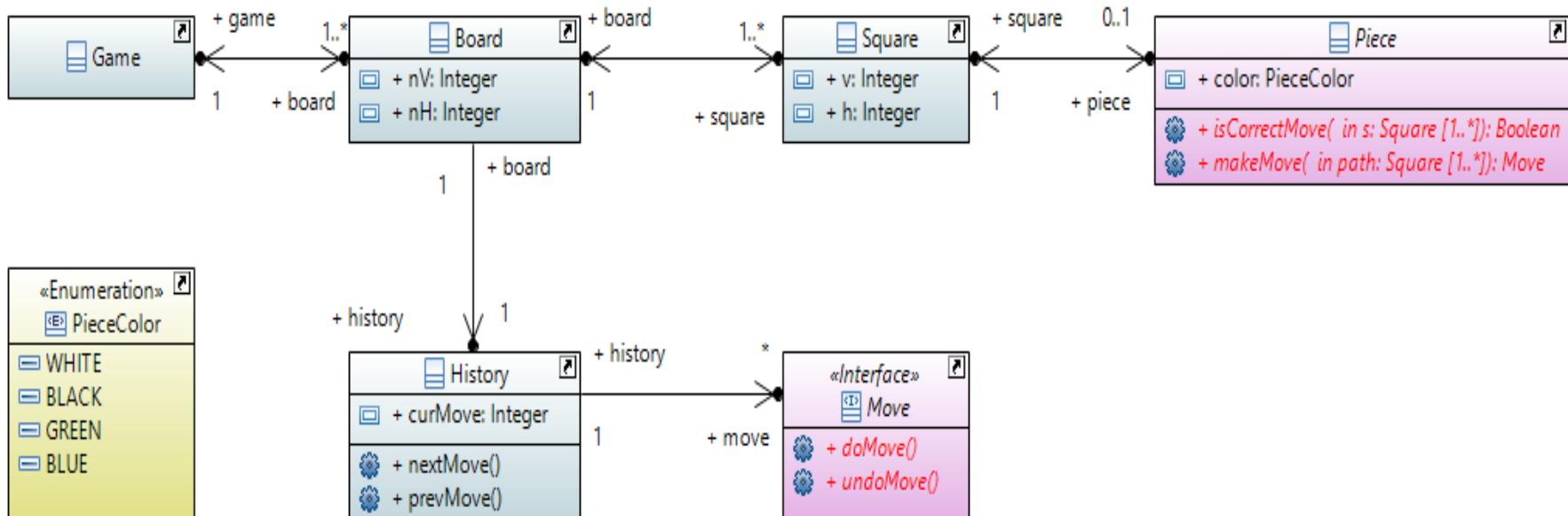
Романов Владимир Юрьевич ©2025

```
public class Board {  
    private Square [ ][ ] squares;  
  
    public Board(int nV, int nH) {  
        squares = new Square[nV][nH];  
  
        for (int v = 0; v < nV; v++)  
            for (int h = 0; h < nH; h++)  
                squares[v][h] = new Square(this, v, h);  
    }  
  
    public boolean isEmpty(int v, int h) {  
        return getSquare(v, h).piece == null;  
    }  
  
    public Square getSquare(int v, int h) {  
        return squares[v][h];  
    }  
}
```

# Атрибуты и методы классов ядра настольных игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Блокнот игр. Шахматные фигуры

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

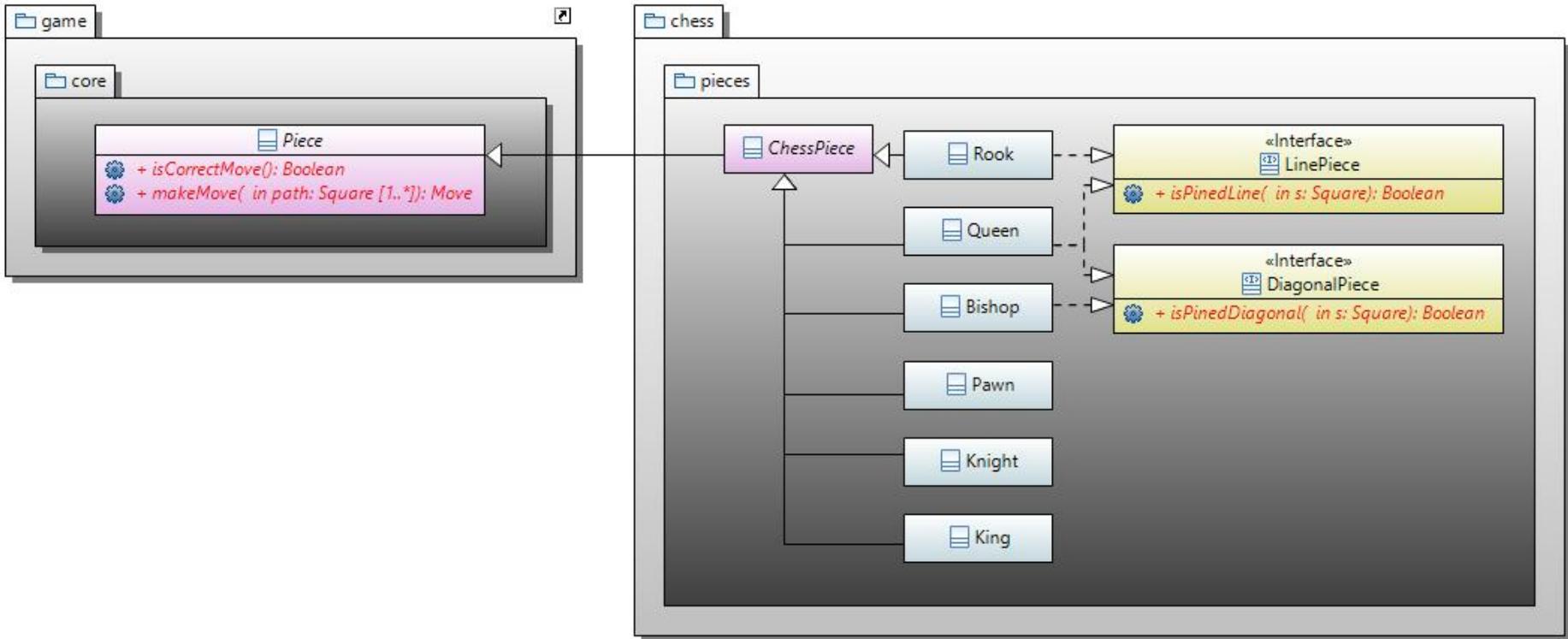


# Проектирование классов

## Классы-фигуры в шахматной программе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

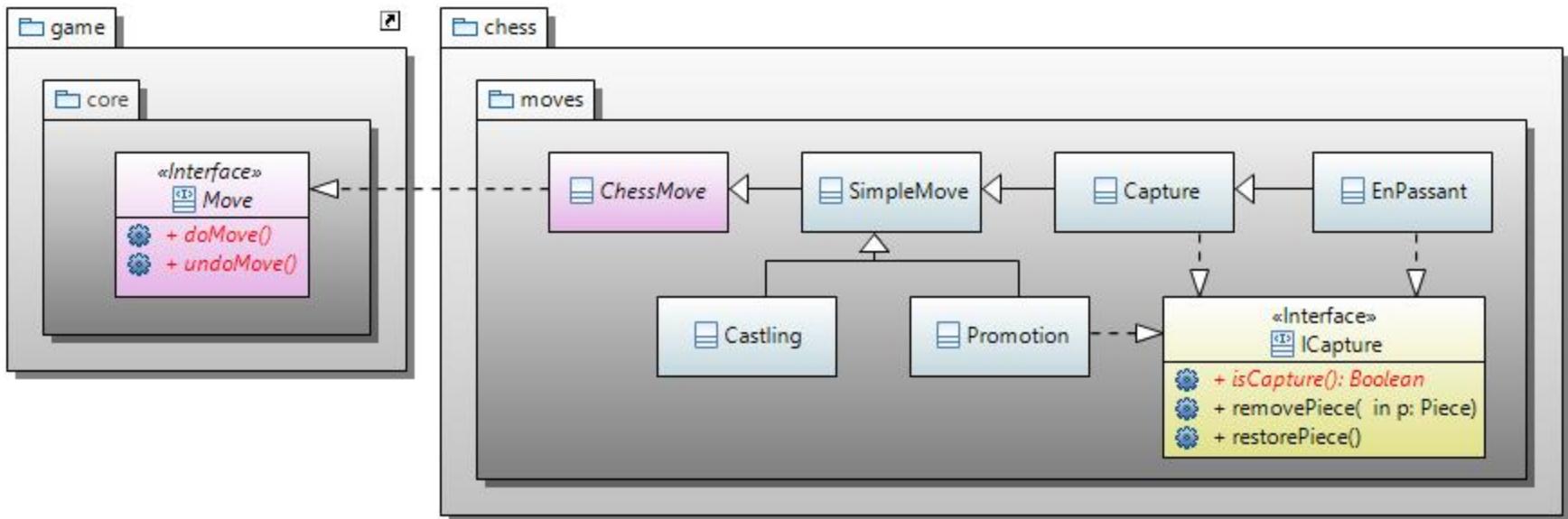


# Проектирование классов

## Классы-ходы в шахматной программе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

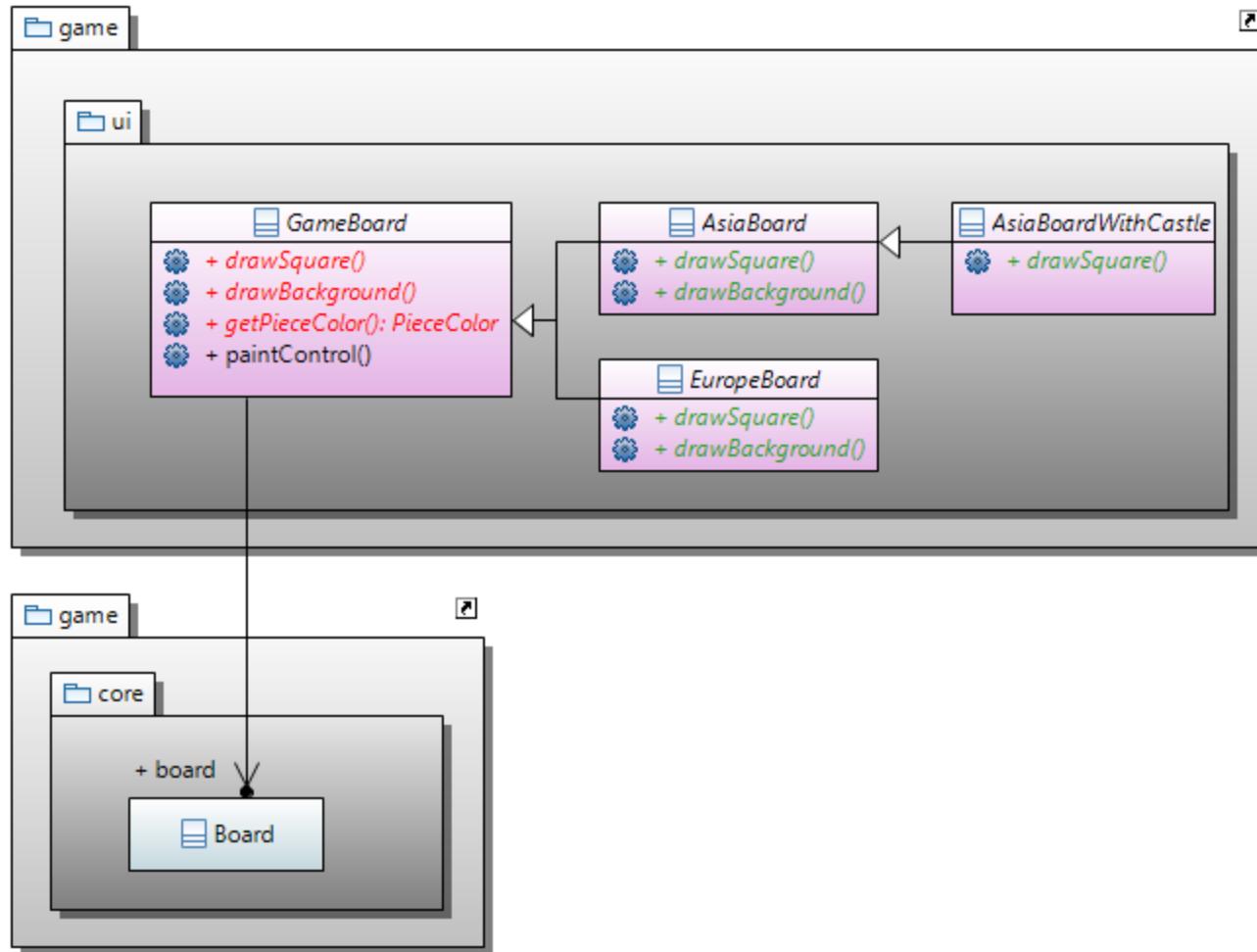


# Проектирование классов

## Базовые классы интерфейса пользователя

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Объявление абстрактных методов в абстрактном классе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class GameBoard extends Canvas implements PaintListener {  
    // ...  
    protected Board board;  
  
    public GameBoard(Composite parent, Board board) {  
        super(parent, SWT.NONE);  
  
        this.board = board;  
  
        addPaintListener(this);  
    }  
    // ...  
}
```

# Объявление абстрактных методов в абстрактном классе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class GameBoard extends Canvas implements PaintListener {  
    // ...  
    abstract  
    protected void drawBackground (GC gc, Rectangle area);  
  
    abstract  
    public Image getPieceImage (Piece piece);  
  
    abstract  
    public void drawSquare (GC gc, int v, int h);  
    // ...  
}
```

# Использование абстрактных методов в абстрактном классе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class GameBoard extends Canvas implements PaintListener {  
    // ...  
    @Override  
    public void paintControl (PaintEvent e) {  
        GC gc = e.gc;  
        Rectangle clientArea = getClientArea();  
  
        drawBackground(gc, clientArea);  
  
        int squareWidth = getClientArea().width / board.nV;  
        int squareHeight = getClientArea().height / board.nH;  
  
        for (int v = 0; v < board.nV; v++) {  
            for (int h = 0; h < board.nH; h++) {  
                drawSquare(gc, v, h);  
  
                drawPiece(gc, v, h);  
            }  
        }  
    }  
    // ...
```

# Использование абстрактных методов в абстрактном классе

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class GameBoard extends Canvas implements PaintListener {  
    // ...  
  
    private void drawPiece (GC gc, int v, int h, int squareWidth, int squareHeight) {  
        Piece piece = board.getSquare(v, h).getPiece();  
        if (piece == null) return;  
  
        int squareWidth = getClientArea().width / board.nV;  
        int squareHeight = getClientArea().height / board.nH;  
  
        int dx = squareWidth / 8;  
        int dy = squareHeight / 8;  
        int x = v * squareWidth + dx;  
        int y = h * squareHeight + dy;  
  
        Image image = getPieceImage(piece);  
        Rectangle bounds = image.getBounds();  
        gc.drawImage(image,  
                    0, 0, bounds.width, bounds.height,  
                    x, y, squareWidth - 2 * dx, squareHeight - 2 * dy);  
    }  
    // ...
```

# Класс EuropeBoard (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
abstract public class EuropeBoard extends GameBoard implements PaintListener {  
    private static final Color BLACK = new Color(null, 0, 0, 0);  
    private static final Color WHITE = new Color(null, 255, 255, 255);  
    private static final Color GREEN = new Color(null, 0, 192, 0);  
  
    public EuropeBoard(Composite parent, Board board) {  
        super(parent, board);  
    }  
  
    @Override  
    public void drawSquare(GC gc, int v, int h, int sw, int sh) {  
        int sx = v * sw;  
        int sy = h * sh;  
  
        boolean isWhiteSquare = ((v + h) % 2 == 0);  
        Color squareColor = isWhiteSquare ? WHITE : GREEN;  
  
        gc.setBackground(squareColor);  
        gc.fillRectangle(sx, sy, sw, sh);  
  
        gc.setForeground(BLACK);  
        gc.drawRectangle(sx, sy, sw, sh);  
    }  
    // ...
```

# *Класс EuropeBoard (2)*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
// ...  
  
@Override  
protected void drawBackground(GC gc, Rectangle area) {  
    gc.setForeground(BLACK);  
    gc.drawRectangle(area);  
}  
}
```

# *Klacc AsiaBoard (1)*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
abstract public class AsiaBoard extends GameBoard {
    private static final Color YELLOW = new Color(null, 255, 255, 0);

    public AsiaBoard(Composite parent, Board board) {
        super(parent, board);
    }

    @Override
    public void drawSquare(GC gc, int v, int h) {
        int sw = getClientArea().width / board.nV;
        int sh = getClientArea().height / board.nH;

        int x = v * sw + sw/2;
        int y = h * sh + sh/2;

        if (v != 0) gc.drawLine(x, y, x - sw/2, y);
        if (v != nV-1) gc.drawLine(x, y, x + sw/2, y);

        if (h != 0) gc.drawLine(x, y, x, y - sh/2);
        if (h != nH-1) gc.drawLine(x, y, x, y + sh/2);
    }
    // ...
}
```

# *Klacc AsiaBoard (2)*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

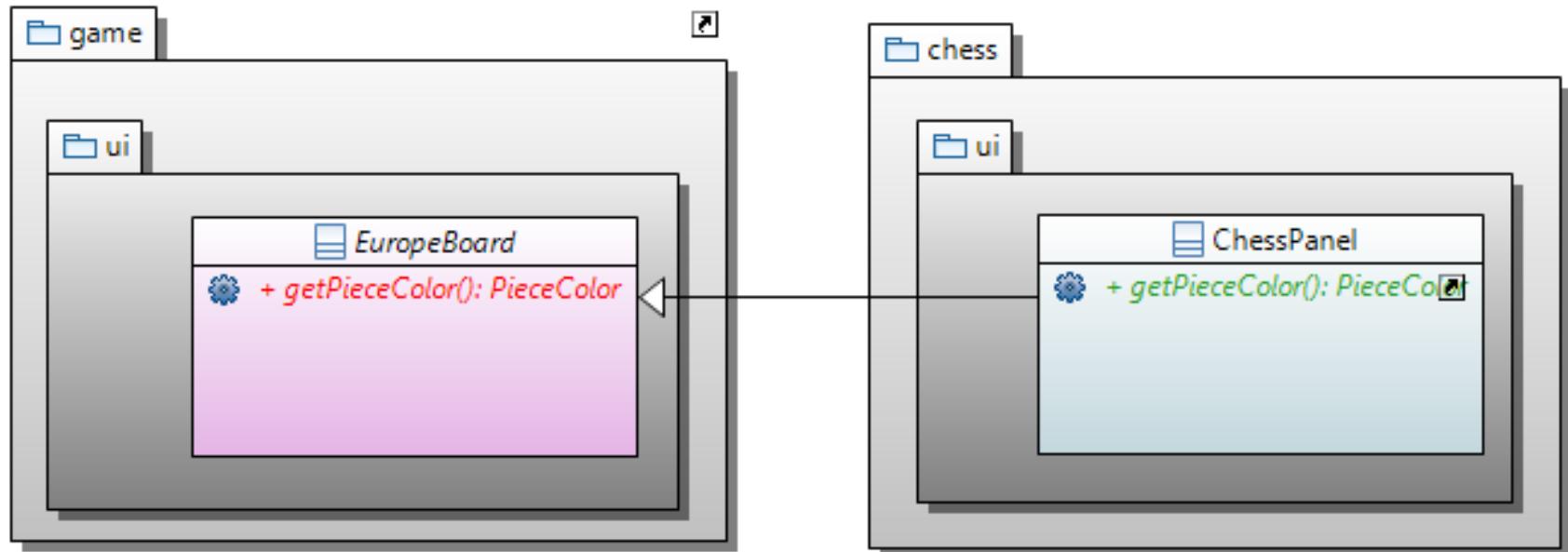
```
// ...  
  
@Override  
protected void drawBackground(GC gc, Rectangle area) {  
    gc.setBackground(YELLOW);  
    gc.fillRectangle(area);  
}  
}
```

# Проектирование классов

## Классы интерфейса пользователя шахмат

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



# Реализация некоторых абстрактных методов в абстрактном классе *EuropeBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class EuropeBoard extends GameBoard {  
    public EuropeBoard (Composite composite, Board board) {  
        super(composite, board);  
    }  
  
    @Override  
    protected void drawBackground (GC gc, Rectangle area) {  
        // Рисуем фон – деревянную доску.  
        Image image = Gamelimages.woodMedium;  
        Rectangle bounds = image.getBounds();  
  
        gc.drawImage(image,  
                    0, 0, bounds .width, bounds.height,  
                    area.x, area.y, area.width, area.height);  
    }  
}
```

# Реализация некоторых абстрактных методов в абстрактном классе *EuropeBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class EuropeBoard extends GameBoard {  
  
    @Override  
    public void drawSquare (GC gc, int v, int h, int squareWidth, int squareHeight) {  
        boolean isWhiteSquare = ((v + h) % 2 == 0);  
  
        Rectangle bounds = GameImages.woodDark.getBounds();  
  
        if ( !isWhiteSquare )  
            gc.drawImage(GameImages.imageWoodDark,  
                        0, 0, bounds.width, bounds.height,  
                        v * squareWidth, h * squareHeight,  
                        squareWidth, squareHeight);  
  
        gc.setForeground( new Color(null, 0, 0, 0 ) );  
        gc.drawRectangle(v * squareWidth, h * squareHeight, squareWidth, squareHeight);  
    }  
}
```

# Реализация некоторых абстрактных методов в абстрактном классе *AsiaBoard*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

```
public abstract class AsiaBoard extends GameBoard {  
  
    @Override  
    protected void drawBackground (GC gc, Rectangle area) {  
        // Рисуем фон – деревянную доску.  
        Rectangle bounds = GameImages.ImageWoodLight.getBounds();  
  
        gc.drawImage(GameImages.woodLight,  
                    0, 0, bounds.width, bounds.height,  
                    area.x, area.y, area.width, area.height);  
    }  
  
    @Override  
    public void drawSquare (GC gc, int v, int h, int squareWidth, int squareHeight)  
    {  
        // Рисуем клетку – пересечение линий в центре клетки  
        // ...  
    }  
}
```

# Блокнот игр. Доска для игры в азиатские игры Ренджю, Го, Сянци (китайские шахматы)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

