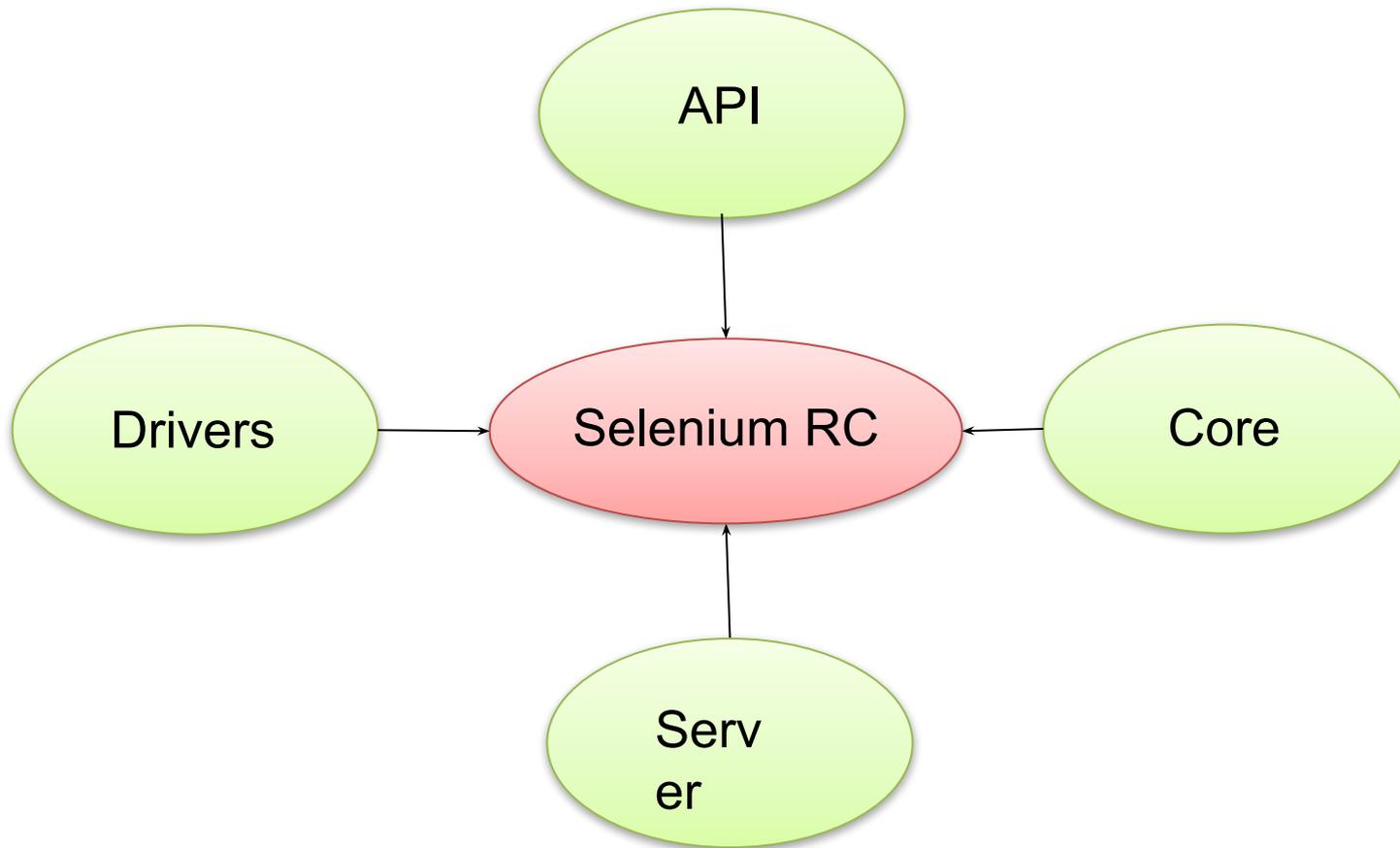


# AT.03.001 Selenium RC

Resource Development Dep.

Author: Kiryl Zayets

# Selenium RC



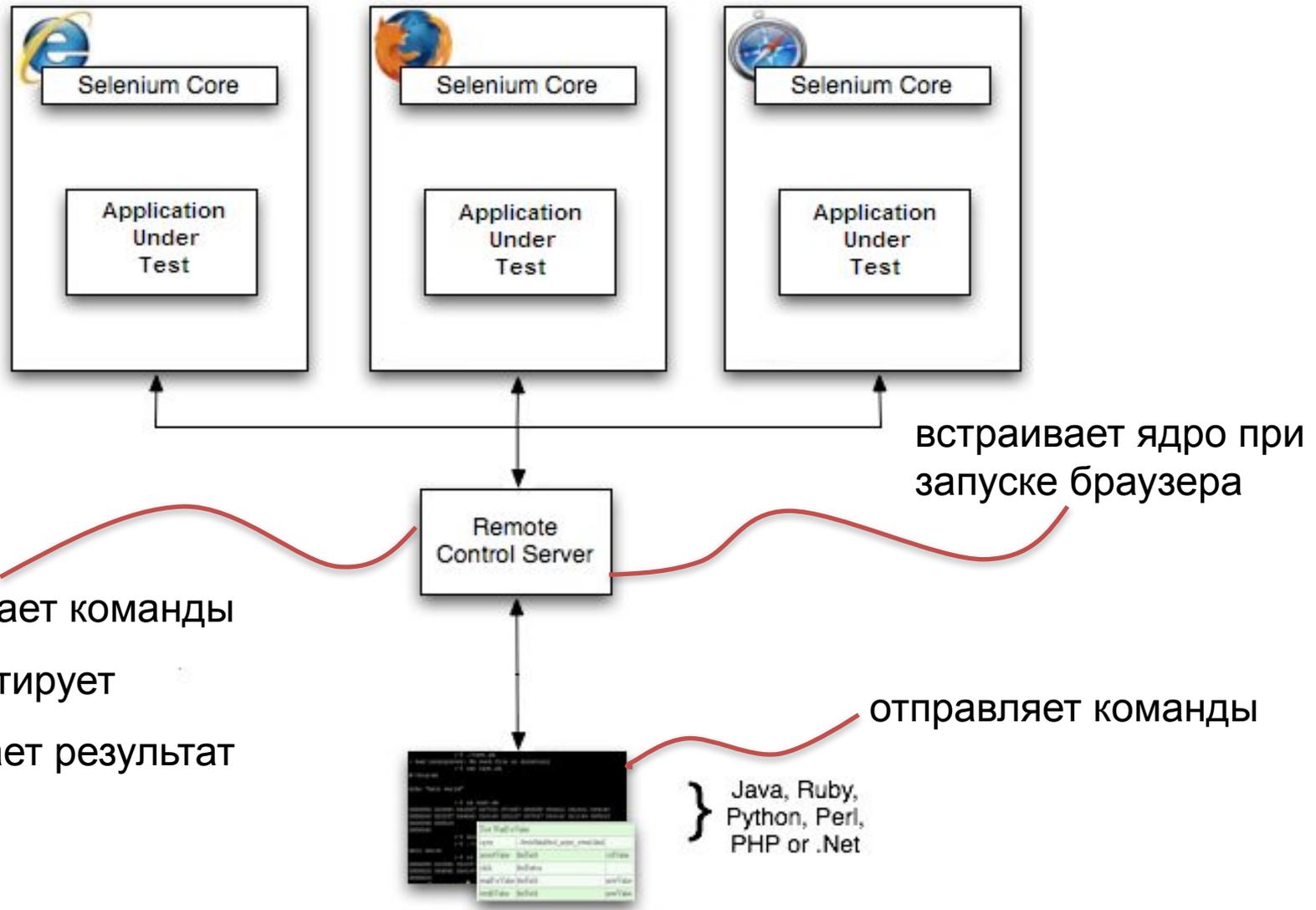
## Selenium server

```
java -jar selenium-server.jar
```

```
C:\WINDOWS\system32\cmd.exe - java -jar selenium-server.jar
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

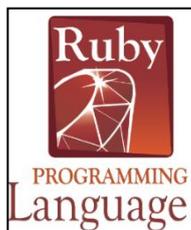
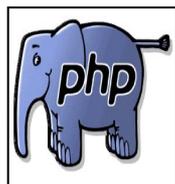
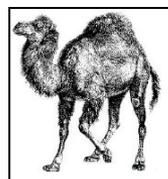
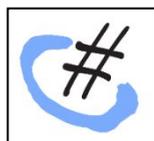
D:\Documents\Selenium presentation\Selenium RC\Demo\lib>java -jar selenium-server.jar
11:52:19.936 INFO - Java: Sun Microsystems Inc. 17.0-b17
11:52:19.936 INFO - OS: Windows XP 5.1 x86
11:52:20.108 INFO - v1.0.1 [2696], with Core v@VERSION@ [ @REVISION@]
11:52:20.436 INFO - Version Jetty/5.1.x
11:52:20.436 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
11:52:20.436 INFO - Started HttpContext[/selenium-server,/selenium-server]
11:52:20.436 INFO - Started HttpContext[/,/]
11:52:20.515 INFO - Started SocketListener on 0.0.0.0:4444
11:52:20.515 INFO - Started org.mortbay.jetty.Server@4a65e0
```

## Место сервера в процессе



## Клиентские драйверы

Клиентские библиотеки обеспечивают поддержку языков, которые позволяют запускать команды из вашего кода, написанного на OO языке. Существуют различные клиентские библиотеки для каждого из поддерживаемых языков.



Любой язык исполняющий  
HTTP!

# Ядро

JavaScript фреймворк (набор инструкций);

Встраивается внутрь браузера и эмулирует действия пользователя;

Выполняет тесты на различных платформах и браузерах;



## Организация тестирования

Для начала необходимо средство для организации тестов (тестовый движок) TestNg или junit (Java).

```
package example1;

import org.testng.annotations.*;

public class SimpleTest {

    @BeforeClass
    public void setUp() {
        // code that will be invoked when this test is instantiated
    }

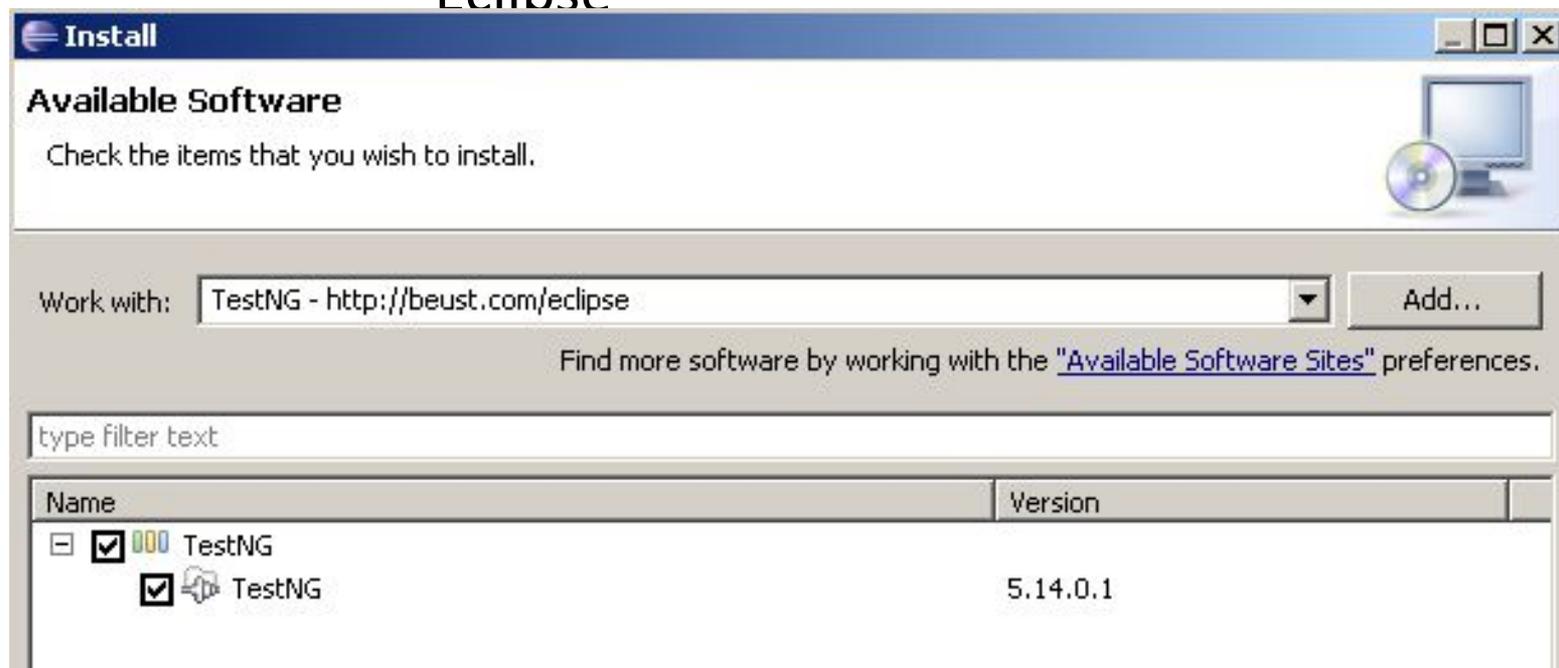
    @Test(groups = { "fast" })
    public void aFastTest() {
        System.out.println("Fast test");
    }

    @Test(groups = { "slow" })
    public void aSlowTest() {
        System.out.println("Slow test");
    }

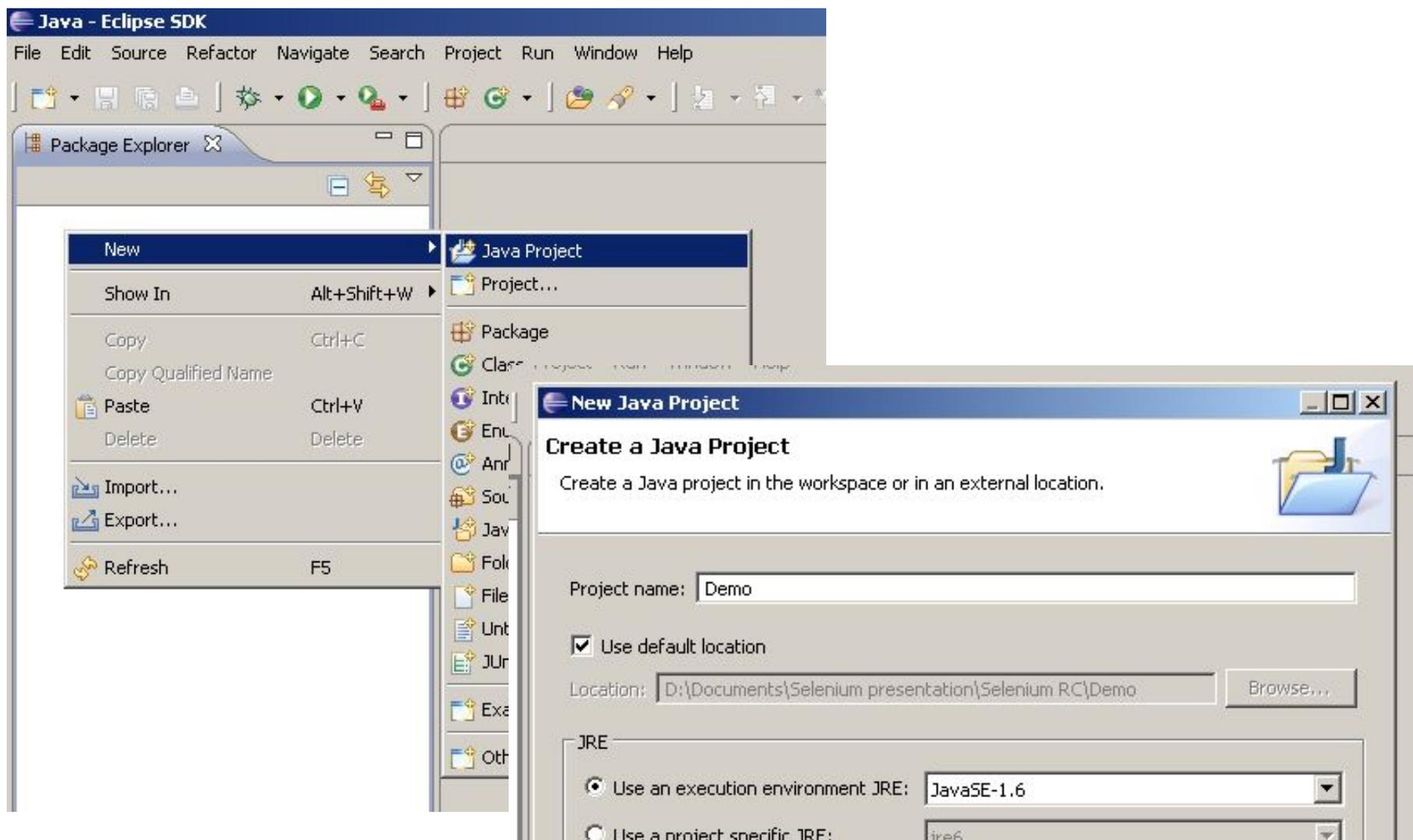
}
```

# С чего начать ?

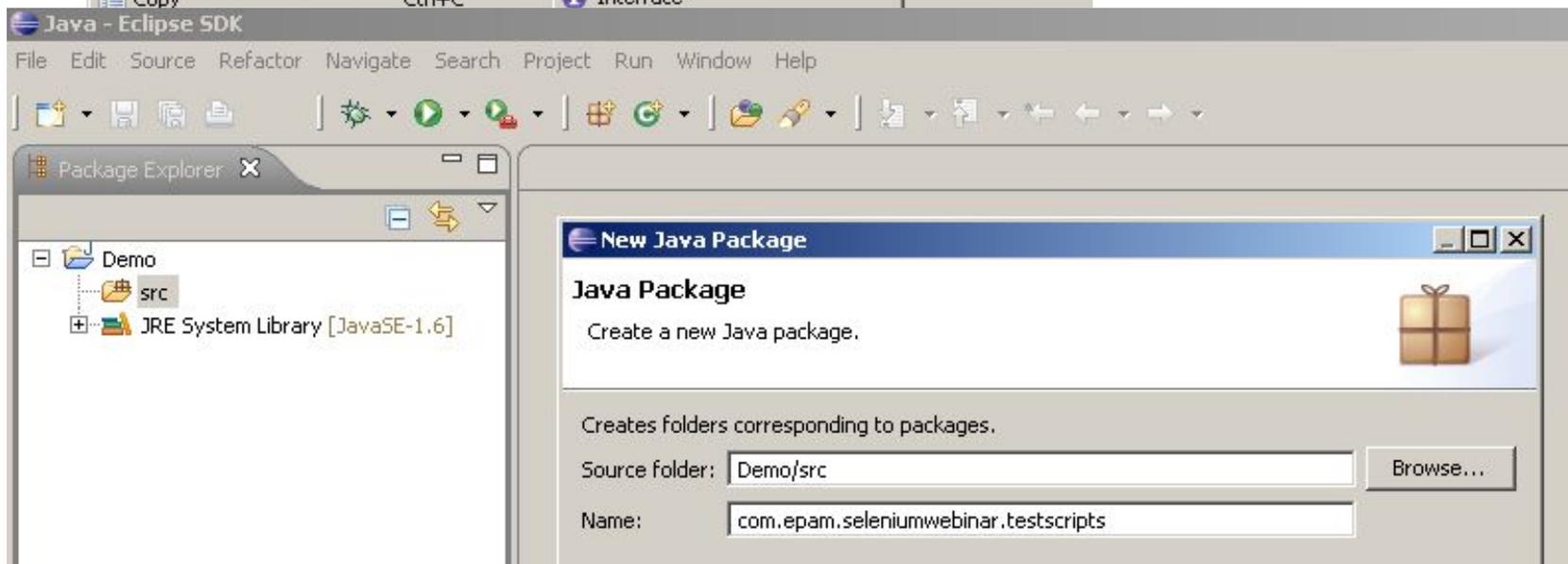
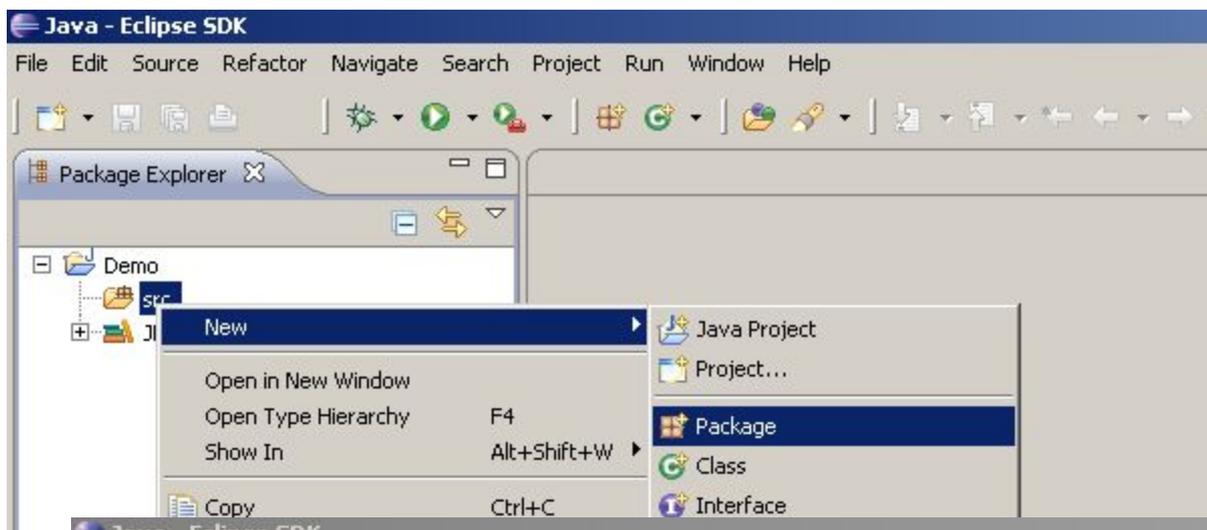
## TestNG плагин для Eclipse



# Создаем проект



# Создаем пакет



# Добавляем тест

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project named 'Demo' with a 'src' folder and a package 'com.epam'. A context menu is open over the package, with the 'New' option selected. The 'New' submenu is visible, showing various options like 'Annotation', 'Source Folder', 'Java Working Set', 'Folder', 'File', 'JUnit Test Case', and 'Other...'. The 'TestNG class' wizard is open, showing the following configuration:

- Source folder: /Demo/src
- Package name: com.epam.seleniumwebinar.testscripts
- Class name: NewTest
- Annotations:
  - @BeforeMethod
  - @AfterMethod
  - @DataProvider
  - @BeforeClass
  - @AfterClass
  - @BeforeTest
  - @AfterTest
  - @BeforeSuite
  - @AfterSuite
- XML suite file: (empty field)

# Добавляем библиотеку Selenium и клиентский драйвер

The screenshot shows the Eclipse IDE interface. On the left, a file explorer window displays the directory structure: D:\Documents\Selenium presentation\Selenium RC\Demo\lib. It contains three files: log4j-1.2.15.jar (383 KB), selenium-java-client-driver.jar (54 KB), and selenium-server.jar (5,465 KB). The main IDE window shows the Package Explorer on the left with the 'lib' folder selected, containing the same three jars. The editor on the right shows the code for 'NewTest.java' with imports for TestNG annotations. A context menu is open over the 'lib' folder, with 'Build Path' selected, and a sub-menu is open showing 'Add to Build Path' and 'Configure Build Path...'. The IDE title bar indicates the current project is 'Java - Demo/src/com/epam/seleniumwebinar/testscripts/NewTest.java - Eclipse SDK'.

Name	Size	Type	Date Modified
log4j-1.2.15.jar	383 KB	Executable Jar File	8/25/2007 12:29 AM
selenium-java-client-driver.jar	54 KB	Executable Jar File	6/9/2009 11:51 AM
selenium-server.jar	5,465 KB	Executable Jar File	6/9/2009 11:51 AM

```
package com.epam.seleniumwebinar.testscri

import org.testng.annotations.Test;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.AfterClass;

public class NewTest {
    @Test
    public void f() {
        class
    }
    public void beforeClass() {
        class
    }
    public void afterClass() {

```

## Пишем тест

```
NewTest.java X
public class NewTest {

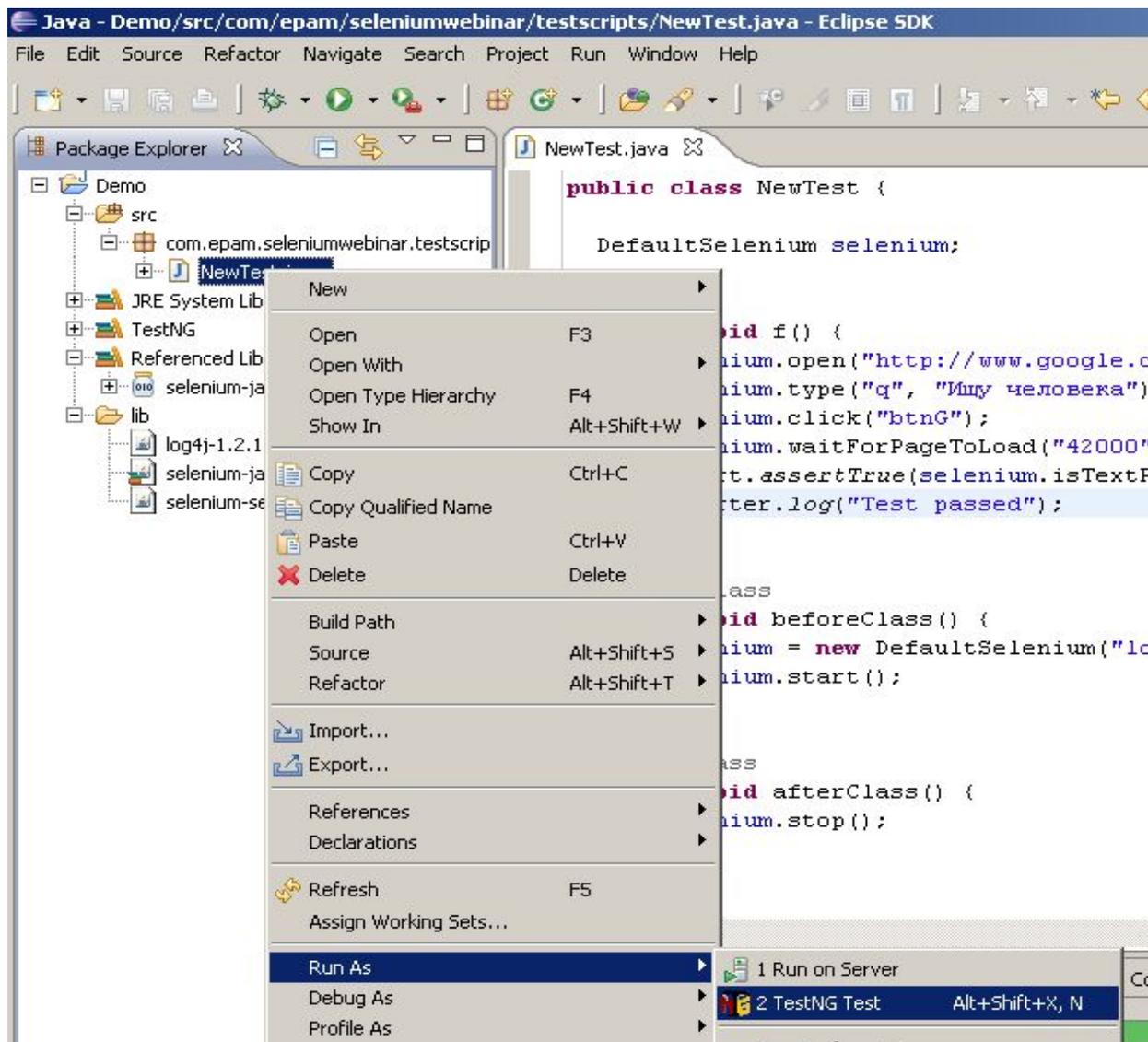
    DefaultSelenium selenium;

    @Test
    public void f() {
        selenium.open("http://www.google.com");
        selenium.type("q", "Ищу человека");
        selenium.click("btnG");
        selenium.waitForPageToLoad("42000");
        Assert.assertTrue(selenium.isTextPresent("Человек"), "Не нашёл человека");
    }

    @BeforeClass
    public void beforeClass() {
        selenium = new DefaultSelenium("localhost", 4444, "**firefox", "http://example.com/");
        selenium.start();
    }

    @AfterClass
    public void afterClass() {
        selenium.stop();
    }
}
```

# Запуск теста



# Результат

```
@Test
public void f() {
    selenium.open("http://www.google.com");
    selenium.type("q", "Ищу человека");
    selenium.click("btnG");
    selenium.waitForPageToLoad("42000");
    Assert.assertTrue(selenium.isTextPresent("человек"), "Не нашёл человека");
    Reporter.log("Test passed");
}

@BeforeClass
public void beforeClass() {
    selenium = new DefaultSelenium("localhost", 4444, "*firefox", "http://example.
    selenium.start();
}

@BeforeClass
```

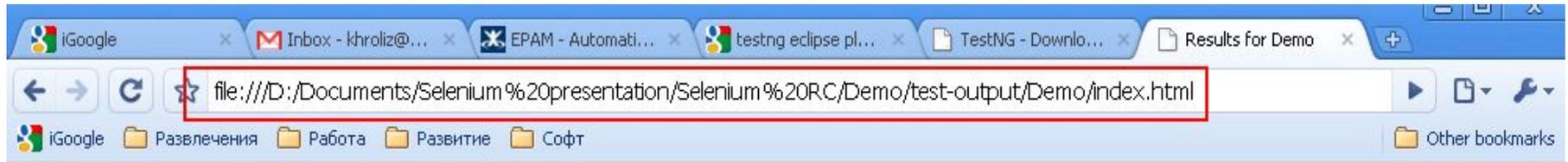
Problems Javadoc Declaration TestNG Console

Tests: 0/1 Methods: 1/1 (10594 ms)  Passed: 1  Failed: 0

All Tests Failed Tests Failure Exception

- Demo ( 1/0/0/0 )
  - com.epam.seleniumwebinar.testscripts.NewTest ( 1/0/0/0 )
    - com.epam.seleniumwebinar.testscripts.NewTest.f()

# Результат



## Results for *Demo*

1 test	<a href="#">1 class</a>	1 method: <a href="#">chronological</a> <a href="#">alphabetical</a> <a href="#">not run (0)</a>
<a href="#">0 group</a>	<a href="#">reporter output</a>	<a href="#">testng.xml</a>

com.epam.seleniumwebinar.testscripts.NewTest  
(1/0/0)

## com.epam.seleniumwebinar.testscripts.NewTest

Tests passed/Failed/Skipped:	1/0/0
Started on:	Mon Aug 30 12:10:40 EEST 2010
Total time:	10 seconds (10140 ms)
Included groups:	
Excluded groups:	

(Hover the method name to see the test class name)

PASSED TESTS			
Test method	Exception	Time (seconds)	Instance
f Test class: com.epam.seleniumwebinar.testscripts.NewTest <a href="#">Hide output</a> <a href="#">Show all outputs</a> Test passed		1	com.epam.seleniumwebinar.testscripts

## Базовый класс для всех тест кейсов

```
public class BaseTest {
    SeleniumServerActions serverActions;
    @BeforeSuite
    public void beforeSuite() throws Exception{
        serverActions = new SeleniumServerActions();
        serverActions.start();
    }
    @BeforeClass
    public void beforeClass() throws Exception{

        Selenium = new SeleniumCore(browserSettings.host.getValue(),
            Integer.parseInt(browserSettings.port.getValue()),
            browserSettings.browser.getValue(),
            browserSettings.url.getValue());
        Selenium.start();
    }

    @AfterClass
    public void afterClass() throws Exception{
        Selenium.stop();
    }
    @AfterSuite
    public void afterSuite() throws Exception{
        serverActions.stopServer();
    }
}
```

```
public class NewTest extends BaseTest {

    @Test
```

## Где хранить настройки?

- ✓ Найдите грань между настройками и тестовыми данными:
  - ✓ Selenium host
  - ✓ Selenium port
  - ✓ Timeout
  - ✓ Base URL
- ✓ properties-файлы – стандарт для Java

## Где хранить тестовые данные?

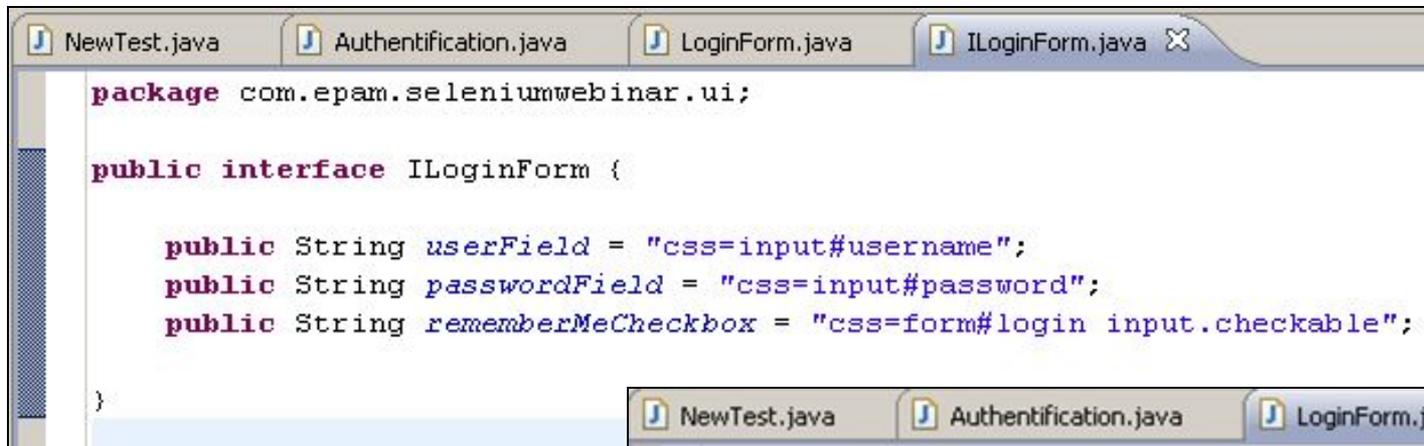
- ✓ Excel
- ✓ CSV-файлы
- ✓ База данных
- ✓ XML

<?xml?>



## Где хранить локаторы?

- ✓ Поля интерфейсов



```
package com.epam.seleniumwebinar.ui;

public interface ILoginForm {

    public String userField = "css=input#username";
    public String passwordField = "css=input#password";
    public String rememberMeCheckbox = "css=form#login input.checkable";

}
```



```
package com.epam.seleniumwebinar.ui;

public class LoginForm implements ILoginForm {

    public void checkRememberMe() {
        //some code
    }

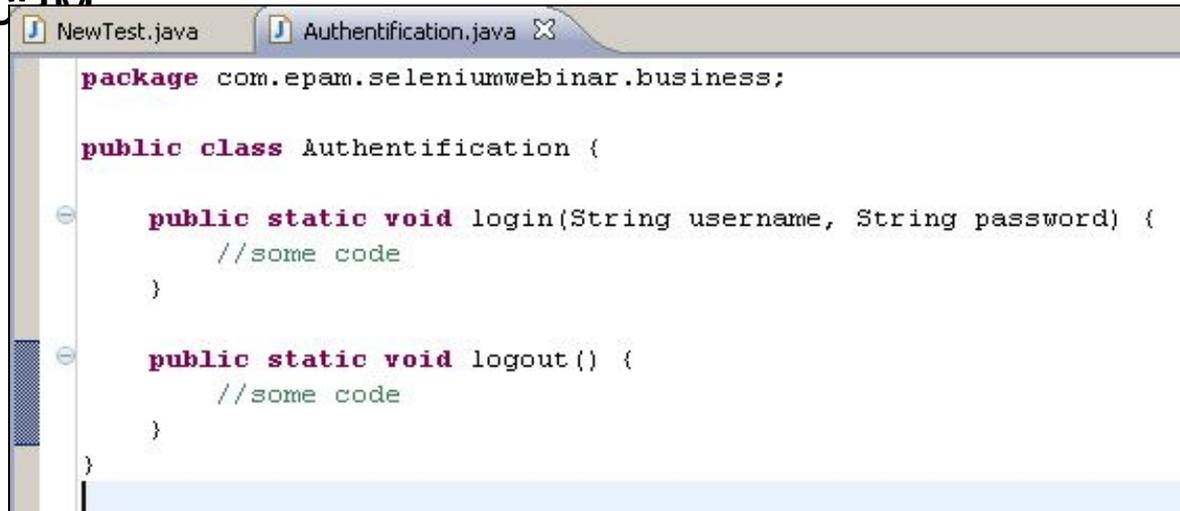
}
```

## Структура UI Layer

- ✓ Разбивка на страницы
- ✓ Сбор «особенных» UI элементов в отдельные классы:
  - ✓ Различные кастомизированные кнопки-поля (нужно вызывать специальные события)
- ✓ Иерархия объектов

## Структура Business Layer

- Набор статических методов, объединённых по классам



```
package com.epam.seleniumwebinar.business;

public class Authentication {

    public static void login(String username, String password) {
        //some code
    }

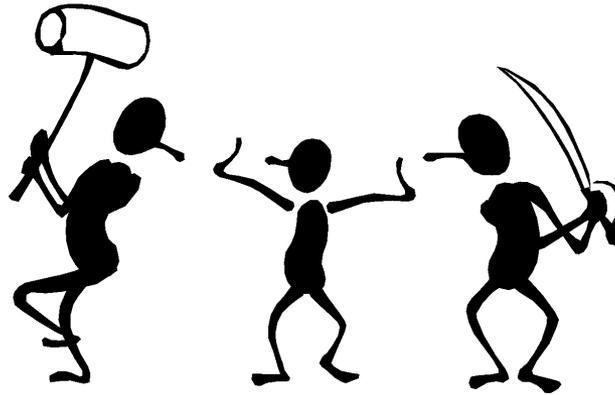
    public static void logout() {
        //some code
    }
}
```

- Чем проще, тем лучше

## Итого

- ✓ Selenium RC – это API для манипуляции с браузером
- ✓ Лучшие практики
  - ✓ Интеграция с инструментами для unit-тестирования
  - ✓ UI Layer
  - ✓ Business Layer
  - ✓ Тестовые данные

# Вопросы?



# Thanks for Your Attention

EPAM Systems — AT.03.001.SeleniumRC

By

Kiryl Zayets

[www.epam.com](http://www.epam.com)



EPAM Systems

41 University Drive, Suite 202 | Newtown, PA 18940

**p:** +1 267 759 9000 | **f:** +1 +1 267 759 8989 | **e:** [info@epam.com](mailto:info@epam.com) | **w:** [www.epam.com](http://www.epam.com)