

Лабораторная работа № 1 Основы работы в Eclipse IDE

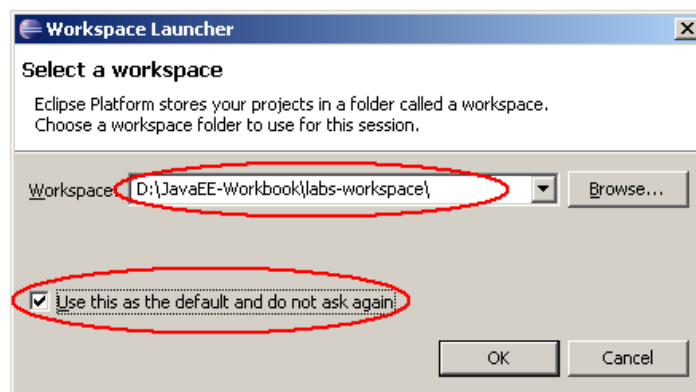
Целью данной работы является общее ознакомление с работой в IDE Eclipse. Для того чтобы выполнять задания по курсу Java- программирование на своих домашних десктопах или портативных ПК, установите Java 10 SE и соответствующую среду времени выполнения (java runtime environment) с сайта [oracle.com](https://www.oracle.com/downloads/) (<https://www.oracle.com/downloads/>).

1. Установка и запуск Eclipse

Для домашних компьютеров (в классах 204 КЦ все установлено) скачайте и установите содержимое с сайта Eclipse Foundation <https://www.eclipse.org/downloads/packages/release/> (версия Photon). Предположим, что установка была выполнена в C:\. Для запуска Eclipse зайдите в каталог C:\eclipse и запустите файл eclipse.exe.

2. Создание рабочей области (workspace)

При первом запуске Eclipse предлагает выбрать каталог рабочей области (Workspace). Это каталог, в котором будут храниться все файлы, относящиеся к проектам – настройки проектов, исходные файлы программ, результаты компиляции и сборки и т.д.



Укажите каталог рабочей области, выберите флажок Use this as the default... и нажмите ОК.

При дальнейшей работе, переключение между рабочими областями, а также создание новой рабочей области выполняется с помощью команды меню File/Switch Workspace.

3. Основные представления, перспективы и редакторы

Прежде чем приступить к созданию проекта Eclipse, познакомимся поближе с основными элементами Eclipse Workbench.

Eclipse Workbench состоит из:

- перспектив (perspectives);
- представлений (views);
- редакторов (editors).

Перспектива – группа представлений и редакторов в окне Workbench. Количество перспектив в одном окне Workbench не ограничено. Так же и перспектива может содержать неограниченное количество представлений и редакторов. В одном окне каждая из перспектив может иметь различные наборы представлений, но все перспективы делят между собой один и тот же набор редакторов.

Для подключения перспективы используется команда основного меню Window/Open Perspective. Перечень открытых перспектив по умолчанию отображаются в правой верхней части экрана. С помощью этого же списка выполняется переключение между перспективами:



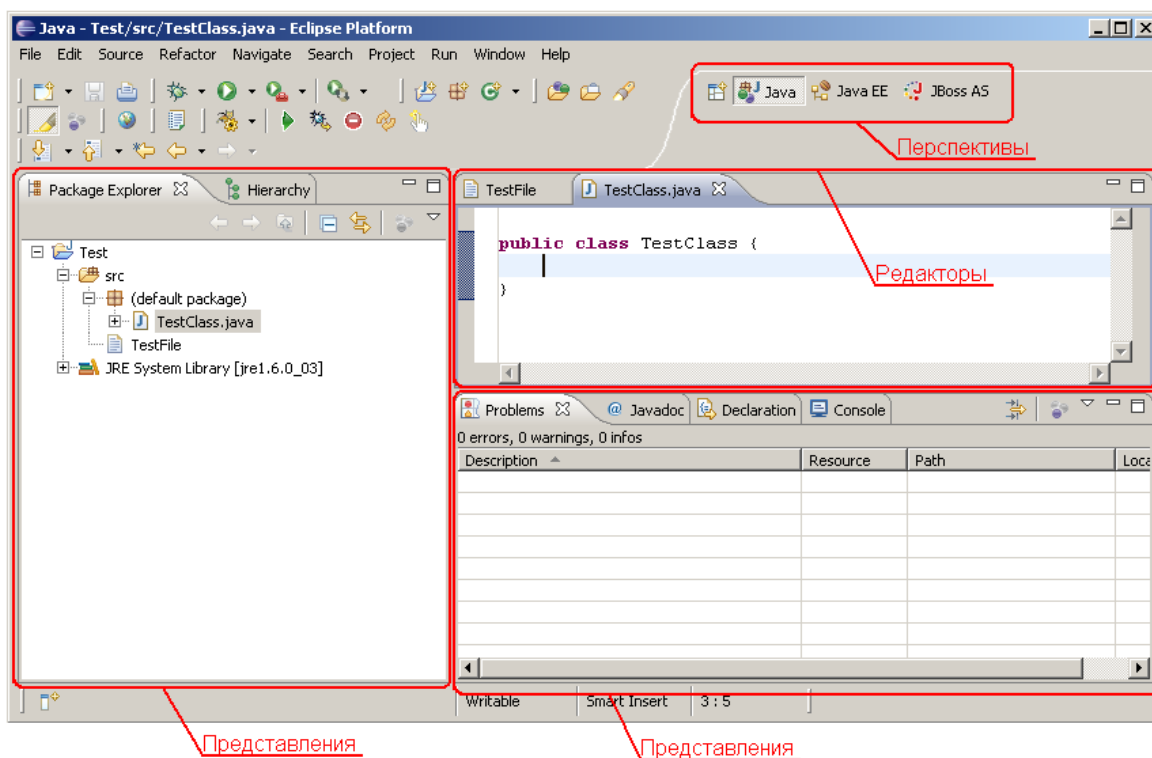
Представление – это визуальный компонент Workbench. Обычно он используется для перемещения по иерархии некоторой информации (например, структуры проекта), открытия редакторов, просмотра результатов выполнения и т.д. Все изменения, сделанные в представлениях, немедленно сохраняются. В большинстве случаев, в окне Workbench может существовать только один экземпляр представления некоторого типа.

Многие представления входят в состав тех или иных перспектив и отображаются при их открытии. Тем не менее, с помощью команды меню Window/Show View можно открыть любое представление, независимо от текущей перспективы.

Восстановить набор представлений текущей перспективы можно с помощью команды Window/Reset Perspective.

Редактор также является визуальным компонентом Workbench. Используется для просмотра и редактирования некоторого ресурса, например Java-класса или файла, содержащего SQL-запросы. Изменения, делающиеся в редакторе, подчиняются модели жизненного цикла «открыть-сохранить-закрыть» (an open-save-close lifecycle model). В Workbench может существовать несколько экземпляров редакторов.

Редакторы того или иного типа обычно отображаются автоматически при двойном щелчке на соответствующий ресурс (например, Java-класс). Сохранение данных выполняется с помощью команды меню File/Save, либо нажатием на клавиши Ctrl-S.



В данный момент времени может быть активным только одно представление или редактор. Активным является представление или редактор у которого подсвечен заголовок. На активный элемент будут воздействовать общие операции вырезки, копирования и вставки (cut, copy, paste). Также активный элемент обуславливает содержимое строки состояния (status line). Если закладка редактора белая, то это означает, что данный редактор неактивен, однако представления могут отображать информацию, полученную из редактора, бывшего активным последним.

4. Создание проекта и ресурсов проекта

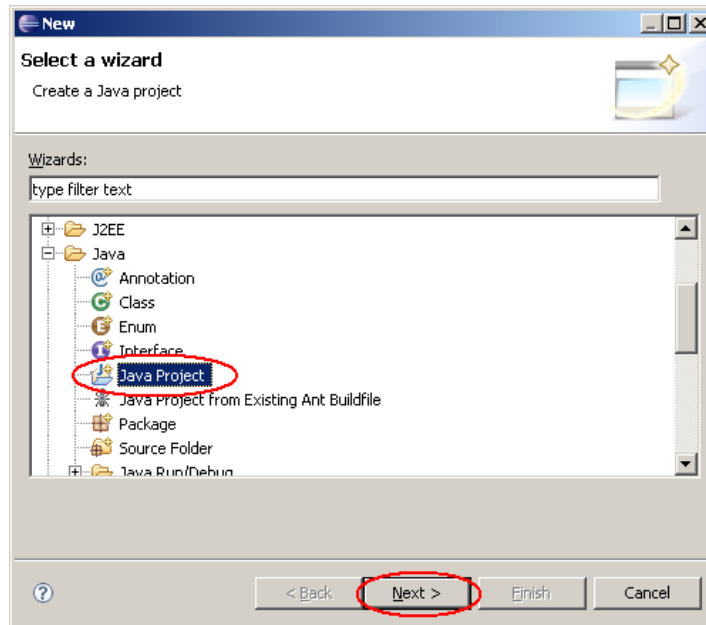
Проект Eclipse – это совокупность служебных и конфигурационных файлов Eclipse и ресурсов проекта, относящихся к конкретному разрабатываемому приложению или подсистеме приложения. Физически на диске проект представляет собой каталог по имени проекта внутри каталога рабочей области, внутри которого содержатся файлы – составляющие проекта.

Под **ресурсом проекта** понимаются любые элементы, из которых строится приложение – Java-классы, Java-библиотеки, HTML-страницы, изображения, текстовые файлы, наборы SQL-скриптов и т.д. Строго говоря, сам проект также является ресурсом. Физически каждый ресурс проекта представлен каталогами и файлами соответствующего типа.

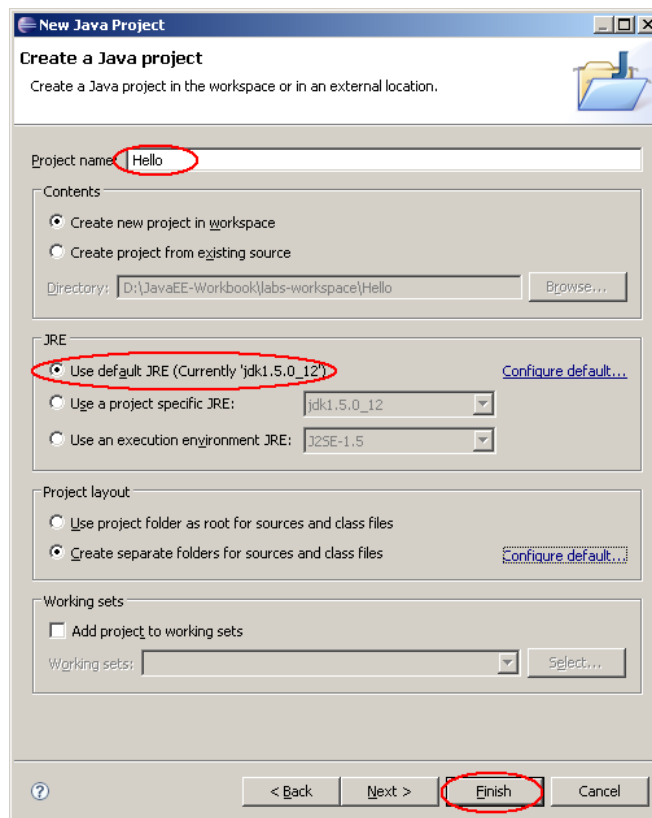
Существует несколько способов создания проектов и ресурсов проекта. Основным из них является команда основного меню File/New. С помощью подпункта Other можно отобразить все типы ресурсов, доступных для создания. Для удобства выбора типы создаваемых проектов и ресурсов делятся на условные категории, обычно в зависимости от типа

разрабатываемого приложения. Например, в категории Java размещаются обычные Java-проекты, Java-классы, интерфейсы, пакеты и т.д., а в категории Web – Web-проекты, HTML-страницы, JSP-страницы и т.д.

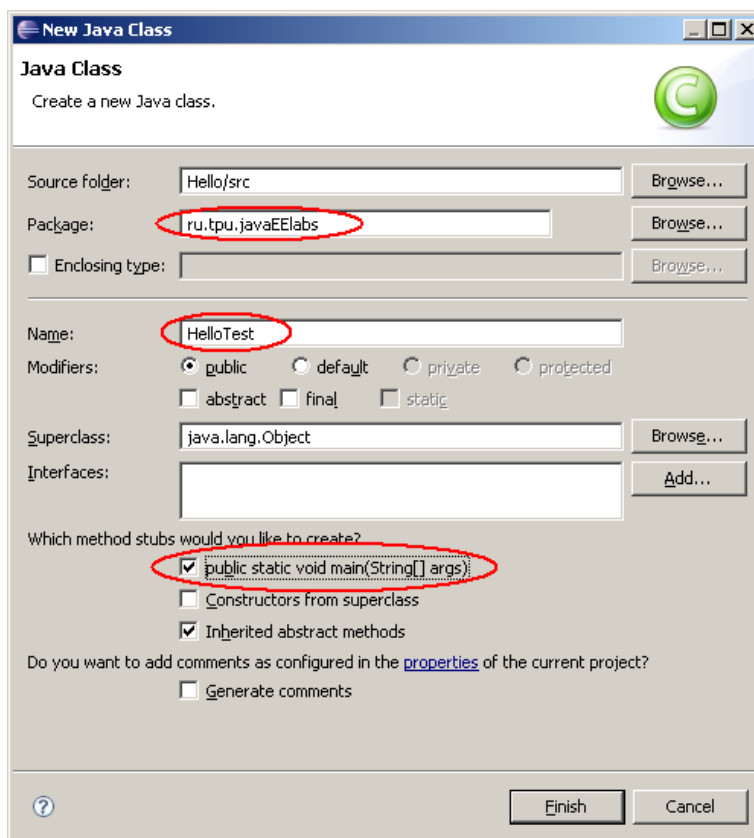
- 1) Закройте перспективу Welcome, появившуюся на экране после первого запуска Eclipse.
- 2) Выберите в меню File/New/Other.
- 3) Выберите категорию Java, а в ней Java Project и нажмите Next.



- 4) Введите имя проекта – Hello. Затем, если в разделе JRE не указана или указана неверная настройка JRE, выберите Configure default... и добавьте путь к JDK, установленной на предыдущем этапе (например, C:\jdk1.8.0_12).



- 5) Нажмите Finish.
- 6) Eclipse предложит открыть перспективы, связанные с разработкой Java-приложения. Выберите флажок Remember my decision и нажмите Yes.
- 7) В результате создания проекта автоматически открывается перспектива Java, в левой части окна открывается представление Package Explorer, в котором отражается новый проект Hello, содержащий подкаталог src, предназначенный для хранения исходных файлов программы.
- 8) Далее, нам необходимо добавить новый ресурс проекта – обычный Java-класс:
- 9) В представлении Package Explorer нажмите правой кнопкой мыши на значок проекта Hello и выберите New/Class. То же самое действие можно выполнить с помощью команды File/New/Other, затем выбрав категорию Java и тип ресурса Class.
- 10) В появившемся окне укажите имя пакета и имя нового класса. Имя пакета может состоять из нескольких уровней, разделенных точкой. Также выберите флажок public static void main(...) для автогенерации main-метода.
- 11) Нажмите Finish



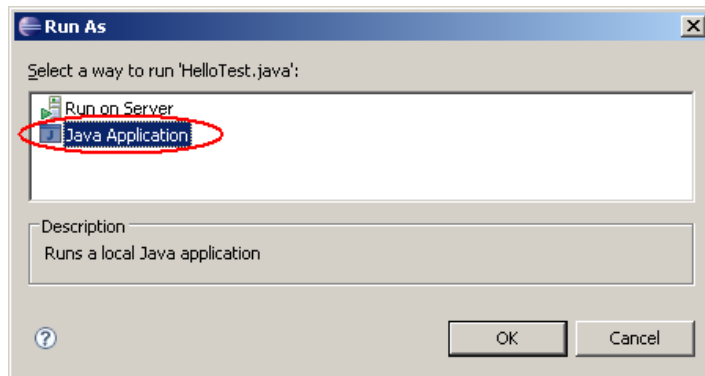
12) В методе main запишите команду вывода сообщения на экран:

```
public static void main(String[] args) {  
    System.out.println("Hello Java");  
}
```

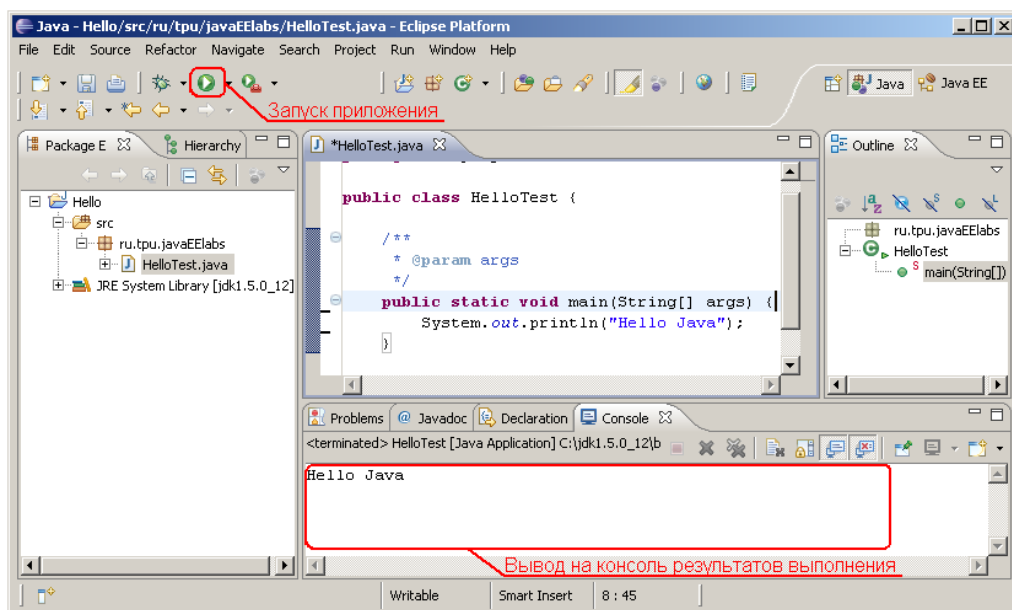
13) Сохраните файл, нажав на клавиши Ctrl-S.

5. Запуск приложения

- 1) Для запуска приложения выделите файл Hello.java в представлении Package Explorer, выберите команду меню Run/Run или нажмите на соответствующую кнопку панели инструментов. Запустить приложение можно также нажав правой кнопкой мыши на Hello.java в представлении Package Explorer и выбрав команду Run As/Java Application.
- 2) В появившемся окне Run As... выберите Java Application и нажмите OK.

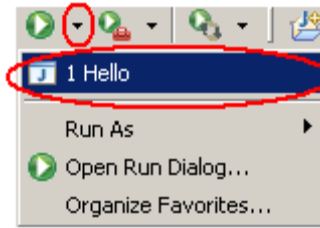


Если появилось окно сохранения файлов перед запуском Save and Launch, выберите флажок Always save resources... и нажмите OK.



3) После запуска приложения в нижней части экрана в окне Console отображается результат выполнения программы.

При первом запуске приложения создается конфигурация запуска, которая используется при последующих запусках программы. В дальнейшем, при выполнении лабораторных работ, рабочая область будет содержать несколько проектов, и соответственно несколько Java-классов, содержащих метод `main()`. Для того, чтобы выбрать приложение для запуска, следует выделить соответствующий java-файл в представлении Package Explorer и выполнить Run/Run, либо нажав правой кнопкой мыши на java-файл в представлении Package Explorer и выполнить команду Run As/Java Application. Для повторных запусков приложения можно нажать на кнопку Run панели инструментов для запуска последнего запускавшегося приложения, либо открыть список конфигураций запуска и выбрать другое приложение:



6. Автогенерация и автоформатирование кода

Ниже приведены некоторые полезные функции по генерации стандартных конструкций в Java-коде, а также автоматическое форматирование кода. Функции автогенерации кода сосредоточены в меню Source и начинаются со слова Generate (кроме Override/Implement Methods). Эти функции обрабатывают только на коде активного редактора java-кода.

Override/Implement Methods	переопределить/реализовать метод базового класса/интерфейса
Generate Getters And Setters	создать методы get и/или set для полей класса
Generate Delegate Methods	создать методы-делегаты для методов объекта, определенного в классе
Generate hashCode() and equals()	переопределить и сгенерировать содержимое методов hashCode() и equals()
Generate Constructors using Fields	создать конструктор, принимающий поля класса в качестве параметров
Generate Constructors using Superclass	создать конструктор используя суперкласс

Ниже приведен пример работы команды Generate Constructors using Fields:

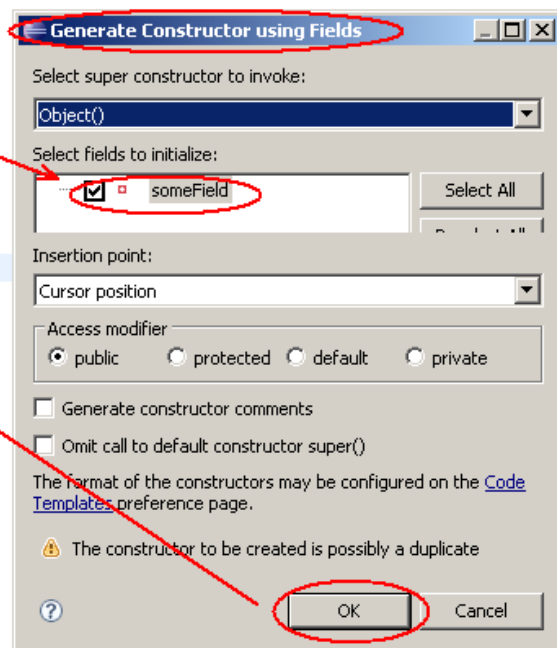

```
package ru.tpu.javaEELabs;
```

```
public class Hello {
```

```
    private String someField;
```

```
    public Hello(String someField) {  
        super();  
        this.someField = someField;  
    }
```

```
}
```



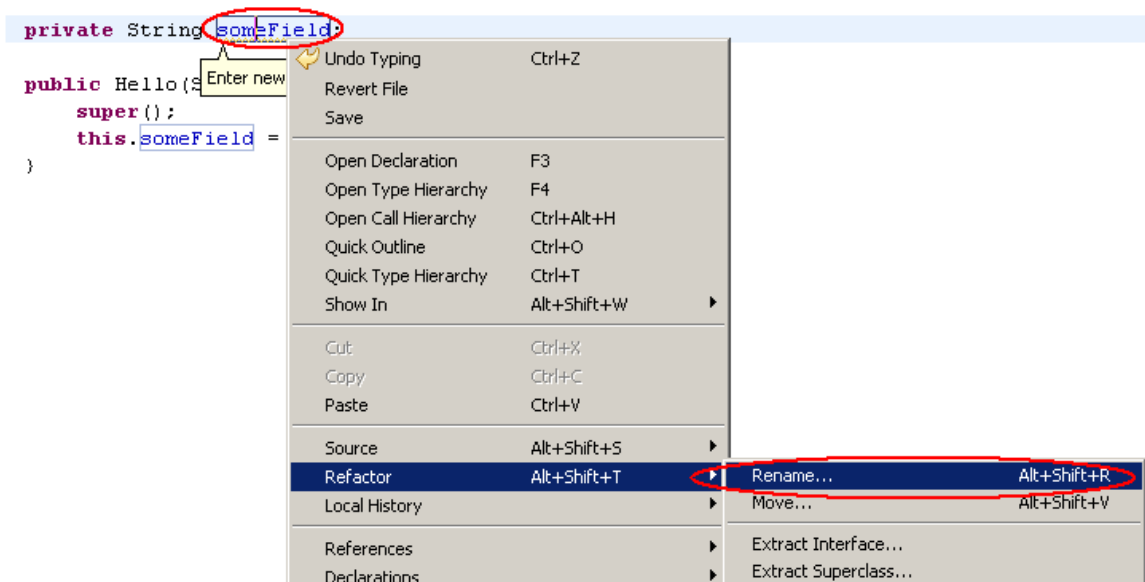
Автоформатирование кода выполняется с помощью команды меню Source/Format, либо с помощью комбинации клавиш Ctrl-Shift-F. Автоформатирование выполняется над текущим файлом открытым в окне редактора.

7. Рефакторинг

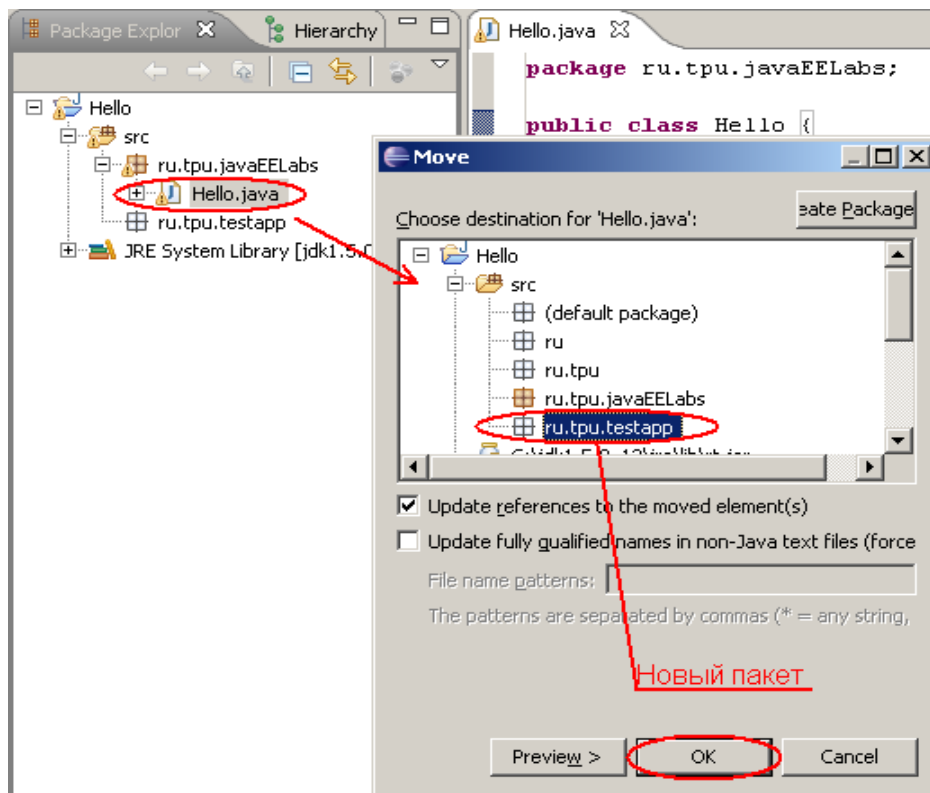
Рефакторинг – это безопасное изменение структуры ресурсов проекта. Действия по рефакторингу включают в себя изменение названий классов, полей и методов, изменение принадлежности классов пакетам, извлечение интерфейсов из описания класса и пр. При выполнении действия рефактор (refactor) обеспечивает сохранение целостности кода. Например, изменение имени класса приводит к изменению этого имени во всех элементах кода, где используется этот класс.

Команды рефакторинга вызываются над определенным элементом – классом, полем, методом и т.д. Функции рефактора (refactor) удобнее всего вызывать путем выбора элемента, вызова щелчком правой кнопкой мыши контекстного меню и выбора в нем одной из команд меню Refactor. Также команды рефакторинга можно вызвать, используя основное меню Refactor.

Ниже приведен пример вызова команды переименования над полем класса. Перед вызовом меню курсор редактора предварительно устанавливается на изменяемое поле someField.




Следующий пример демонстрирует возможность смены пакета, в котором размещается класс. Действие рефактора Move вызывается над классом Hello в представлении Package Explorer. В результате выполнения действия класс перемещается из пакета ru.tpu.javaEELabs в пакет ru.tpu.testapp.

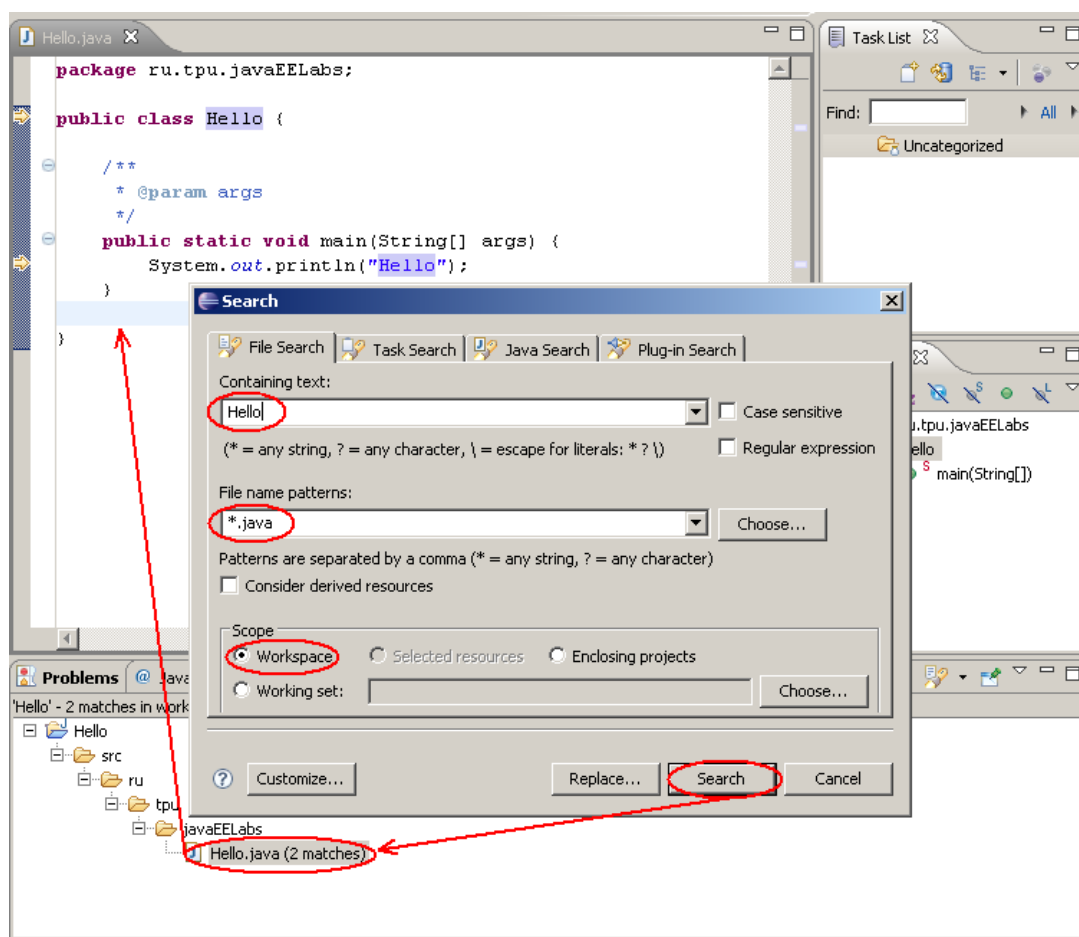


8. Поиск

Eclipse Workbench позволяет производить поиск файлов и строк, а также замену (без соблюдения целостности проекта) одной строки на

другую. Рассмотрим пример поиска файлов в рабочей области, содержащих слово “Hello”.

- 1) Нажмите на кнопку поиска  или сочетание клавиш Ctrl-N
- 2) В появившемся окне выберите закладку File Search и в поле Containing Text введите строку Hello.
- 3) Опция File Name Patterns определяет шаблон имен файлов для поиска. Изменить шаблон можно путем прямого ввода нового значения в поле, либо с помощью кнопки Choose. По умолчанию выбран шаблон *.java, т.е. все файлы с расширением java.
- 4) Опция Scope определяет область поиска. Убедитесь, что в этом разделе выбран пункт Workspace (по всей рабочей области).
- 5) Нажмите Search.
- 6) В результате выполнения поиска отображается представление Search, в котором показаны результаты поиска. Щелчок на найденный файл открывает его в окне редактора.



Задание

1. Выполнить вышеуказанные пункты (1-8) методических указаний
2. Сформировать отчет по проделанной работе (Word), включая скриншоты выполнения каждого пункта.
3. К отчету приложить файлы Hello.java и Hello.class

4. Отправить по E-mail файл отчета и файлы Hello.java, Hello.class преподавателю
5. Титульный лист оформлять следующим образом:

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт – ИШИТР
Отделение – ОИТ
Направление – Программная инженерия

Лабораторная работа №1
Основы работы в Eclipse IDE

по дисциплине: Программирование на Java

Выполнил: студент гр. 8K81 _____ Петров П.П.
(Номер группы) (Подпись) (Ф.И.О.)

Проверил: доцент ОИТ ИШИТР _____ Фофанов О.Б.
(Подпись) (Ф.И.О.)

Томск 2019 г.