

Восьмая независимая
Научно-практическая конференция
«Разработка ПО 2012»



Software Engineering
Conference in Russia

Юзабилити-тестирование интегрированных сред разработки

Софья Чебанова
Александр Марков

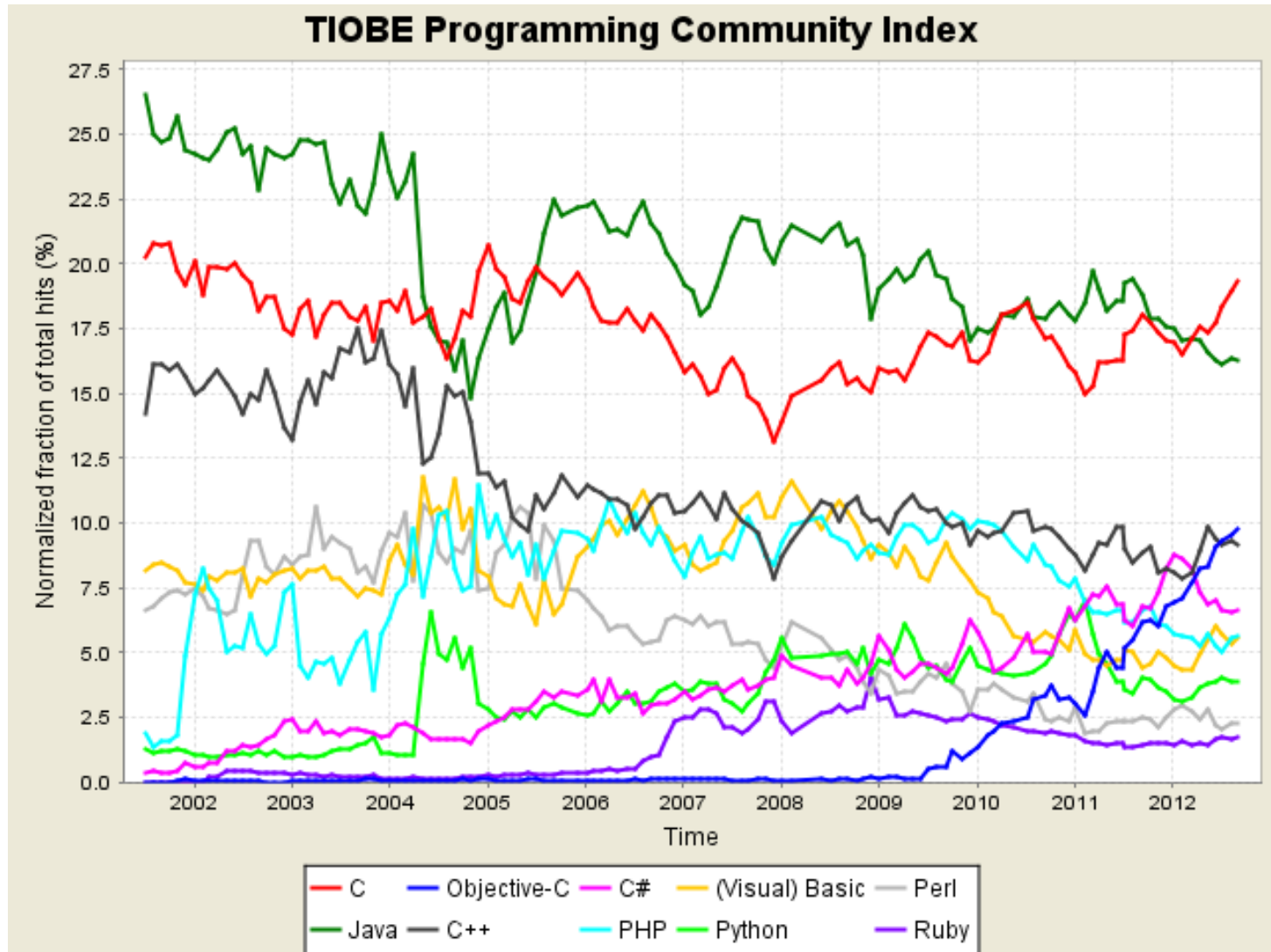
НИУ ИТМО

2 ноября 2012 г.

Цель исследования

Сравнение удобства использования (юзабилити) интерфейсов интегрированных сред разработки с помощью юзабилити-тестирования





Языки программирования



<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Объект исследования

Integrated Development Environments:

-  Eclipse IDE Juno
-  IntelliJ IDEA 11.1.3 Ultimate
-  NetBeans IDE 7.2
-  Microsoft Visual Studio 2010 Ultimate

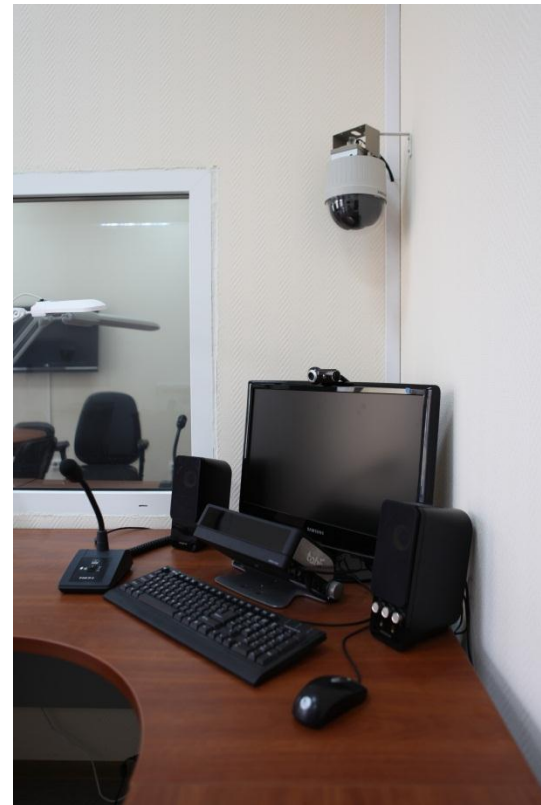
Что такое юзабилити-тестирование

Набор методик, позволяющих измерить характеристики взаимодействия пользователя с продуктом с целью оценки уровня юзабилити продукта

Алан Купер «Об интерфейсе»

Методика тестирования (1)

- Оборудованная лаборатория
- Eye Tracking



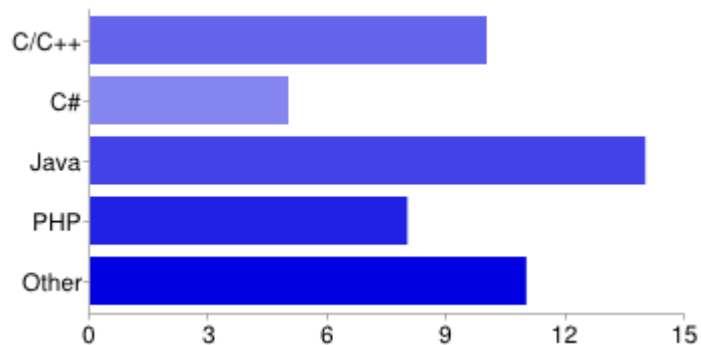
Методика тестирования (2)

- Разработка сценария
- Размышление респондента вслух
 - во время тестирования – *Concurrent Think Aloud*
 - после тестирования – *Retrospective Think Aloud*
- Анкетирование с применением шкалы *Likert Scale*

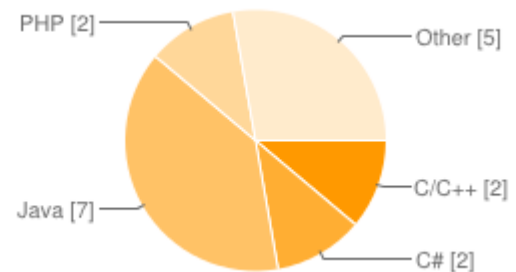
Респонденты (1)

Средний возраст = 23

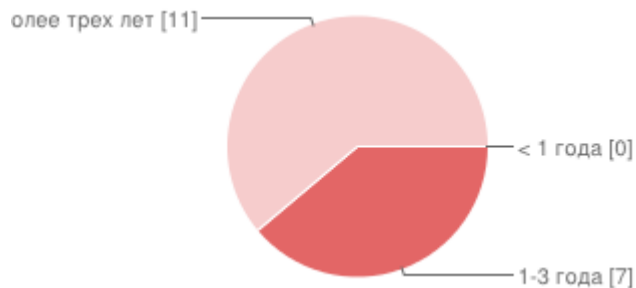
Какие языки программирования Вы использовали на практике?



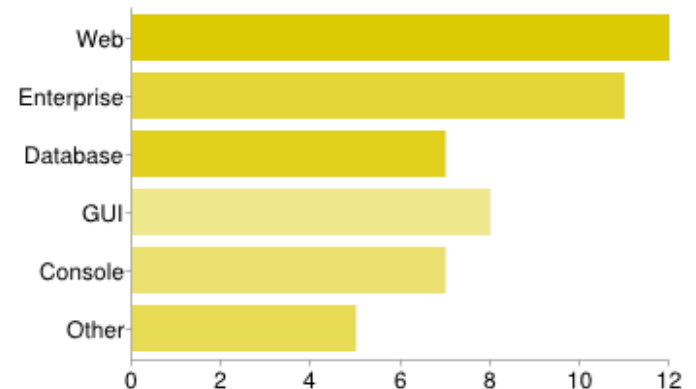
Укажите Ваш основной язык программирования.



Какой у Вас опыт разработки приложений на основном языке?

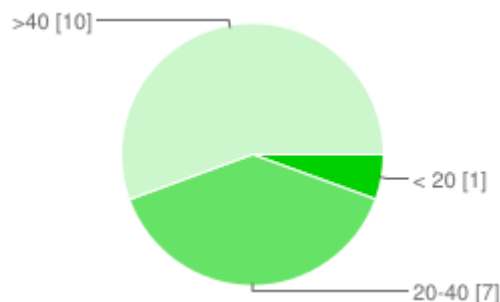


Какие типы приложений Вы разрабатываете/разрабатывали?

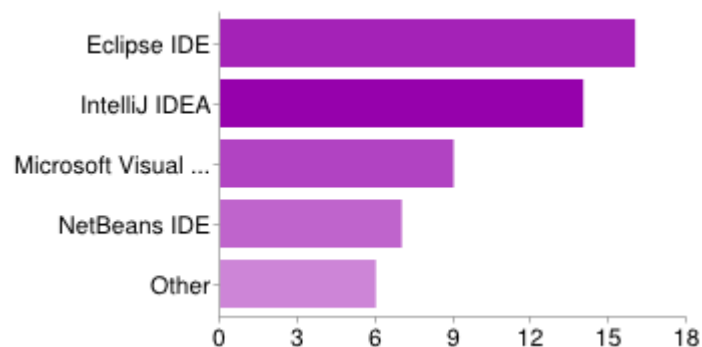


Респонденты (2)

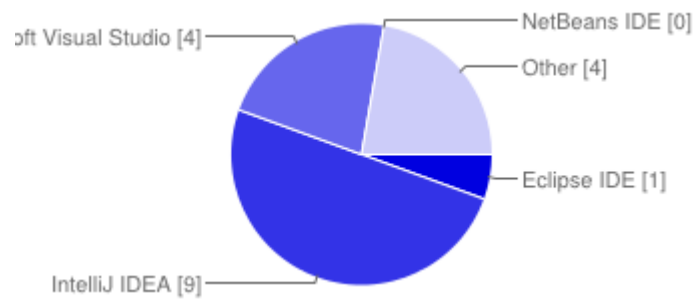
Сколько часов в неделю Вы занимаетесь программированием?



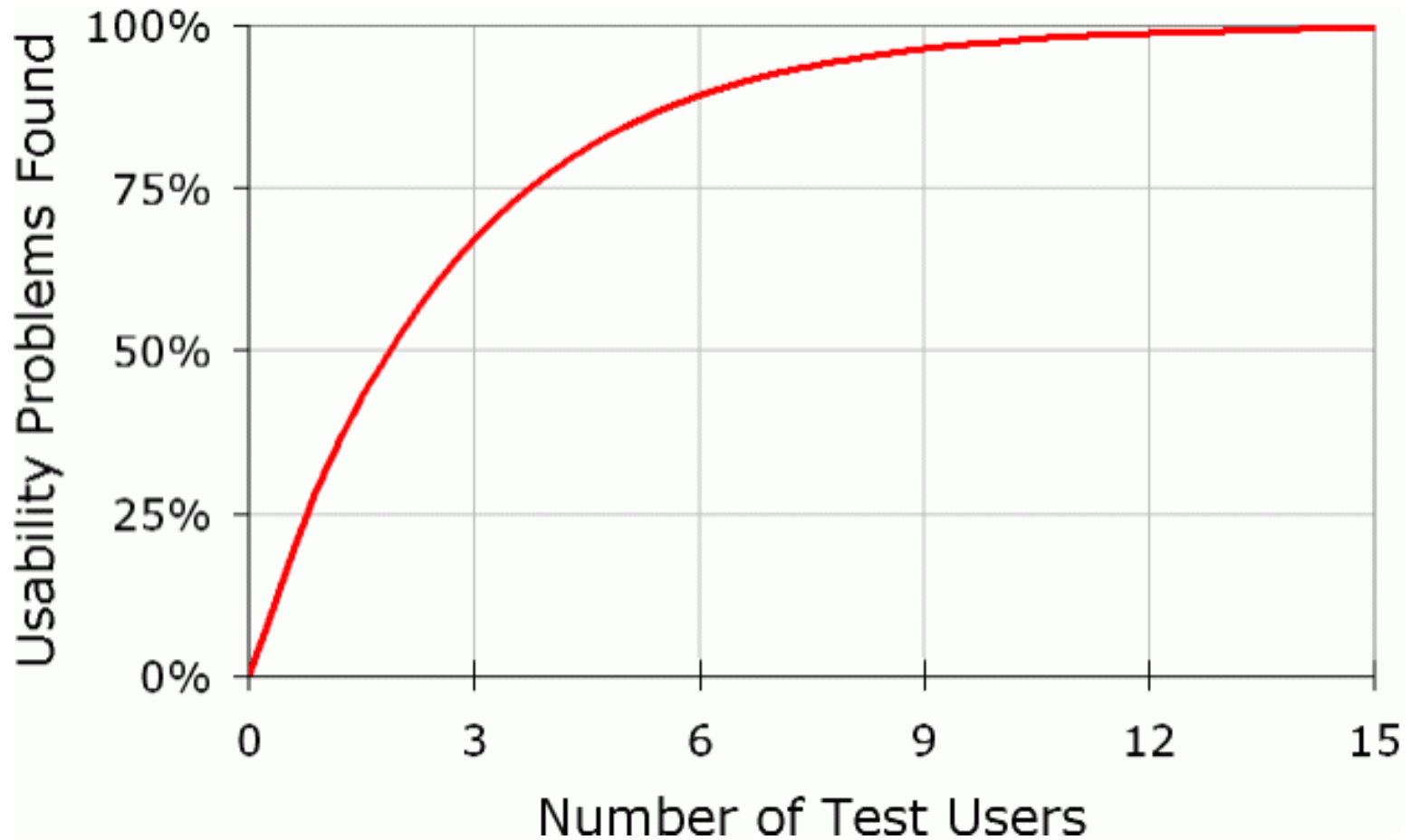
В каких средах разработки (IDE) Вы работали?



Какая Ваша основная среда разработки?



Респонденты (3)



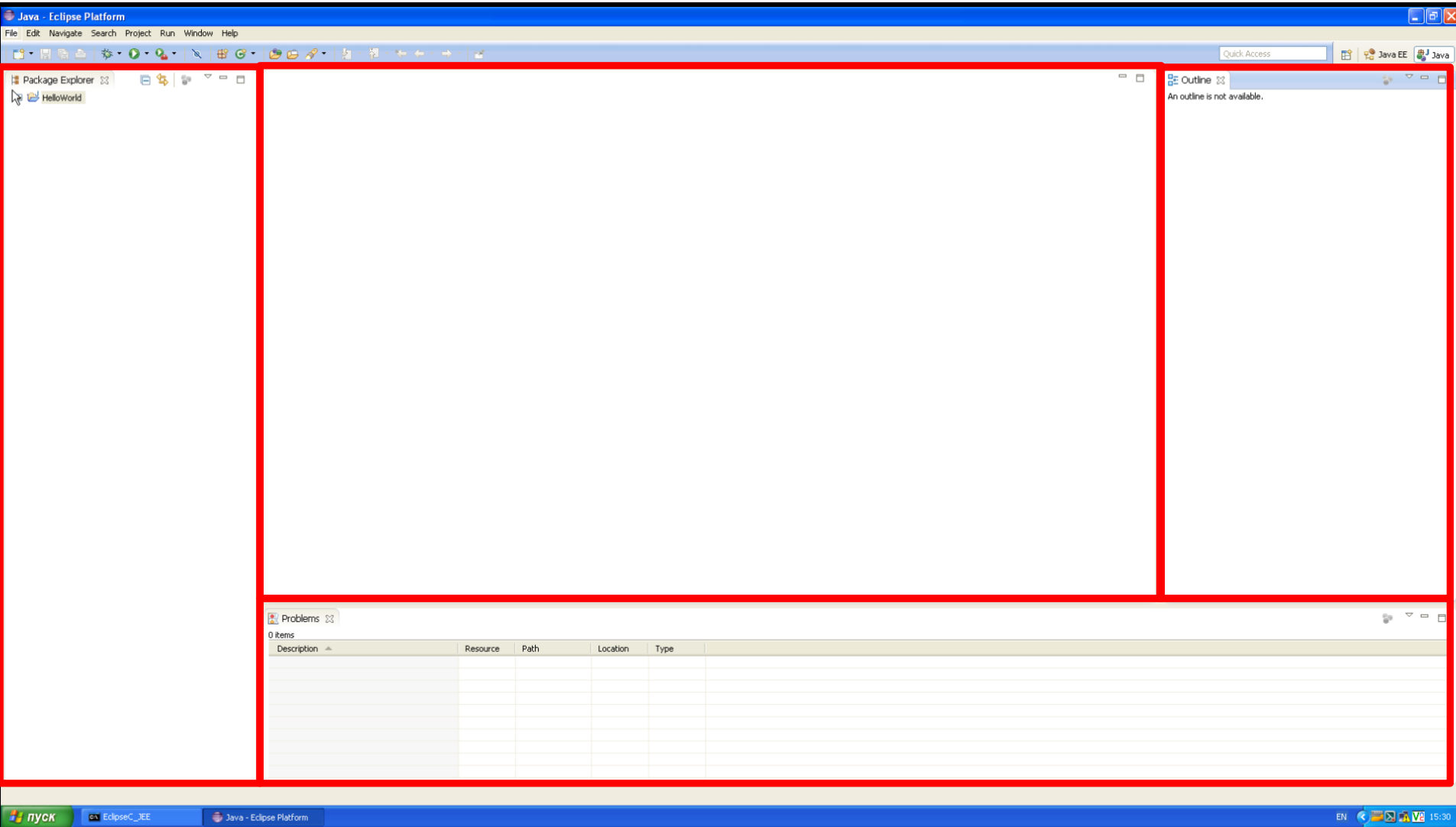
<http://www.useit.com/alertbox/20000319.html> - Usability Testing with 5 Users (Jakob Nielsen's Alertbox)

Сценарий тестирования

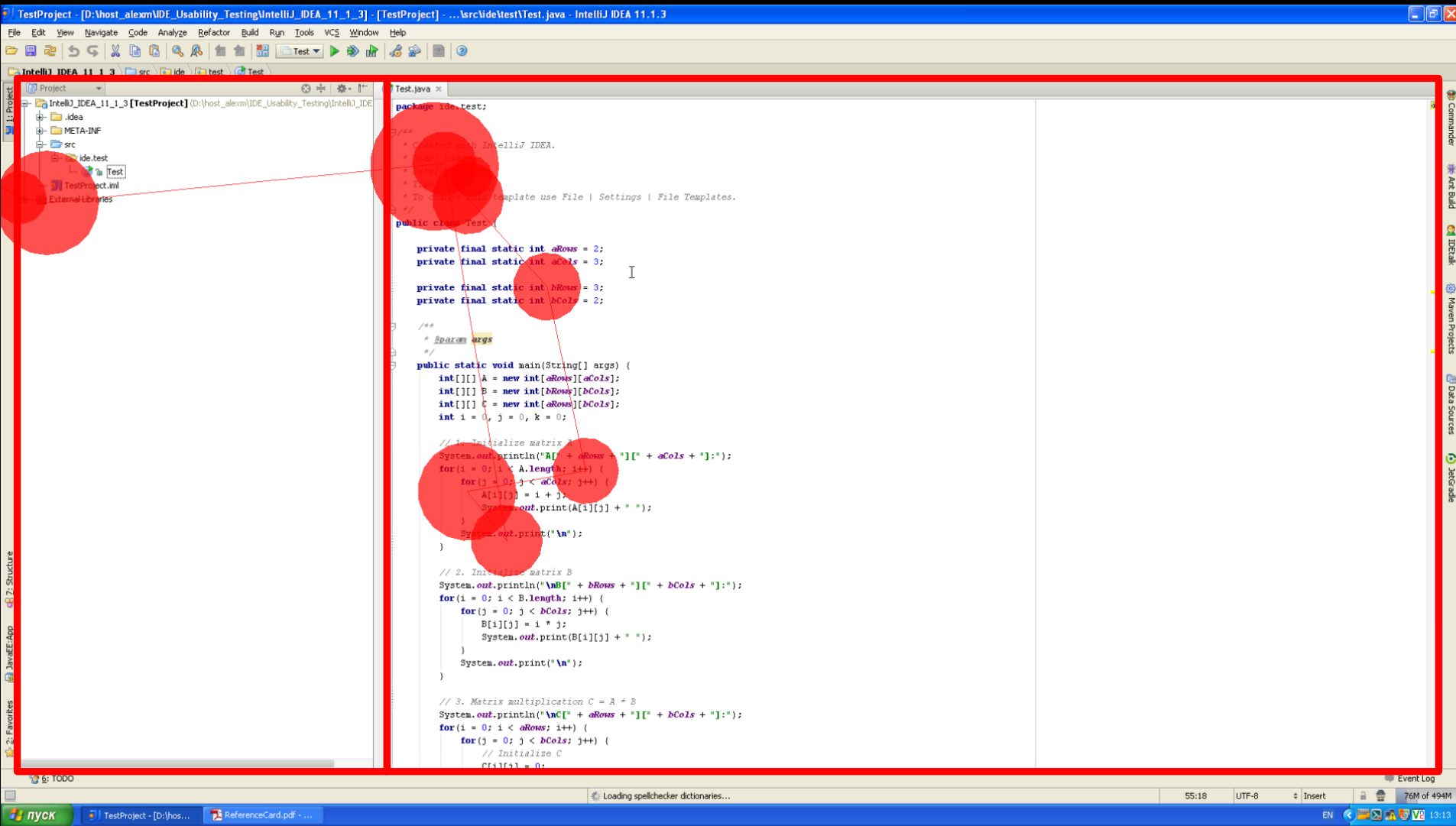
1. Свободный просмотр интерфейса
2. Настройка среды + Hello World
3. Отладка приложения
4. Рефакторинг кода

Результаты тестирования

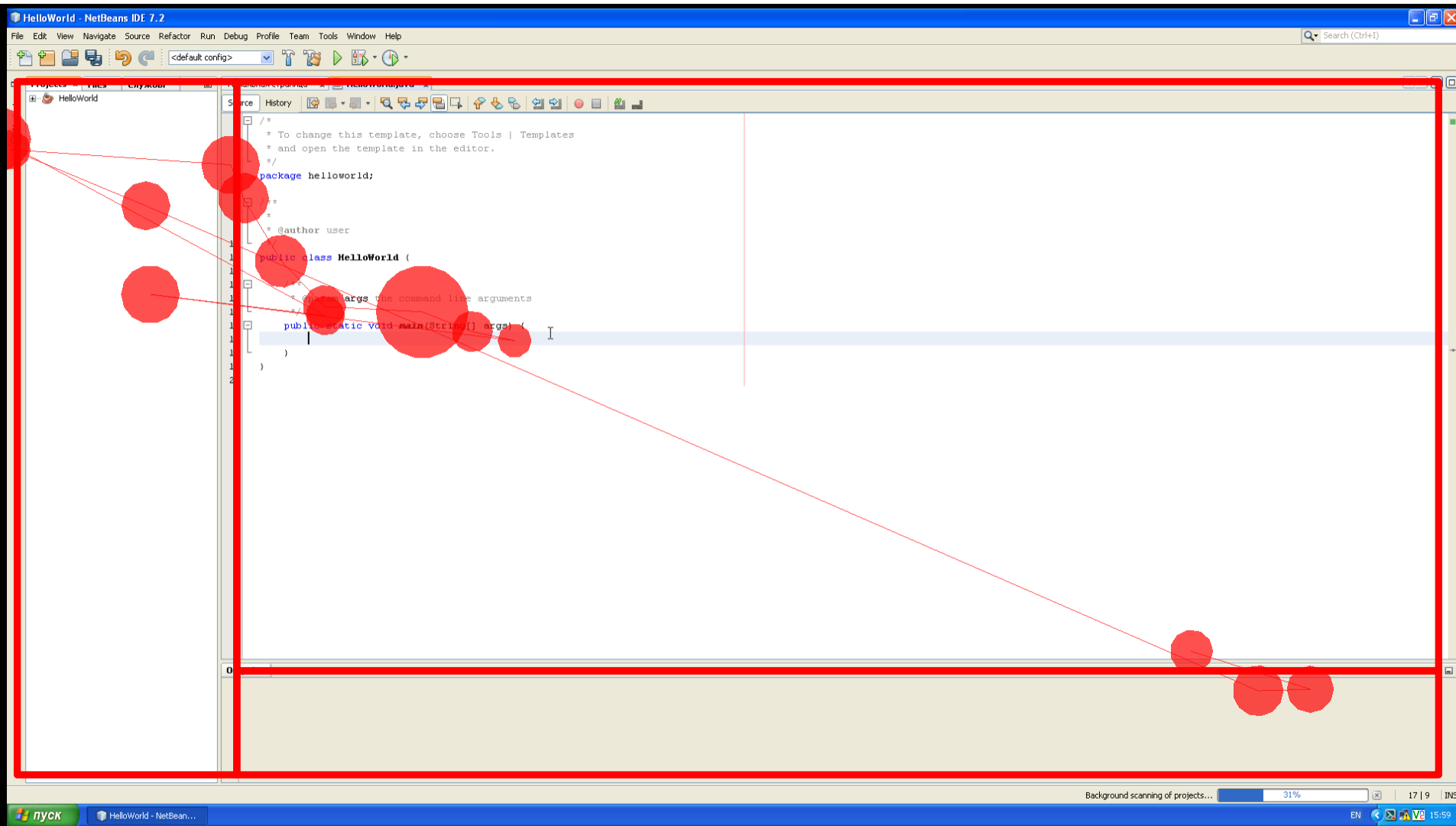
Компоновка Eclipse



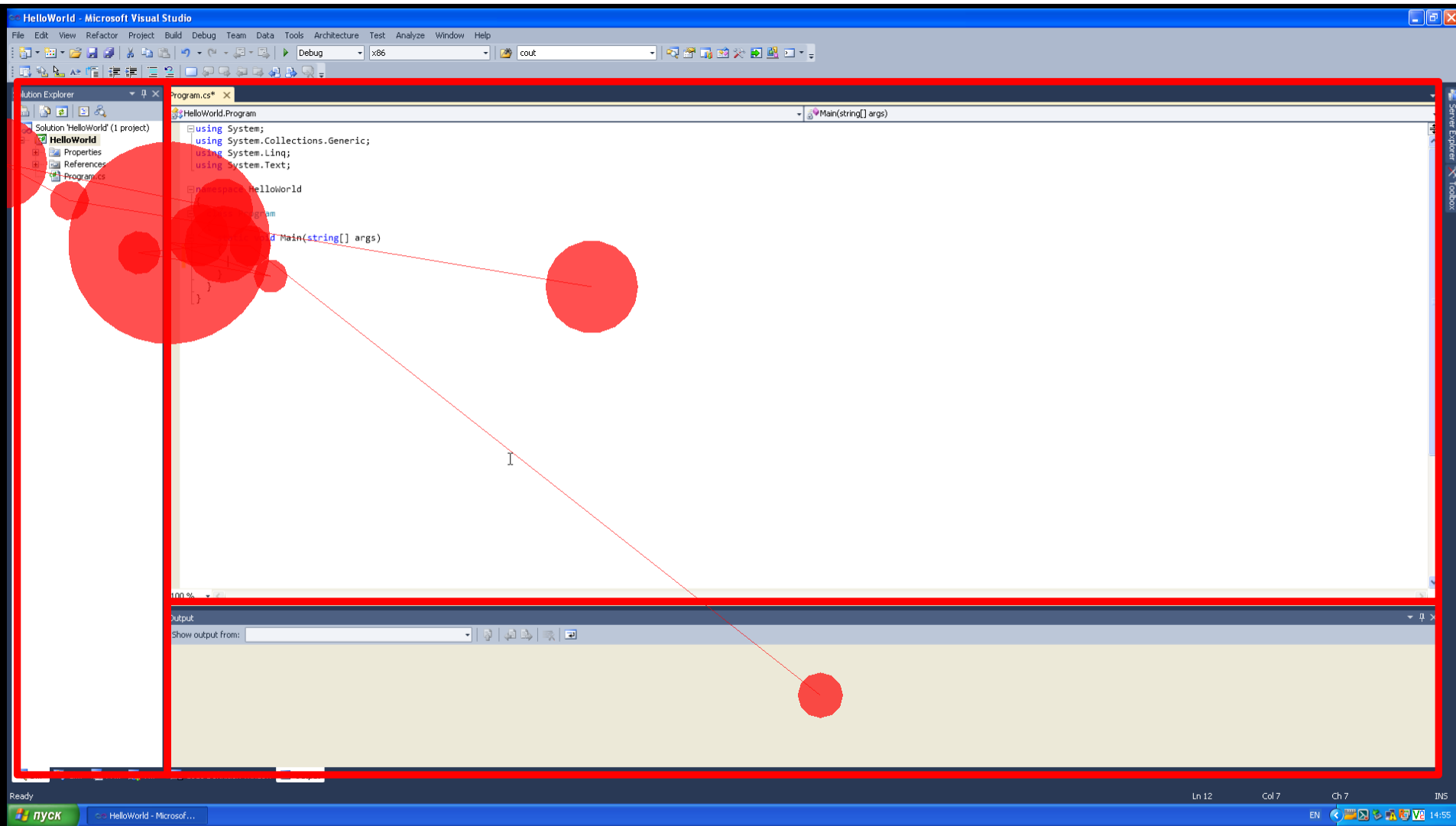
Компоновка IDEA



Компоновка NetBeans



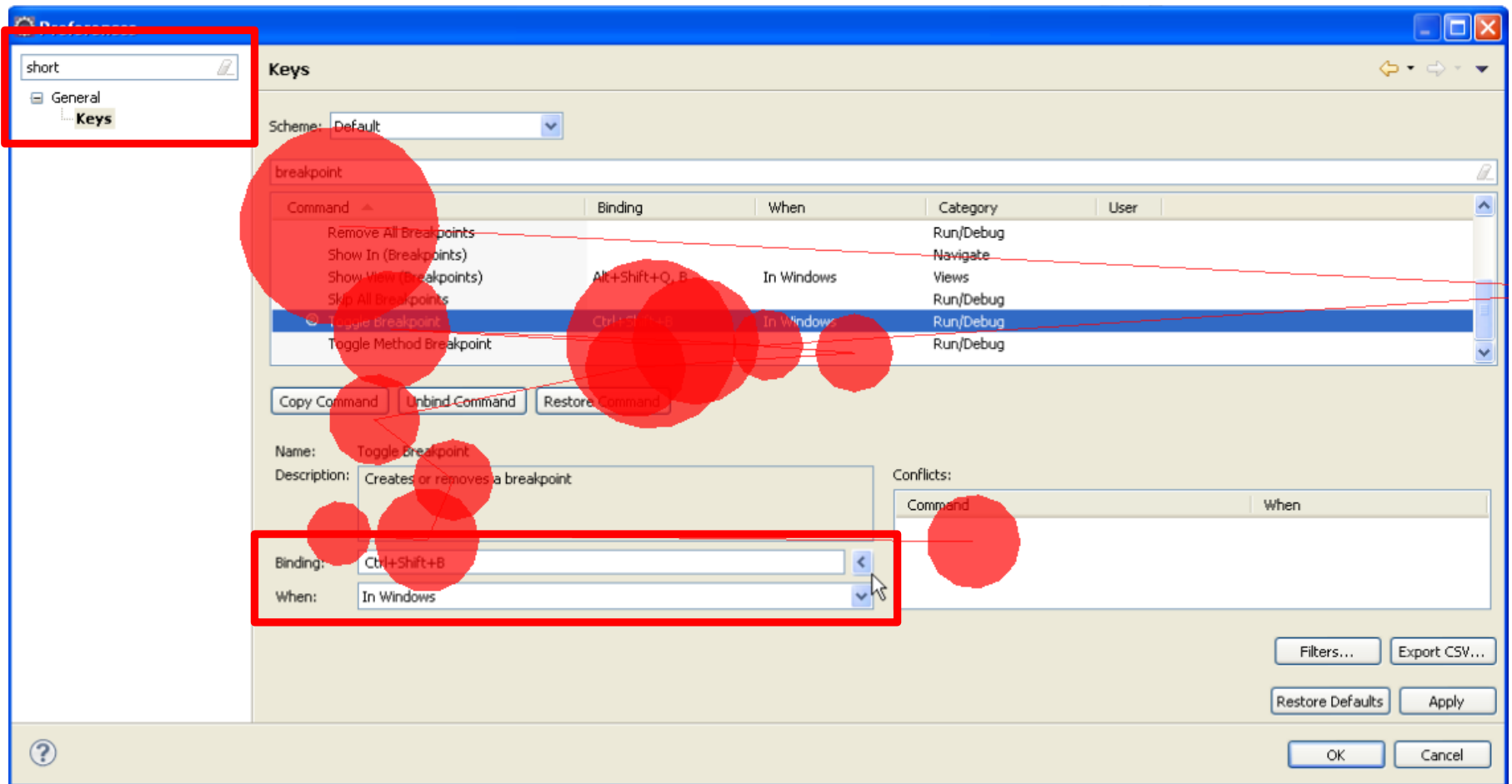
Компоновка VS



Оценка компоновки по умолчанию

IDE	Оценка
VS	4,20
IDEA	4,17
NetBeans	3,63
Eclipse	3,00

Настройки Eclipse



Настройки IDEA (1)

The screenshot shows the IntelliJ IDEA 11.1.3 interface with the Settings dialog box open. The dialog is titled "Settings" and has a search bar at the top. The left sidebar lists various settings categories, with "Schemas and DTDs" selected. The main area of the dialog is divided into two sections: "External Schemas and DTDs" and "Ignored Schemas and DTDs".

Red circles and lines highlight specific elements:

- A red circle highlights the search bar in the Settings dialog.
- A red circle highlights the "Schemas and DTDs" category in the left sidebar.
- A red circle highlights the "External Schemas and DTDs" section header.
- A red circle highlights the "Ignored Schemas and DTDs" section header.
- A red circle highlights the "Default HTML language level" section.
- A red circle highlights the "HTML 4" radio button.
- A red circle highlights the "Other doctype" radio button.
- A red circle highlights the "OK" button.
- A red circle highlights the "Cancel" button.
- A red circle highlights the "Apply" button.
- A red circle highlights the "Help" button.

The "External Schemas and DTDs" section contains a table with columns "URI" and "Location". The "Ignored Schemas and DTDs" section contains a list of URIs:

- <http://relaxng.org/ns/compatibility/annotations/1.0>
- <http://www.w3.org/1998/Math/MathML>
- <http://www.w3.org/2000/svg>
- <urn:idea:ext-plugin#extensions>

The "Default HTML language level" section has the following options:

- HTML 4 ("http://www.w3.org/TR/html4/loose.dtd")
- HTML 5
- Other doctype: _____

The bottom of the dialog has buttons for "OK", "Cancel", "Apply", and "Help".

Настройки IDEA (2)

The screenshot displays the IntelliJ IDEA 11.1.3 interface with the 'IntelliJ IDEA Default Keymap' dialog box open. The dialog is divided into three main sections: Editing, Usage Search, and Refactoring. Each section contains a list of keyboard shortcuts and their corresponding actions.

IntelliJ IDEA Default Keymap - Editing

Ctrl + Space	Basic code completion (the name of any class, method or variable)
Ctrl + Shift + Space	Smart code completion (filters the list of methods and variables by expected type)
Ctrl + Alt + Space	Class name completion (the name of any project class independently of current imports)
Ctrl + Shift + Enter	Complete statement
Ctrl + P	Parameter info (within method call arguments)
Ctrl + O	Quick documentation lookup
Shift + F1	External Doc
Ctrl + mouse over code	Brief Info
Ctrl + F1	Show descriptions of error or warning at caret
Alt + Insert	Generate code... (Getters, Setters, Constructors, hashCode/equals, toString)
Ctrl + O	Override methods
Ctrl + I	Implement methods
Ctrl + Alt + T	Surround with... (if, else, try, catch, for, synchronized, etc.)
Ctrl + /	Comment/uncomment with line comment
Ctrl + Shift + /	Comment/uncomment with block comment
Ctrl + W	Select successively increasing code blocks
Ctrl + Shift + W	Decrease current selection to previous state
Alt + O	Contact info
Alt + Enter	Show intention actions and quick-fixes
Ctrl + Alt + L	Reformat code
Ctrl + Alt + O	Optimize imports
Ctrl + Alt + I	Auto-indent (lines)
Tab / Shift + Tab	indent/unindent selected lines
Ctrl + X or Shift + Delete	Cut current line or selected block to clipboard
Ctrl + C or Ctrl + Insert	Copy current line or selected block to clipboard
Ctrl + V or Shift + Insert	Paste from clipboard
Ctrl + Shift + V	Paste from recent buffers...
Ctrl + D	Duplicate current line or selected block
Ctrl + Y	Delete line at caret
Ctrl + Shift + J	Smart line join
Ctrl + Enter	Smart line split
Shift + Enter	Start new line
Ctrl + Shift + U	Toggle case for word at caret or selected block
Ctrl + Shift + / /	Select till code block end/start
Ctrl + Delete	Delete to word end
Ctrl + Backspace	Delete to word start
Ctrl + NumPad+/-	Expand/collapse code block
Ctrl + Shift + NumPad+	Expand all
Ctrl + Shift + NumPad-	Collapse all
Ctrl + F4	Close active editor tab

IntelliJ IDEA Default Keymap - Usage Search

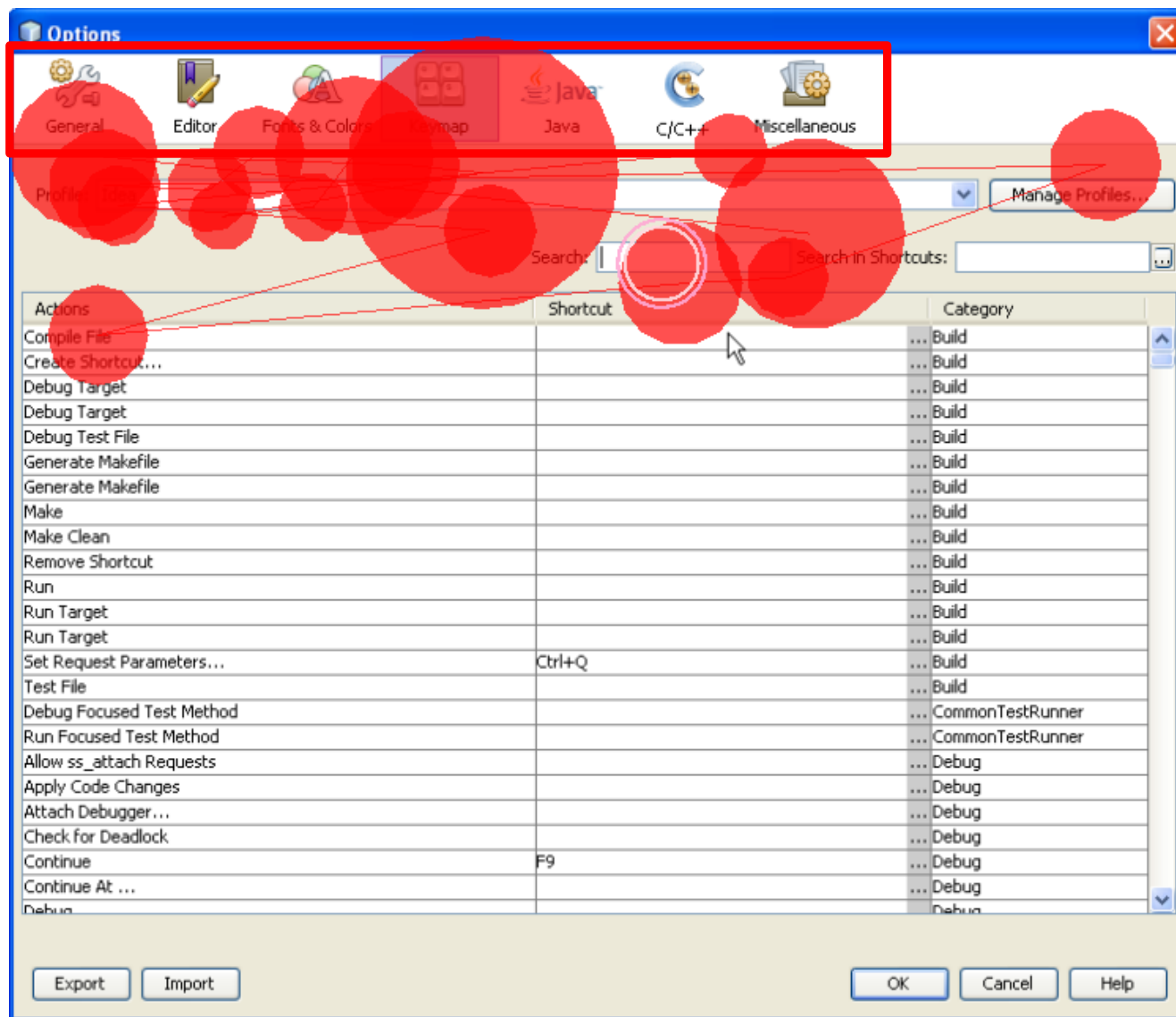
Alt + F7 / Ctrl + F7	Find usages / Find usages in file
Ctrl + Shift + F7	Highlight usages in file
Ctrl + Alt + F7	Show usages

IntelliJ IDEA Default Keymap - Refactoring

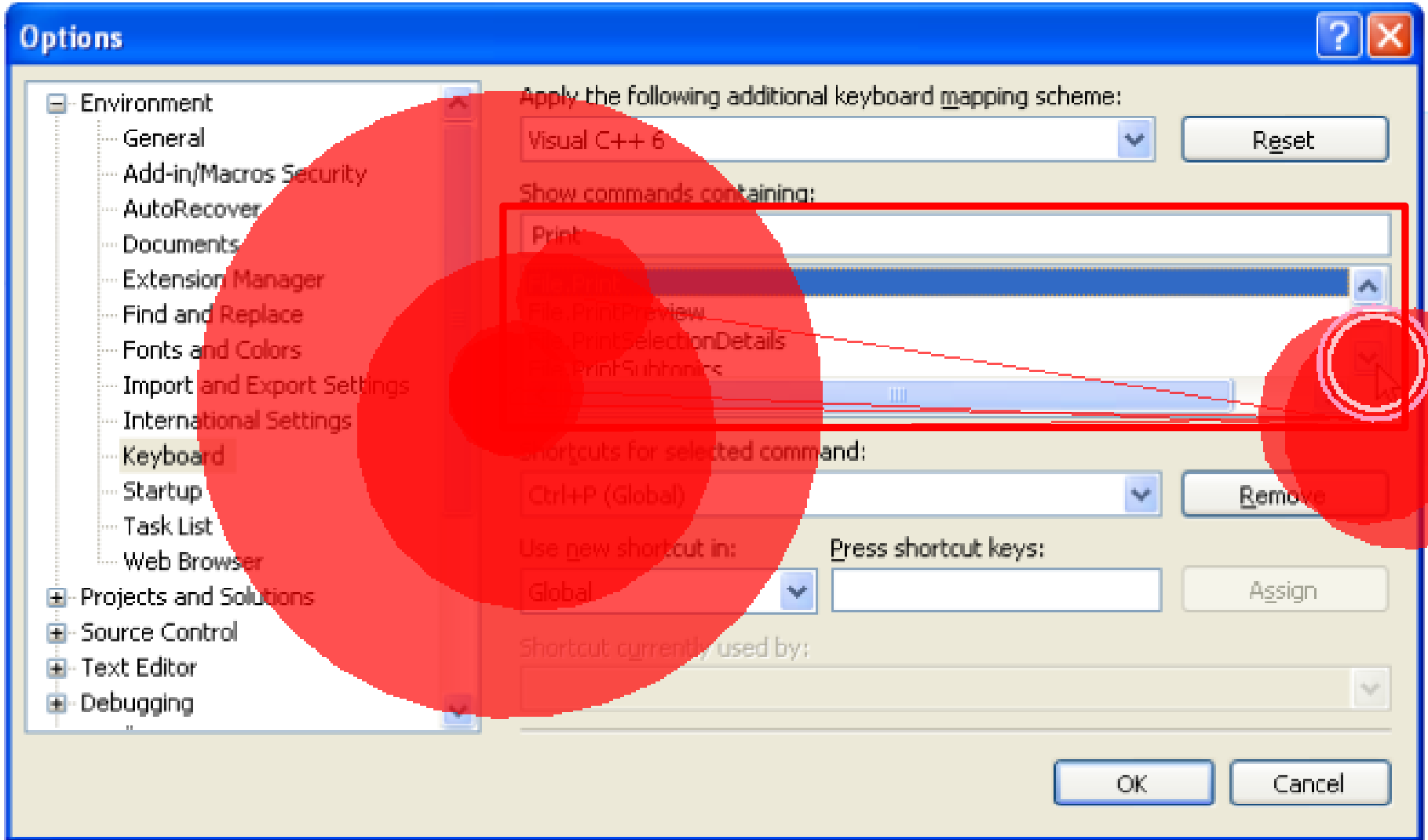
F8	Copy
F8	Move
Alt + Delete	Safe Delete
Shift + F6	Rename
Ctrl + F6	Change Signature
Ctrl + Alt + N	Inline
Ctrl + Alt + M	Extract Method
Ctrl + Alt + V	Introduce Variable
Ctrl + Alt + F	Introduce Field
Ctrl + Alt + C	Introduce Constant
Ctrl + Alt + P	Introduce Parameter

The 'Find Action' dialog is also visible in the bottom right corner, showing a search for 'Find Action (Ctrl+Shift+A)' with a list of actions including 'Copy (Ctrl+C)', 'Revert Changes', 'Copy Query (Ctrl+Alt+Shift+C)', and 'Copy Paths (Ctrl+Shift+C)'.

Настройки NetBeans



Настройки Visual Studio



Эффективность настроек среды

IDE	Оценка
IDEA	4,00
NetBeans	3,75
VS	3,40
Eclipse	3,17

Eclipse Debugger

The screenshot shows the Eclipse IDE in a debug state. The main editor displays the source code of `Test.java` with a breakpoint at line 25. The Variables view shows the state of variables `A`, `C`, and `A[0][0]`. The Console view shows the output of the program. Red circles and lines highlight the connection between the code, the variable values, and the console output.

```
10
11 /**
12  * @param args
13  */
14 public static void main(String[] args) {
15     int[][] A = new int[aRows][aCols];
16     int[][] B = new int[bRows][bCols];
17     int[][] C = new int[aRows][bCols];
18     int i = 0, j = 0, k = 0;
19
20     // 1. Initialize Matrix A
21     System.out.println("A[" + aRows + "][" + aCols + "]:");
22     for (i = 0; i < A.length; i++) {
23         for (j = 0; j < A[0].length; j++) {
24             A[i][j] = i + j;
25             System.out.print(A[i][j] + " ");
26         }
27         System.out.println();
28     }
29 }
```

Name	Value
"A[0][0]"	0
"C"	(id=19)
"A"	(id=17)
"A[0][0]"	2

```
Test
├── aRows : int
├── aCols : int
├── bRows : int
├── bCols : int
└── main(String[]) : void
```

```
Test [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (08.10.2012 11:17:53)
A[2][3]:
0 1
```


IDEA Debugger

The screenshot displays the IntelliJ IDEA 11.1.3 IDE interface. The main editor shows a Java file named `Test.java` with the following code:

```
// 2. Initialize matrix B
System.out.println("\nB[" + bRows + "][" + bCols + "]:");
for(i = 0; i < B.length; i++) {
    for(j = 0; j < bCols; j++) {
        B[i][j] = i * j;
        System.out.print(B[i][j] + " ");
    }
    System.out.print("\n");
}

// 3. Matrix multiplication C = A * B
System.out.println("\nC[" + aRows + "][" + bCols + "]:");
for(i = 0; i < aRows; i++) {
    for(j = 0; j < bCols; j++) {
        // Initialize C
        C[i][j] = 0;
        for(k = 0; k < B.length; k++) {
            C[i][j] = C[i][j] + A[i][k] * B[k][j];
        }
        System.out.print(C[i][j] + " ");
    }
    System.out.print("\n");
}
```

The debugger is active, with a breakpoint set on the line `C[i][j] = C[i][j] + A[i][k] * B[k][j];`. The console shows the execution flow, and the Variables window displays the state of variables:

- `i`: 1
- `j`: 1
- `k`: 1
- `B`: `{int[2]@74}`
- `A`: `{int[2]@74}`
- `C`: `{int[2]@69}`
- `C[0][0]`: 0

Red circles and lines highlight the variables `i`, `j`, and `k` in the code and their corresponding values in the Variables window.

NetBeans Debugger

The screenshot shows the NetBeans IDE 7.2 interface. The main editor displays the source code of a Java class named `Test`. The code includes static variables `aRows`, `aCols`, `bRows`, and `bCols`, and a `main` method that initializes arrays `A`, `B`, and `C`, and prints the matrix `A`. A red horizontal bar highlights the line `int[] A = new int[aRows][aCols];` at line 22, and a green horizontal bar highlights the line `int[] B = new int[bRows][bCols];` at line 23. A red circle highlights the line `int[] C = new int[aRows][bCols];` at line 24. A red circle highlights the line `int i = 0, j = 0, k = 0;` at line 25, which has a red vertical bar indicating a breakpoint. The 'Evaluate Code' window is open, showing the state of variables `A`, `B`, `C`, and `args`. The 'Type' column in the 'Evaluate Code' window is highlighted with a red box. The 'Type' column contains the following entries: `int[][]`, `int[][]`, `int`, `int[][]`, and `String[]`. The 'Output' and 'Evaluation Result' tabs are also visible. The status bar at the bottom indicates 'Thread main stopped at Test.java:25' and 'Test (debug) running... 25 | 1 | INS 16:03'.

```
10  /*
11  public class Test {
12
13      private final static int aRows = 2;
14      private final static int aCols = 3;
15      private final static int bRows = 3;
16      private final static int bCols = 2;
17
18      /**
19       * @param args
20       */
21      public static void main(String[] args) {
22          int[] A = new int[aRows][aCols];
23          int[] B = new int[bRows][bCols];
24          int[] C = new int[aRows][bCols];
25          int i = 0, j = 0, k = 0;
26
27          // Print matrix A
28          System.out.println("A[" + aRows + "][" + aCols + "]:");
29          for (i = 0; i < A.length; i++) {
30              for (j = 0; j < aCols; j++) {
31                  A[i][j] = i + j;
32                  System.out.print(A[i][j] + " ");
33              }
34              System.out.print("\n");
35          }
36      }
37  }
```

Variable	Value	Type
A	#45(length=0)	int[][]
B	3	int[][]
C	#49(length=2)	int
Static		int[][]
args	#43(length=0)	String[]

Visual Studio Debugger

The screenshot displays the Visual Studio IDE during a debugging session. The main window shows the source code of `Program.cs` with the following content:

```
// 2. Initialize matrix B
Console.WriteLine("\nB[" + bRows + "][" + bCols + "]:");
for (i = 0; i < B.GetLength(0); i++)
{
    for (j = 0; j < bCols; j++)
    {
        B[i, j] = i * j;
        Console.Write(B[i, j] + " ");
    }
    Console.WriteLine("\n");
}

// 3. Matrix multiplication C = A * B
Console.WriteLine("\nC[" + aRows + "][" + bCols + "]:");
for (i = 0; i < aRows; i++)
{
    for (j = 0; j < bCols; j++)
    {
        // Initialize
        C[i, j] = 0;
        for (k = 0; k < B.GetLength(0); k++)
        {
            C[i, j] += A[i, k] * B[k, j];
        }
        Console.WriteLine(C[i, j] + " ");
    }
    Console.WriteLine("\n");
}
}
```

The debugger is paused at line 54. The **Call Stack** window shows the following stack:

Name	Lang
Test.exe!Test.Program.Main(string[] args) Line 54	C#
[Native to Managed Transition]	
[Managed to Native Transition]	
microsoft.dll!System.AppDomain.ExecuteAssembly(string assemblyFile, System.Security.Policy.Evidence assemblySecurity, string[] args) + 0x6d bytes	
Microsoft.VisualStudio.HostingProcess.Utilities.dll!Microsoft.VisualStudio.HostingProcess.HostProc.RunUsersAssembly() + 0x2a bytes	
microsoft.dll!System.Threading.ThreadHelper.ThreadStart_Context(object state) + 0x63 bytes	
microsoft.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object state) + 0x5d bytes	
microsoft.dll!System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object state, bool suppressFlowContext) + 0x5d bytes	

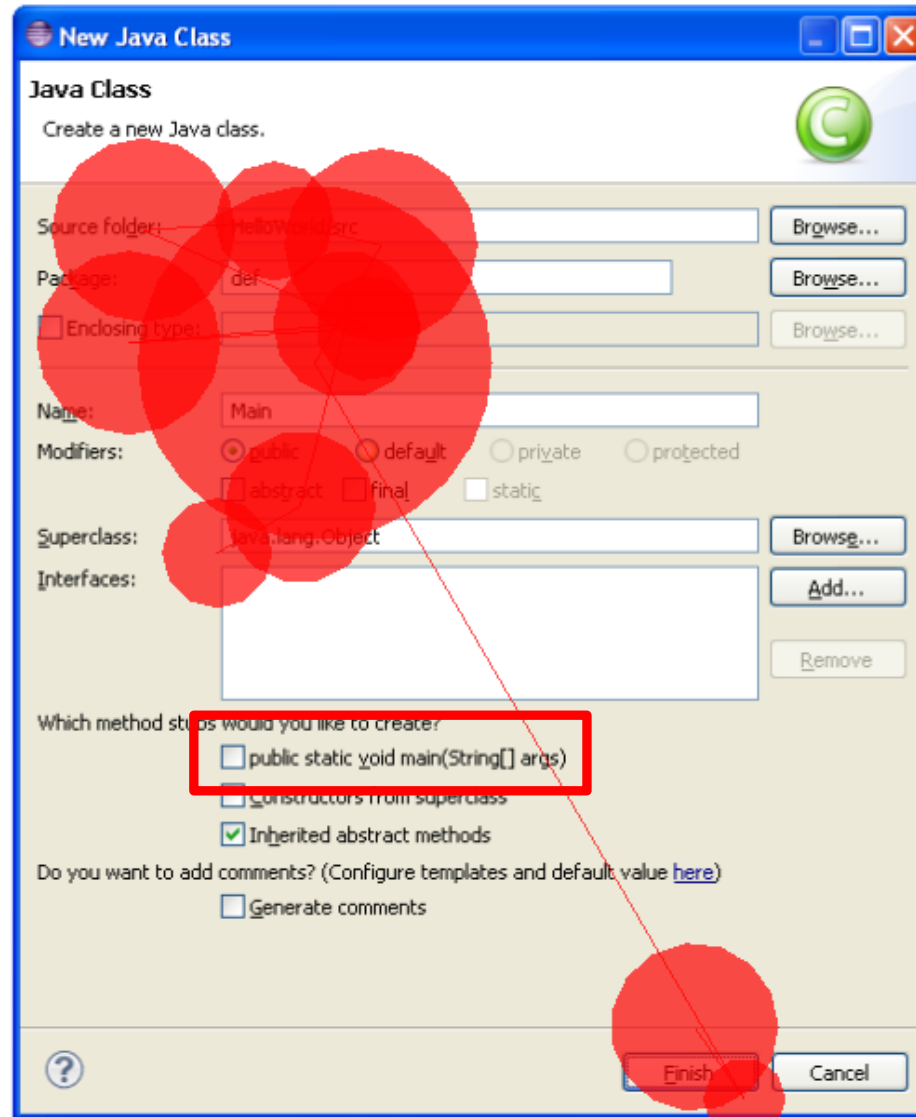
The **Locals** window shows the following variables:

Name	Value	Type
A	{int[2, 3]}	int[,]
A[i, k]	1	int
B	{int[3, 2]}	int[,]
B[k, i]	0	int
C	{int[2, 2]}	int[,]
C[i, j]	0	int
i	0	int
j	0	int

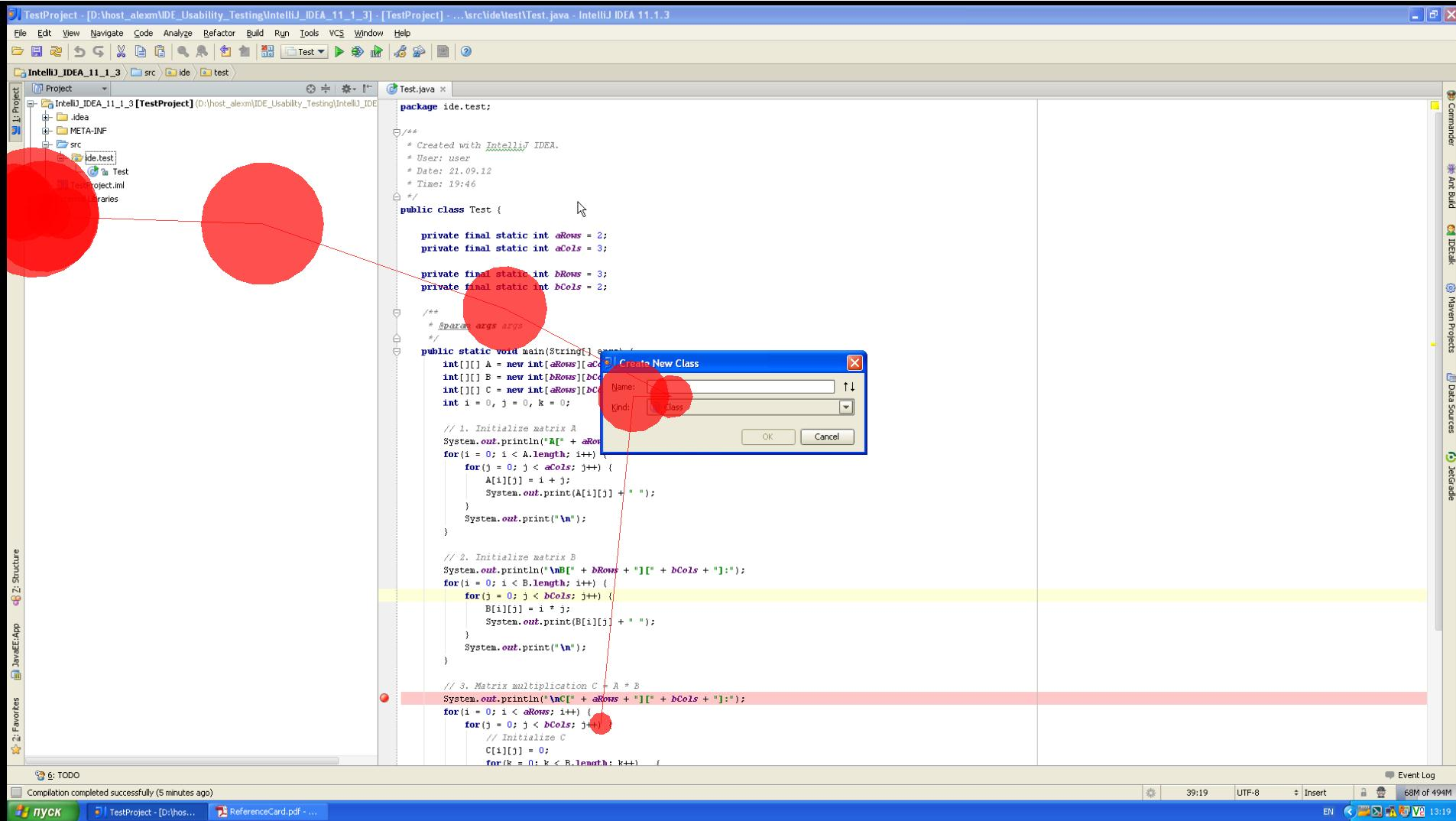
Наглядность отладчика

IDE	Оценка
IDEA	4,33
VS	4,20
Eclipse	4,00
NetBeans	3,75

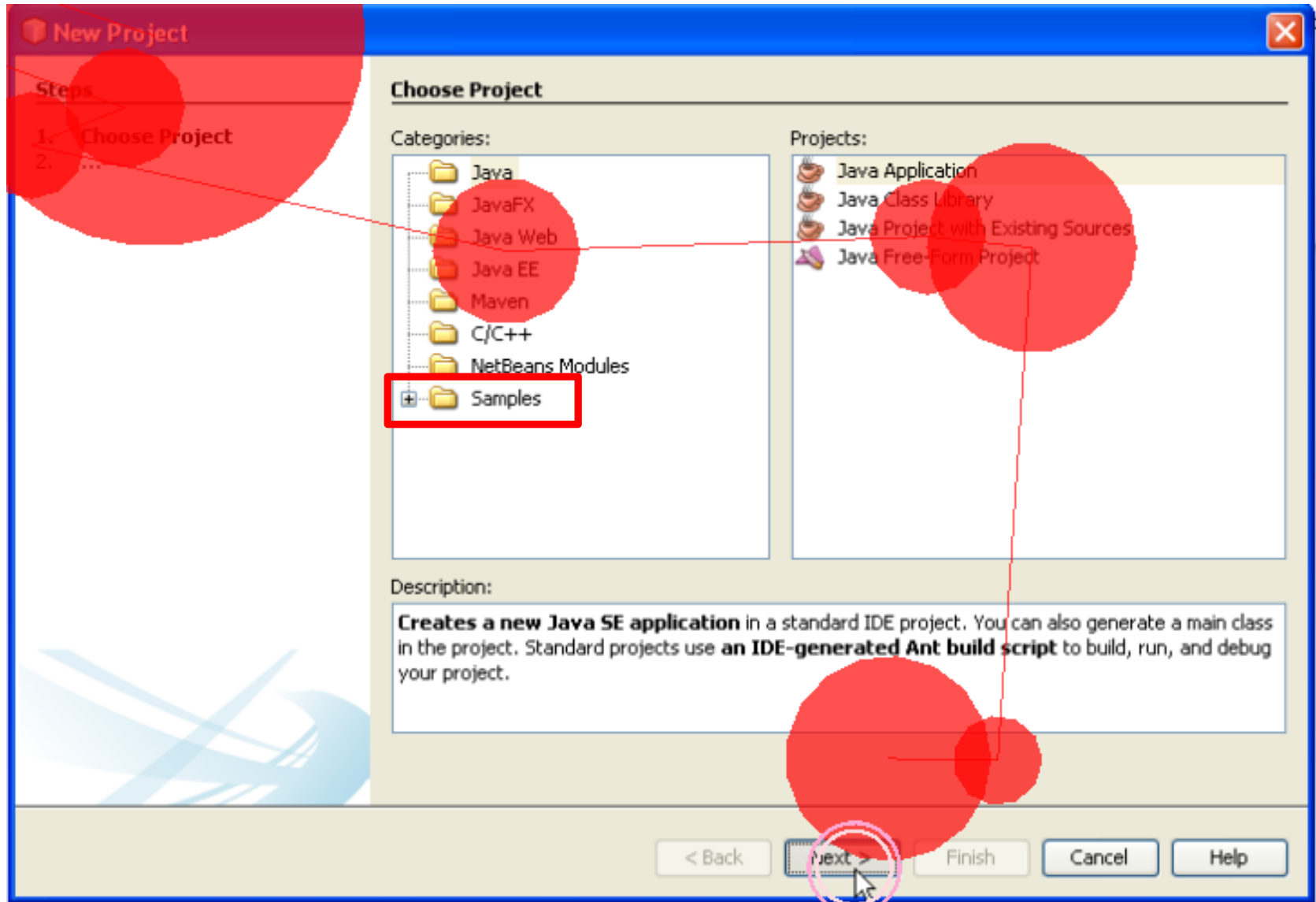
Eclipse New Java Class



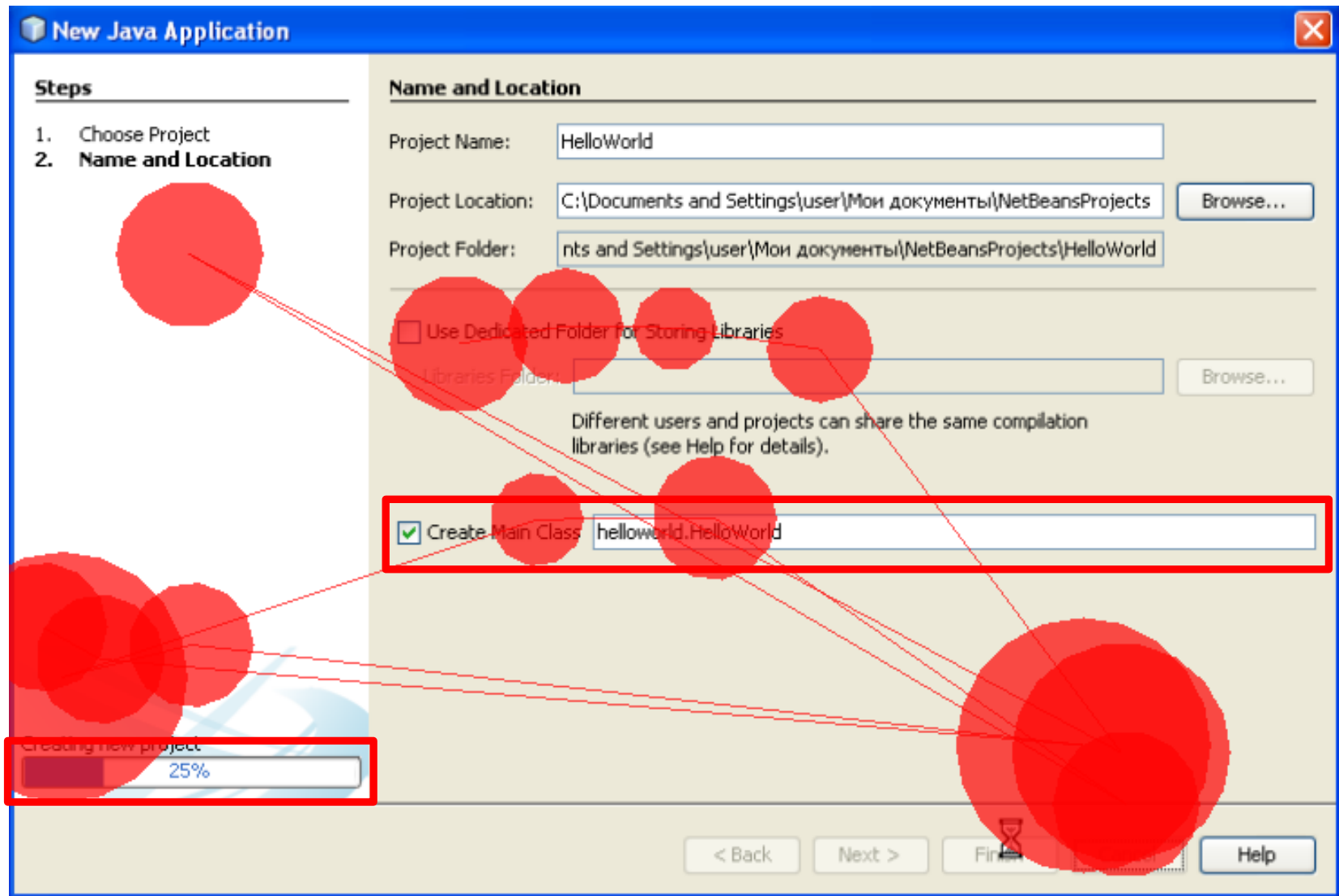
IDEA New Class



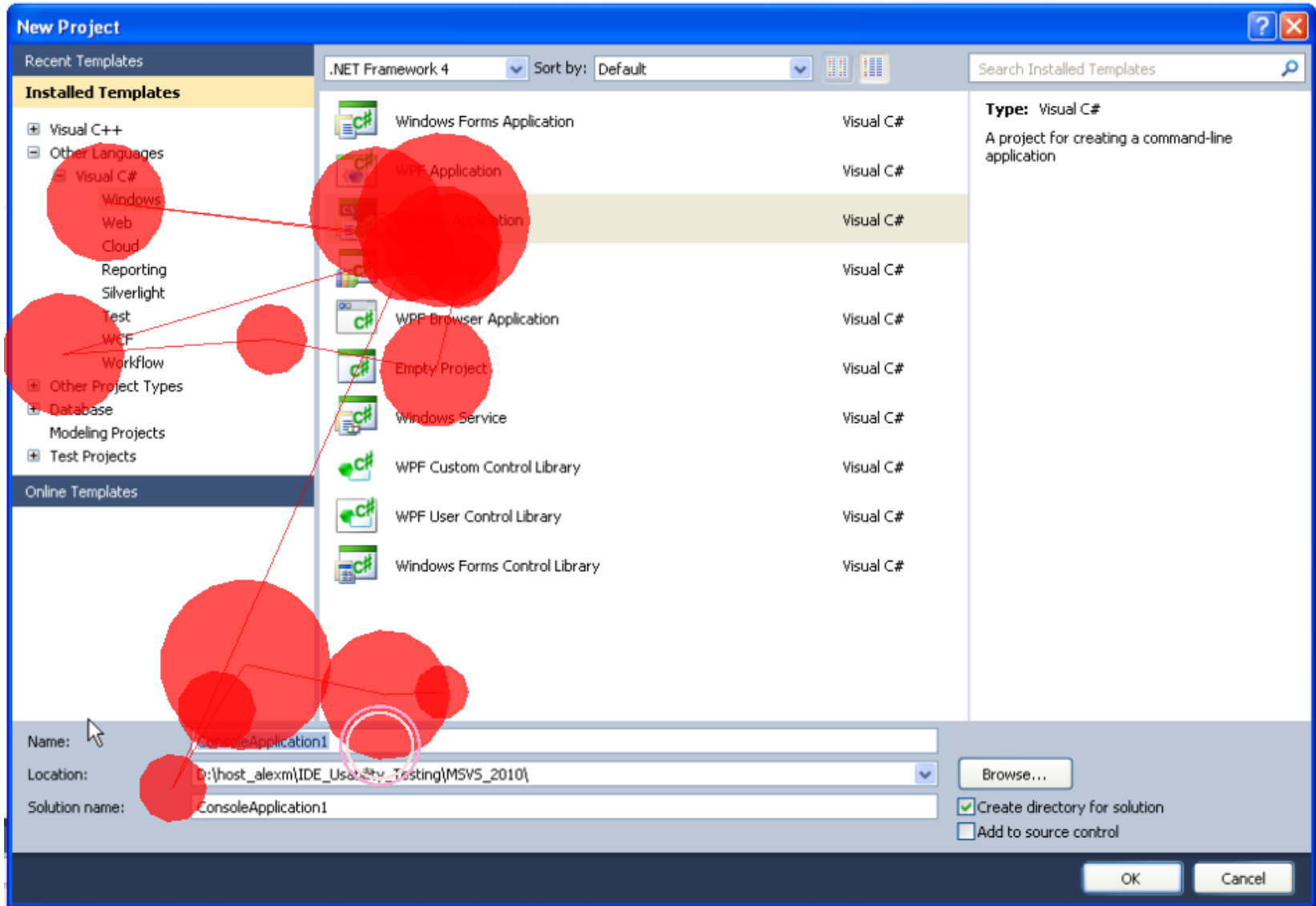
NetBeans New Project



NetBeans New Java Application



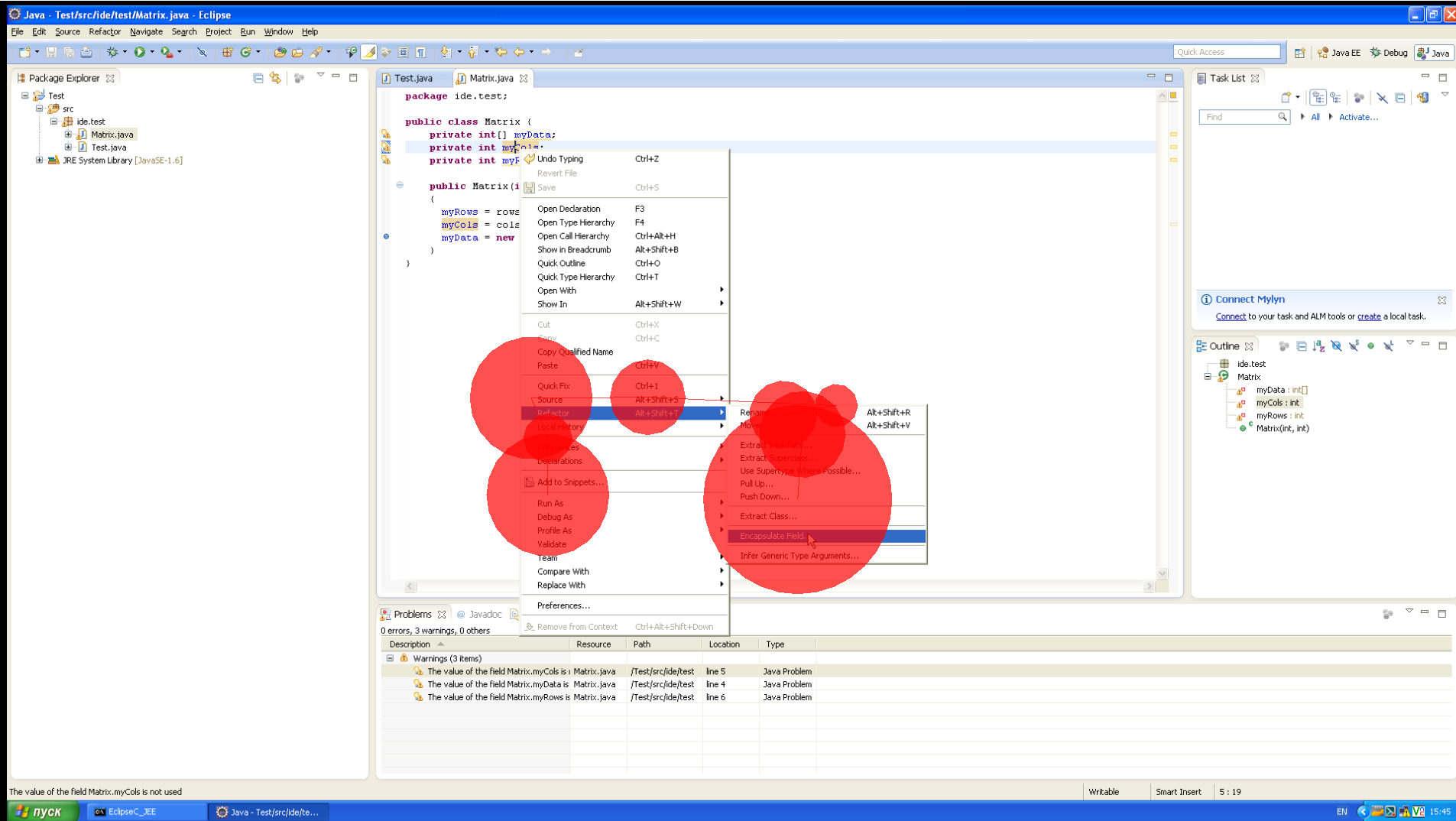
Visual Studio New Project



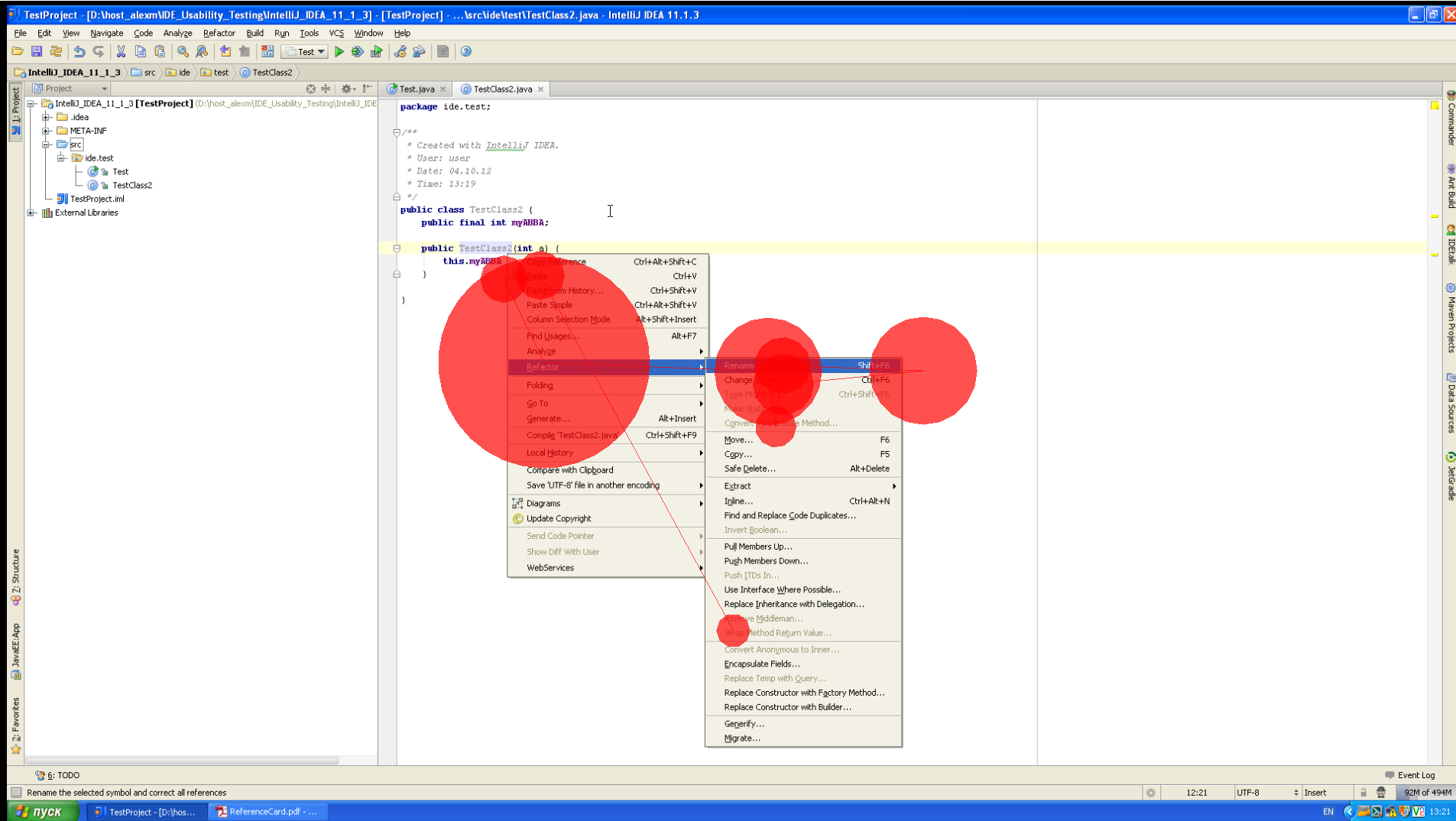
Простота создания приложения и класса

IDE	Приложение	Класс
Eclipse	3,00	3,50
IDEA	4,17	4,50
NetBeans	4,50	4,00
VS	4,00	3,80

Контекстное меню Eclipse



Контекстное меню IDEA



Eclipse QuickFix

The screenshot shows the Eclipse IDE interface. The main editor displays the following code:

```
package ide.test;

public class Matrix {
    private int my;
    private int my;
    public Matrix(
    {
        myRows = row;
        myCols = col;
        myData = new
    )
}
```

A red box highlights a QuickFix dialog for the warning "The value of the field Matrix.myData is not used". The dialog contains the following options:

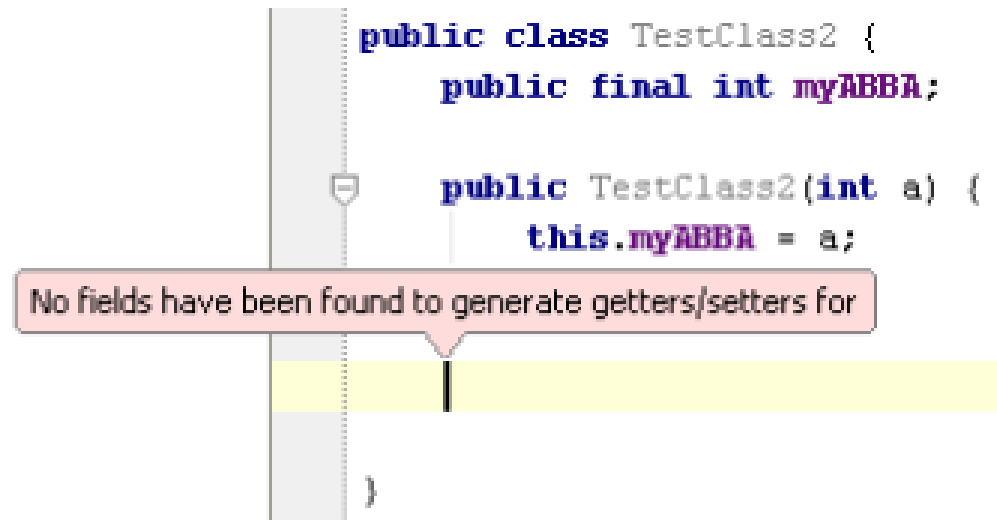
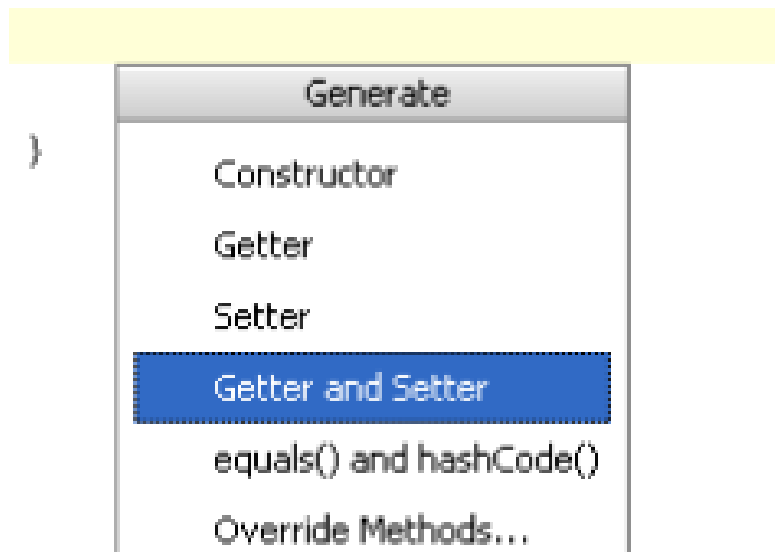
- Remove 'myData', keep assignments with side effects
- Create getter and setter for 'myData'...
- Rename in file (Ctrl+Z, R)
- Rename in workspace (Alt+Shift+R)
- Add @SuppressWarnings('unused') to 'myData'
- Starts the 'Encapsulate field' refactoring to create getter and setters

The Problems view at the bottom shows three warnings:

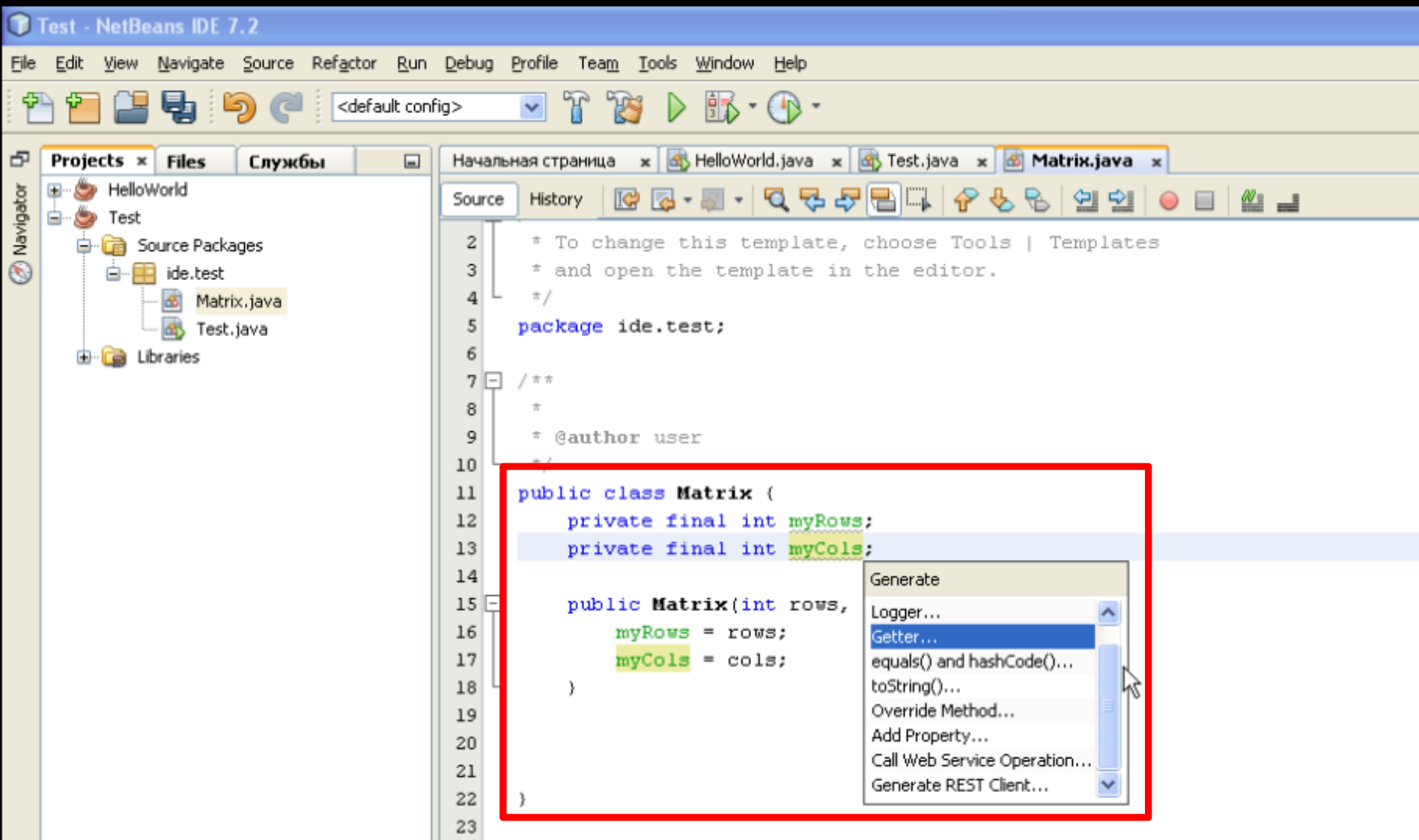
Description	Resource	Path	Location	Type
The value of the field Matrix.myCols is not used	Matrix.java	/Test/src/ide/test	line 5	Java Problem
The value of the field Matrix.myData is not used	Matrix.java	/Test/src/ide/test	line 4	Java Problem
The value of the field Matrix.myRows is not used	Matrix.java	/Test/src/ide/test	line 6	Java Problem

IDEA Auto Generate Getter

```
public class TestClass2 {  
    public final int myABBA;  
  
    public TestClass2(int a) {  
        this.myABBA = a;  
    }  
}
```



NetBeans Auto Generate Getter























IDEA AutoComplete

```
public class TestClass {  
    |  
    }  
    geti    Inserts singleton method getInstance  
    psf    public static final  
    psfi    public static final int  
    psfs    public static final String  
    psvm    main() method declaration  
    $t    String
```

```
TestClass2 test = new TestClass2(10);  
System.out.println("test = " + test);  
System.out.println("test. = " + test.);
```

```
// 1. Initialize matrix A  
System.out.println("A[" + aRows  
for(i = 0; i < A.length; i++) {  
    for(j = 0; j < aCols; j++) {  
        A[i][j] = i + j;  
        System.out.print(A[i][j]  
    }  
    System.out.print("\n");  
}
```

		toString ()	String
		getABBA ()	int
		myABBA	int
		equals (Object obj)	boolean
		getClass ()	Class<?>
		hashCode ()	int
		notify ()	void
		notifyAll ()	void
		wait ()	void
		wait (long timeout)	void

NetBeans AutoComplete

The screenshot shows the NetBeans IDE with a Java source file on the left and a tooltip on the right. The source file contains a class `Test` with several methods and a static method. The tooltip displays the signature and documentation for `java.io.PrintStream.append(CharSequence csq)`. Below the tooltip, an autocomplete list is visible, showing various methods of `PrintStream` and their return types.

```
7 /**
8  *
9  * @author user
10 */
11 public class Test
12
13     private final
14     private final
15     private final
16     private final
17
18     /**
19      * @param arg
20      */
21     public static
22
23     Matrix m
24     System.out.
25     int[][]
26     int[][]
27     int[][]
28     int i =
29
30     // 1. In
```

java.io.PrintStream

public **PrintStream** **append**(CharSequence csq)

Appends the specified character sequence to this output stream.

An invocation of this method of the form `out.append(csq)` behaves in exactly the same way as the invocation

`out.print(csq.toString())`

Depending on the specification of `toString` for the character sequence `csq`, the entire sequence may not be appended. For instance, invoking then `toString` method of a character buffer will return a subsequence whose content depends upon the buffer's position and limit.

Parameters:

- **append**(CharSequence csq) PrintStream
- **append**(char c) PrintStream
- **append**(CharSequence csq, int start, int end) PrintStream
- **checkError**() boolean
- **close**() void
- **equals**(Object obj) boolean
- **flush**() void
- **format**(String format, Object... args) PrintStream
- **format**(Locale l, String format, Object... args) PrintStream
- **getClass**() Class<?>
- **hashCode**() int
- **notify**() void
- **notifyAll**() void
- **print**(Object obj) void
- **print**(String s) void
- **print**(boolean b) void
- **print**(char c) void

Output

Test (debug) x Debugger C

debug:

A[2][3]:

```
0 1 2
1 2 3
```

B[3][2]:

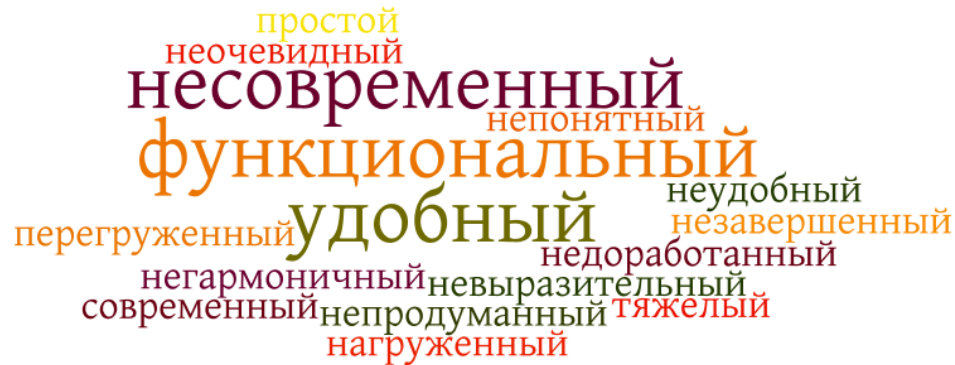
```
0 0
0 1
0 2
```

Средняя оценка параметров

Параметр/IDE	Eclipse	IDEA	NetBeans	VS
Эффективность AutoComplete	3,33	4,67	3,63	3,40
Эффективность Auto Generate	3,17	4,50	3,25	2,40
Эффективность рефакторинга	3,00	4,83	3,63	2,40
Наглядность построения Build	3,00	4,33	3,75	4,20
Быстрота построения Build и запуска Run	3,67	3,67	3,88	4,40

Интерфейс ...

Eclipse



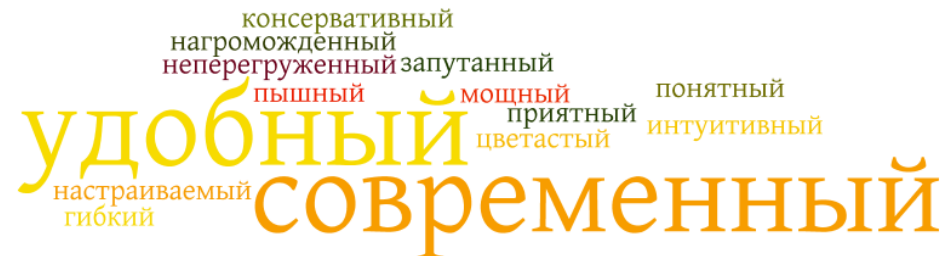
IDEA



NetBeans



Visual Studio



<http://www.wordle.net/create>

Сравнительная характеристика

Параметр/IDE	Eclipse	IDEA	NetBeans	VS
Удобство	2,83	4,17	3,50	3,80
Функциональность	4,33	4,67	3,50	4,20
Производительность	3,50	4,00	3,13	4,00
Дизайн	2,67	4,17	3,00	4,00
Понятность названий	3,33	4,00	3,88	3,60
Отзывчивость	3,17	4,00	3,63	3,80
Уверенность в среде	2,83	4,17	3,63	3,60
Соответствие ожиданиям	3,67	4,17	4,13	4,00
Соответствие требованиям	3,00	5,00	3,25	4,60
Средний балл	3,26	4,26	3,51	3,96
Среднее по заданиям	3,28	4,32	3,78	3,64
Общее среднее	3,27	4,29	3,64	3,80

Рейтинг IDE

Место	IDE	Средняя оценка
1	IntelliJ IDEA	4,29
2	Visual Studio	3,80
3	NetBeans IDE	3,64
4	Eclipse IDE	3,27

СПАСИБО ЗА ВНИМАНИЕ!

E-mail: sofia.chebanova@gmail.com
alxmkv@gmail.com

Twitter: [@Sofikowa](https://twitter.com/Sofikowa)
[@alxmkv](https://twitter.com/alxmkv)