МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

УТВЕРЖДАЮ

Зав. кафедрой АОИ

_____Ю.П. Ехлаков

«___» _____ 2017 г.

МЕТОДЫ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Методические указания по выполнению лабораторных работ и организации самостоятельной работы для студентов, обучающихся по направлению подготовки бакалавра «Программная инженерия»

Разработчики

Ст. преподаватель каф. АОИ

_____ Безходарнов И.В

«____» _____ 2017 г.

2017

Оглавление

Введение ……	3
1. Методические указания по выполнению лабораторных работ …….	
ЛР № 1. Прототипирование интерфейсов ……..	4
ЛР № 2. Анализ проекта/идеи – изучение методики оценки ……	4
ЛР № 3. Технологическая оценка проекта …….	5
ЛР № 4. Автоматизированное функциональное тестирование Windows-приложений ……...	6
ЛР № 5. Автоматизированное функциональное тестирование WEB-сайтов …….	9
ЛР № 6. Поиск и эксплуатация уязвимостей в программном обеспечении (Windows-приложения) …….	12
ЛР № 7. Поиск и эксплуатация уязвимостей в программном обеспечении (ВЕБ сайты) …….	13
ЛР № 8. Continuous Integration – работа с системой автоматизации рутинных задач Jenkins CI …….	13
ЛР № 9. Системы контроля версий – на примере Mercurial (по желанию Git)	13
2. Методические указания по организации самостоятельной работы студента	14
Список рекомендуемой литературы ……	15

ВВЕДЕНИЕ

Целью дисциплины «Методы и технологии программирования» является формирование у студентов направления подготовки бакалавра 09.03.04 «Программная инженерия», навыков, позволяющих формулировать и решать задачи производства программного обеспечения от начальных этапов до выпуска готового продукта и дальнейшей его поддержки, используя методики проектирования, программирования, тестирования программных продуктов на необходимых этапах жизненного цикла программных проектов.

Теоретический курс дает представление о программировании как о технологическом процессе создания (производства) программного обеспечения как с точки зрения технологий, применяемых на разных этапах создания программных продуктов, так и вопросов организации этого производства. В комплексе рассматривается весь процесс от «идеи» до выпуска продукта, детально рассматриваются разные этапы и процесса создания программного обеспечения. При рассмотрении организационных вопросов используются рекомендации стандарта ИСО:9001:2008, преподаются Agile-технологии. При рассмотрении технологических вопросов акцент делается на современные методики, продукты, инструменты, применяемые в действующих компаниях по производству ПО.

В ходе самостоятельной деятельности студенты углубляют полученные знания, изучая отдельные технологические процессы создания ПО, применяя предложенные методики их решения. При выполнении лабораторных работ закрепляются теоретические знания и приобретаются практические навыки применения озвученных в теоретическом разделе методик.

Задачами преподавания дисциплины являются:

- изучение процесса создания ПО в целом и его отдельных этапов;
- изучение методик и технологий решения задач, возникающих на разных этапах процесса создания программного обеспечения;
- изучение вопросов организации технологических процессов с помощью методов их автоматизации.

В результате изучения дисциплины студент должен усвоить общий цикл производства программного обеспечения; приобрести навыки работы по организации различных этапов процесса создания программ; изучить методы эффективного решения технологических задач, возникающих при создании программного обеспечения, в том числе и с помощью методов их автоматизации.

Процесс изучения дисциплины «Методы и технологии программирования» направлен на **формирование профессиональной компетенции в производственной деятельности**:

- готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (**ОПК-3**);
- владением навыками использования различных технологий разработки программного обеспечения (**ПК-3**).

В результате изучения дисциплины **студент должен:**

знать:

- общий жизненный цикл программного обеспечения (ПО) как продукции;
- методы мониторинга, управления, оптимизации процесса производства ПО;
- методы автоматизации различных процессов производства программного обеспечения;
- методы решения типичных технологических задач, возникающих в производстве ПО;

уметь:

- организовывать процессы производства программного обеспечения;
- решать типичные технологические задачи, возникающие при производстве;
- писать техническое задание, тест-кейсы;

владеть:

- методикой оценки целесообразности разработки проектов;
- методами контроля эффективности разработки программного обеспечения.

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа №1. Прототипирование интерфейсов

Задание:

1. Проработать логику и внешний вид графического пользовательского интерфейса для одного из предложенных вариантов без использования средств автоматизации.

2. Найти в интернете любое подходящее бесплатное решение для прототипирования интерфейсов, прорисовать в нем разработанный вариант.

Для зачета представить:

1. Письменное описание бизнес логики работы интерфейса, и
2. Электронный и/или печатный вариант макетов, разработанного интерфейса, сделанных в системе прототипирования.

Варианты заданий:

1. Разработать интерфейсное решение, для связывания между собой удаленного ВЕБ-сервера и локального приложения. Представить интерфейс со стороны ВЕБ сервера и со стороны приложения.
2. Разработать интерфейсное решение – ВЕБ-форма, для регистрации пользователя на ВЕБ-сайте с одновременным представлением его фотографии (загрузкой фотографии на сервер).
3. Разработать интерфейсное решение, для управления удаленной, автоматизированной видео-камерой, имеющей функции – Zoom, повороты по горизонтали и вертикали, ночной режим съемки, автоматическое определение движения.

Лабораторная работа № 2. Анализ проекта/идеи – изучение методики оценки

Задание:

1. В соответствие с методикой, изложенной на лекции оценить один из предложенных вариантов ИТ-проекта.
2. По той же методики оценить собственную идею ИТ-проекта, либо уже существующий, но не из предложенных для п.1 вариантов.

Для зачета представить результаты оценки двух проектов в письменном виде.

Варианты проектов:

1. 1С-бухгалтерий
2. google.com
3. Microsoft Office
4. Navitel
5. 2gis
6. Skype
7. amazon cloud hosting
8. amazon.com

Методика проведения оценки:

<table>
<tr><td rowspan="2">Проблемы

Топ 3 проблемы</td><td>Решения
Топ 3 фичи</td><td rowspan="2">Уникальное предложение ценности

Простое понятное сообщение о том, почему нужно выбрать именно вас</td><td>Фора (нечестное преимущество)

Нельзя купить или быстро скорпировать</td><td rowspan="2">Сегменты заказчиков

Целевые заказчики или пользователи</td></tr>
<tr><td>Метрики оценки продукта

Метрики для оценки успешности продукта</td><td>Каналы продаж

Путь к вашим заказчикам</td></tr>
<tr><td colspan="3">Структура затрат

Фиксированные издержки
Хостинг
Зарплаты
И т.п.</td><td colspan="2">Потоки прибыли

Модели монетизации
Прибыль
Способы заработка</td></tr>
</table>

Лабораторная работа №3. Технологическая оценка проекта

Задание:

Выполнить технологическую оценку проекта для одного из предложенных вариантов по критериям, изученным на лекционных занятиях

Варианты:

1. Калькулятор для текстового режима работы ОС Linux.
2. Кроссплатформенный калькулятор с графической оболочкой для ОС семейства Windows, Linux, MAC OS/X.
3. ВЕБ-сервер с графическим интерфейсом для семейства ОС Winsows.
4. Среда разработки ВЕБ страниц с графическим интерфейсом.

Критерии оценки:

- принципиальная исполнимость проекта;
- платформы, ОС, среды разработки, дополнительное оборудование;
- требования к производительности конечного продукта;
- требования к интерфейсу конечного продукта;
- требования к совместимости с различными ОС;
- требуемые вспомогательные готовые продукты сторонних производителей (базы данных, серверы и т. д.);
- основные библиотеки (если того требует специфика);
- подпроекты, реализация (разработка) которых возможна как логически обособленных частей (Web, database, net, security, GUI etc.);
- оценка времени на реализацию этих частей;

Лабораторная работа №4. Автоматизированное функциональное тестирование Windows-приложений

Задание

1. Используя MS VisualStudio, и ее инструмент управления пакетами nuget создать проект для автоматизированного тестирования, подключить к нему библиотеку TestStack на языке C#.
2. Создать два тест кейса для тестирования программы калькулятор:

   Тест-кейс 1:
   Запустить приложение калькулятор
   Эмулируя нажатие пользователем мышкой на кнопки калькулятора выполнить операцию: 1111*1111=
   Считать полученный калькулятором результат, и проверить на правильность.

   Тест-кейс 2:
   Запустить приложение калькулятор
   Выполнить операцию 2222-1111=
   Считать полученный калькулятором результат, и сравнив его с заранее неверным ответом вызвать исключение.

   Т.о. тест-кейс 2 должен выдать ошибку.

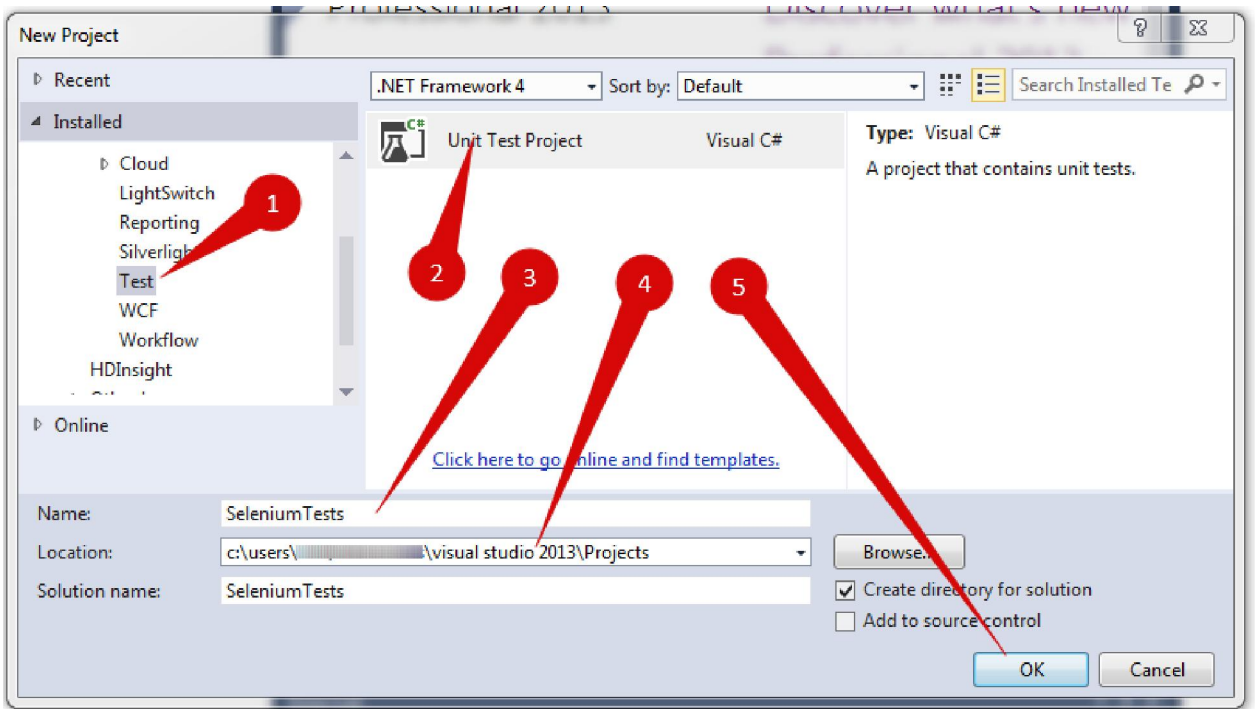
3. Выполнить один из предложенных вариантов.

Варианты заданий:

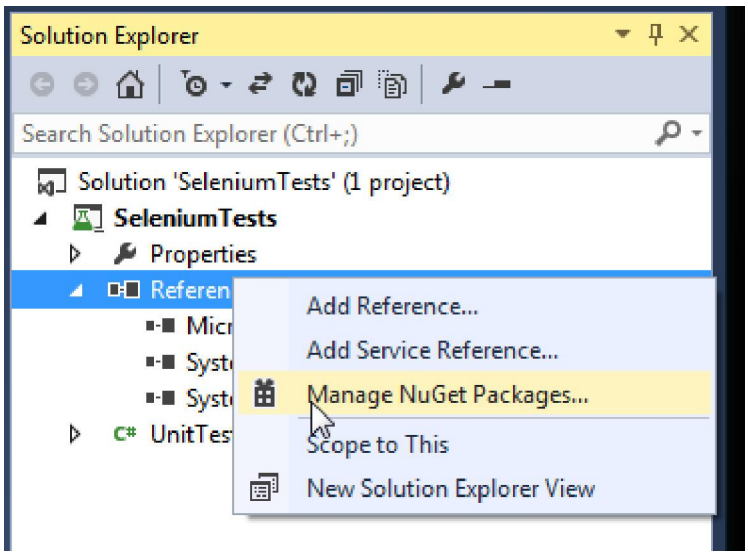
1. Тестирование программы Notepad, шаги:
   a. Открыть программу
   b. Ввести любой текст длинной 10-100 символов
   c. Сохранить файл на диск
   d. Закрыть программу
2. Тестирование программы Notepad, шаги:
   a. Открыть программу
   b. Открыть текстовый файл с диска
   c. Скопировать содержимое файла в буфер обмена
   d. Добавить содержимое буфера обмена к файлу
   e. Сохранить файл на диск
   f. Закрыть программу
3. Тестирование программы Paint:
   a. Открыть программу
   b. Нарисовать любую фигуру
   c. Сохранить файл на диск
   d. Закрыть программу

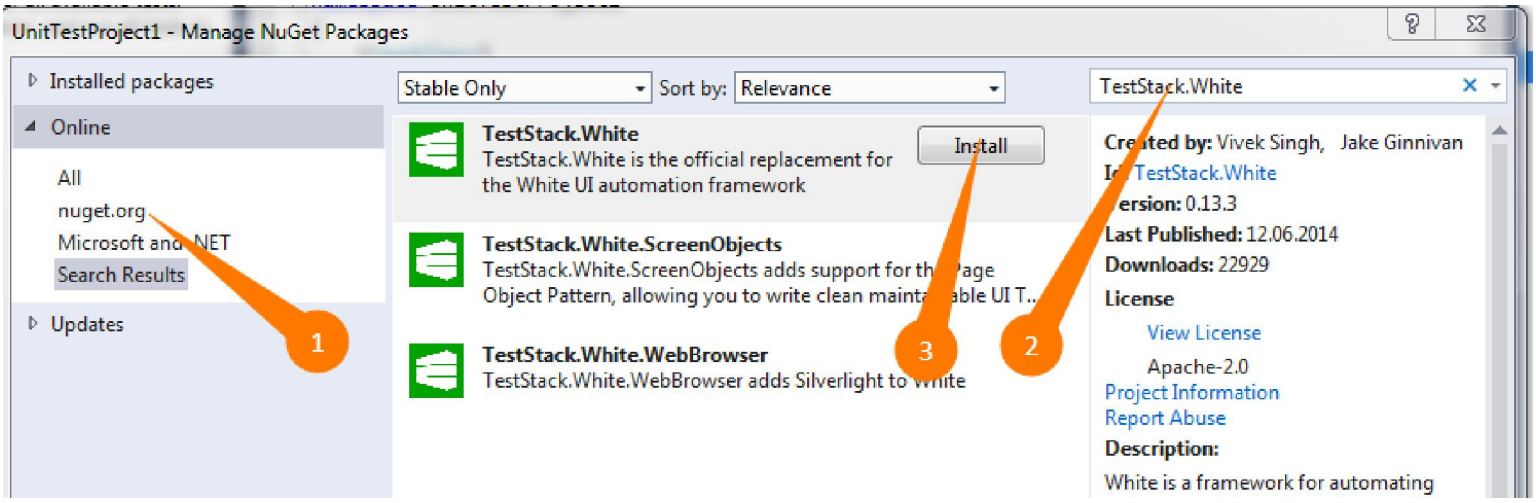
В помощь при выполнении:

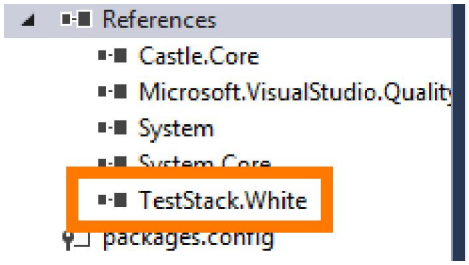
1. Создаем проект для тестов:



2. Добавляем в него библиотеку TestStack.White используя NuGet:







3. Создаем простое приложение:

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System.IO;
using TestStack.White;
using TestStack.White.UIItems.WindowItems;
using TestStack.White.Factory;
using TestStack.White.UIItems;
using TestStack.White.UIItems.Finders;
using System.Windows.Automation;

namespace Example
{
   [TestClass]
   public class UnitTest
   {
      private Window window;

      [TestInitialize]
      public void TestInitialize()
      {
         var applicationPath = Path.Combine(Environment.SystemDirectory, "calc.exe");
         Application application = Application.Launch(applicationPath);
         window = application.GetWindow("Calculator", InitializeOption.NoCache);
      }

      [TestMethod]
      public void TestMultiplyMethod()
      {

         window.Get<Button>("9").Click();
         window.Get<Button>("Multiply").Click();
         window.Get<Button>(SearchCriteria.ByText("3")).Click();
         window.Get<Button>("Equals").Click();

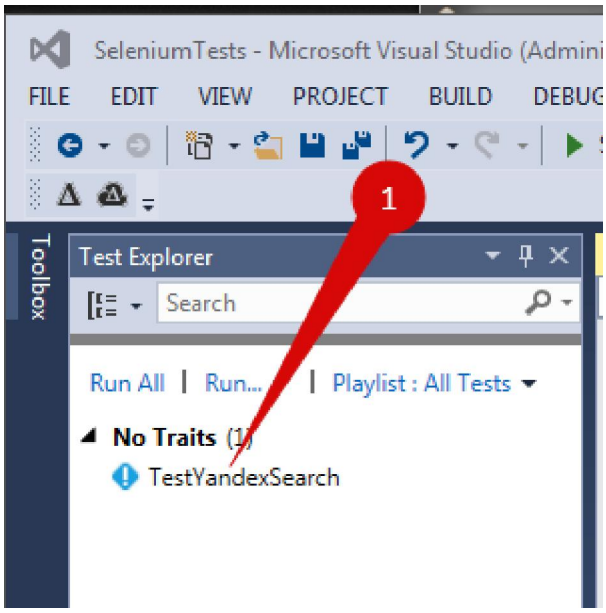
         SearchCriteria searchCriteria = SearchCriteria
            .ByAutomationId("158")
            .AndControlType(ControlType.Text);

         Label textBox = window.Get<Label>(searchCriteria);
```

```
            Assert.AreEqual("27", textBox.Text);
        }

        [TestCleanup]
        public void TestCleanup()
        {
            window.Dispose();
        }
    }
}
```

4. Запуск тестов происходит через меню Test Explorer VS



Лабораторная работа № 5. Автоматизированное функциональное тестирование WEB-сайтов

Задание:

1. Используя MS VisualStudio, и ее инструмент управления пакетами nuget создать проект для автоматизированного тестирования, подключить к нему библиотеку Selenium Web-driver. (По желанию, можно использовать любую другую среду/язык программирования совместимый с Selenium Web-driver).

2. Выполнить начальное задание:
    1. Запустить браузер
    2. выполнить навигацию на любой URL

3. Выполнить один из предложенных вариантов.

Варианты заданий:

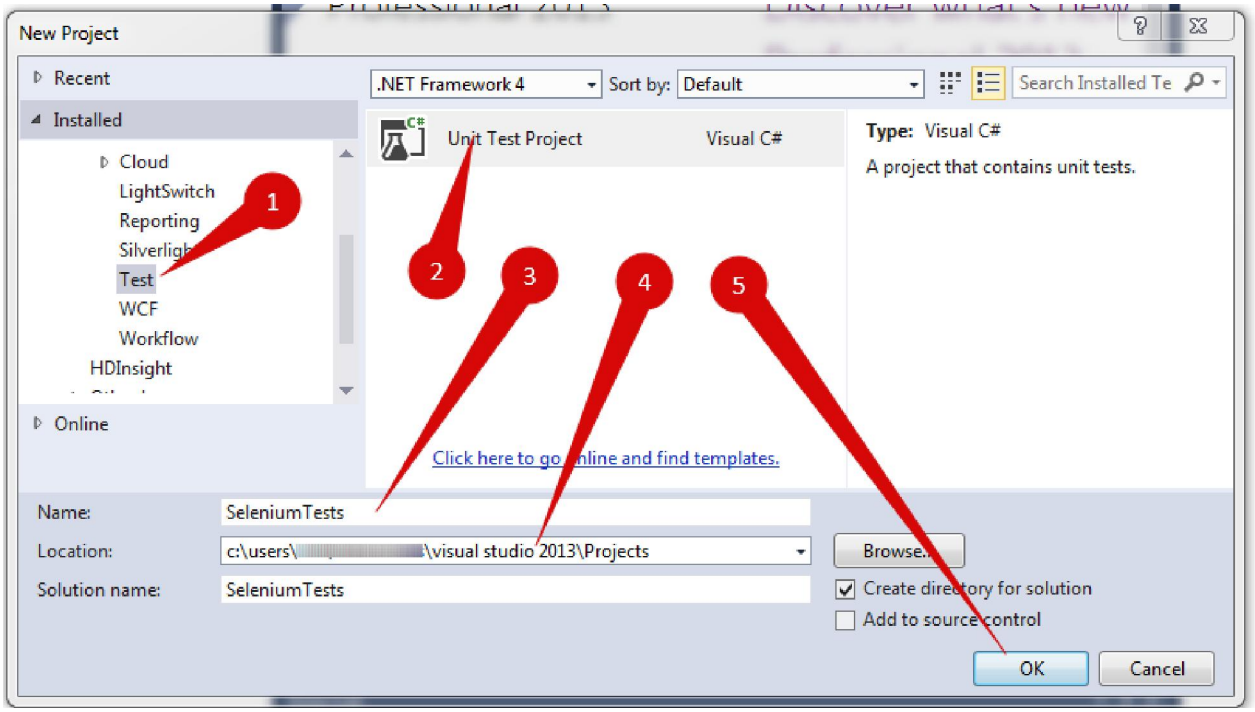
1. Поиск конкретных значений в выдаче поисковых сайтов:
    a. Выполнить поисковый запрос на поисковом сайте (например «шариковые ручки»
    b. Найти в выданных ответах конкретное значение (например «Parker»), вывести номер позиции в которой находится это конкретное значение.

2. Тестирование функциональности выбранного самостоятельно сайта: выберете любой сайт, с достаточно разветвленной функциональностью и напишите для него 5 тест-кейсов, в которых будут присутсвовать все перечисленные ниже случаи:
   i. Ввод текста в поле ввода
   ii. Нажатие на кнопку
   iii. Выбор элемента списка
   iv. Выбор элемента из радио-группы
   v. Поиск элемента на странице по имени класса
   vi. Поиск элемента на странице по типу тега

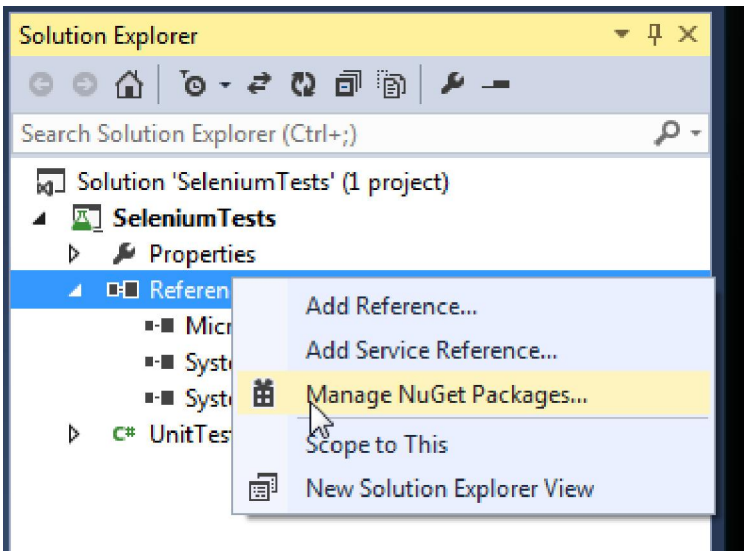
   Все тест кейсы должны быть независимыми друг от друга (т.е. каждый должен выполняться независимо от результата выполнения других тестов).

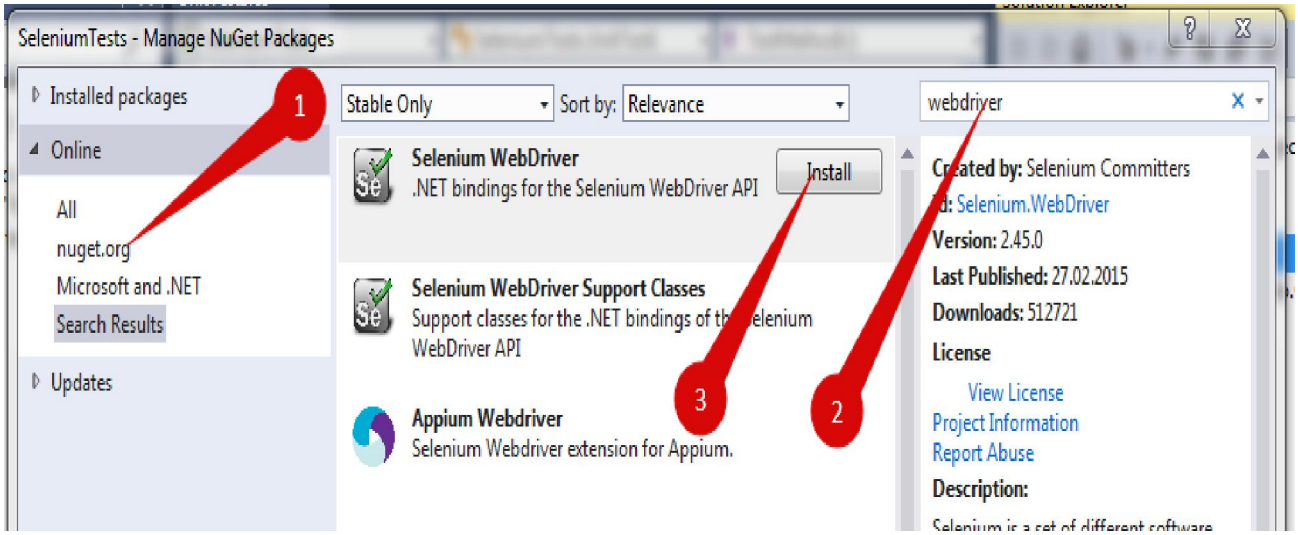
В помощь при выполнении:

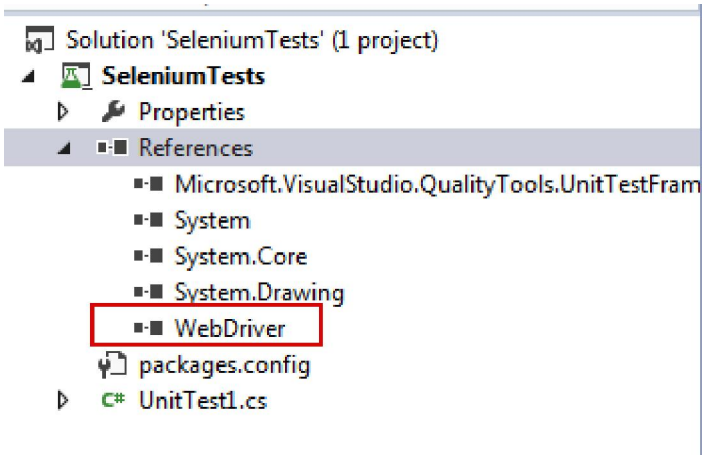
1. Создаем проект для тестов:



2. Добавляем в него Webdriver DLL используюя NuGet:







3. Создаем простое приложение:

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using System.Text;
using OpenQA.Selenium.Firefox;

namespace UnitTestProject1
{
    [TestClass]
    public class SeleniumTests
    {
        private IWebDriver driver;
        private string baseURL;

        [TestInitialize]
        public void TestInitialize()
        {
            driver = new FirefoxDriver();
            baseURL = "http://www.yandex.ru";
        }

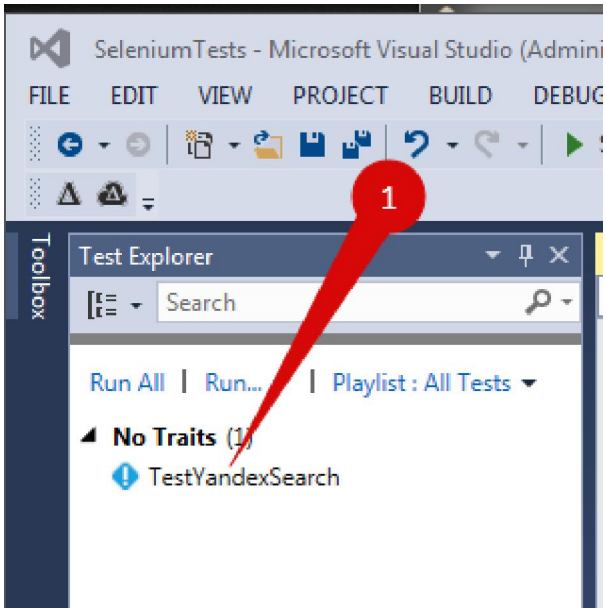
        [TestCleanup]
        public void TestCleanup()
        {
            try
            {
                driver.Quit();
```

```
            }
            catch (Exception)
            {
                // Ignore errors if unable to close the browser
            }
        }

        [TestMethod]
        public void TestYandexSearch()
        {
            driver.Navigate().GoToUrl(baseURL + "/");
            driver.FindElement(By.Id("text")).Click();
            driver.FindElement(By.Id("text")).Clear();
            driver.FindElement(By.Id("text")).SendKeys("Webdriver Sample");
            driver.FindElement(By.XPath("//button[@type='submit']")).Click();

            StringAssert.Contains(driver.Title, "Webdriver Sample");
        }
    }
}
```

4. Открываем Test Explorer VS и запускаем тест:



Лабораторная работа №6. Поиск и эксплуатация уязвимостей в программном обеспечении (Windows-приложения)

Задание: Исследовать внутреннее устройство и особенности поведения программы (по одному из предложенных вариантов), найти способы заставить программу вести себя неадекватным способом (с помощью инструментов внешнего управления программой).

Варианты:

1. Windows Калькулятор
2. Windows Notepad
3. Skype

Лабораторная работа №7. Поиск и эксплуатация уязвимостей в программном обеспечении (ВЕБ сайты)

Задание: с помощью средств внешнего управления браузером разработать:

1. Программу-робот, выполняющую регистрацию на любом ВЕБ сайте с формой регистрации. Учесть особенности (обойти) методы защиты от роботов.
2. Программу-робот, общающуюся через встроенный в какой-либо веб сайт чат, учесть особенности (обойти) защиту от СПАМ ботов.

Лабораторная работа №8. Continuous Integration – работа с системой автоматизации рутинных задач Jenkins CI

Задание:

1. Установить пакет Jenkins и настроить авторизированный вход.
2. Создать задачу, которая по расписанию, раз в 15 минут выполняет архивирование заранее заданного каталога на диске.

Лабораторная работа №9. Системы контроля версий – на примере Mercurial (по желанию Git)

Для выполнения лабораторной работы можно установить пакет Mercurial или Git локально, либо использовать любой из общедоступных серверов в интернете.
Mercurial для Windows: http://mercurial.selenic.com/release/windows/Mercurial-2.9.exe

Задание:
1. Изучить работу следующих команд (http://mercurial.selenic.com/wiki/UnderstandingMercurial):

Init

Add
Status
Forget
Commit
manifest
remove
rename
Diff

Log
Addremove

Clone
Push/pull
Incoming/outgoing
heads
parents
identify
Merge
resolve
update

2. Создать репозиторий, рабочую копию, создать в ней два файла и залить их в репозиторий. Для зачета представить созданный репозиторий, рассказать, что делает каждая из изученных команд, ответить на вопросы преподавателя.

2. Методические указания по организации самостоятельной работы студента

Самостоятельная работа студента является обязательным разделом основной образовательной программы подготовки бакалавра и направлена на овладение профессиональными умениями:

– понимание основ программной инженерии как методологии индустриального проектирования прикладных программных продуктов;

– способность к проведению поиска научно-технической литературы по конкретной тематике, как в периодических изданиях, так и в сети Интернет;

– навыки оформления отчетов по результатам выполненной работы, подготовки презентаций, четкого и логичного изложения своих мыслей.

Самостоятельная работа проводится в форме поиска, изучения и применения на практике (в процессе выполнения лабораторных работ) литературы, технической документации, опыта других разработчиков для достижения результата (выполнения задания лабораторной работы). К самостоятельной работе так же относится подготовка отчетов по лабораторным работам.

Темы для самостоятельной проработки

Понятие технологичности программного обеспечения.

Модули и их свойства.

Нисходящая и восходящая разработка программного обеспечения. Структурное и «неструктурное» программирование.

Средства описания структурных алгоритмов. Стиль оформления программы

Эффективность и технологичность

Программирование «с защитой от ошибок». Сквозной структурный контроль

OLE-механизм в Visual C++. Внедрение.

OLE-механизм в Visual C++. Связывание.

Active X-элемент. Свойства, методы, события.

Тематика индивидуальных заданий:

Разработка программы-сервера: сканер портов (с использованием различных библиотек – winsock, windows, socket, netinet и др.).

Список рекомендуемой литературы

Основная литература

1. Орлов С.А., Цилькер Б.Я., Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. – СПб.: Питер, 2012. – 608 с. (Экземпляров всего: 15) **ГРИФ**

2. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления: учеб. пособие для вузов. – М.: Горячая линия - Телеком, 2011 – 304 с.

3. Грекул В. И. Проектирование информационных систем. Курс лекций : Учебное пособие для вузов / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. - М. : Интернет-Университет Информационных Технологий, 2005. - 298[5] с.

4. Карпенков С.Х. Современные средства информационных технологий [Текст]: учеб. пособие для студентов высш. учеб. заведений, обучающихся по напр. подгот. дипломир. специалистов "Информатика и вычислит. техника" и "Информац. системы" / С. Х. Карпенков. 2-е изд. испр. и доп. - М.: КНОРУС, 2013. - 400 с.

5. Долженко А.И. Современные технологии программирования. Разработка приложений на базе WPF и Silverlight [Текст]: учеб. для студентов вузов, обучающихся по напр. 080800 "Приклад. информатика (по обл.)" и др. экон. спец. / А. И. Долженко; Рост. гос. экон. ун-т (РИНХ). - Электрон. изд. - Ростов н/Д: Изд-во РГЭУ (РИНХ), 2011. - 266 с.

6. Подбельский В.В. Курс программирования на языке Си [Текст]: учеб. для студентов высш. учеб. заведений, обучающихся по напр. "Бизнес-информатика" / В. В. Подбельский, С. С. Фомин. - М.: ДМК Пресс, 2012. - 384 с.

Дополнительная литература

1. Розенберг Д. Применение объектного моделирования с использованием UML и анализ прецедентов на примере книжного Internet-магазина : пер. с англ. / Д. Розенберг, К. Скотт. - М. : ДМК, 2002. - 158[2] с.

2. Российская Федерация. Законы. Федеральный закон "Об информации, информационных технологиях и о защите информации". - М. : Омега-Л, 2007. - 19 с.

3. Макконнелл, С. Совершенный код. Мастер-класс : практ. рук. по разраб. програм. обеспечения / [пер. с англ. под общ. ред. В. Г. Вшивцева]. - М.: Рус. Ред., 2007. - 896 с.

4. Оскерко, В. С., Пунчик, З. В. и др. Технологии баз данных : учеб. пособие для вузов. - Минск : Изд-во БГЭУ, 2007. - 171 с.

5. Эспозито, Д. Microsoft ASP.NET 2.0. Углубленное изучение: пер. с англ. : мастер-класс. - СПб. : Питер, 2007. - 592 с.

6. Алексеев, В. Е., Таланов, В. А. Графы и алгоритмы. Структуры данных. Модели вычислений: учеб. - М.: Интернет-Ун-т информ. технологий, 2006. - 320 с.

7. Подбельский В.В., Фомин С.С. Программирование на языке Си: учебное пособие для вузов. М.:Финансы и статистика, 2007, 600 с.

8. Электронный ресурс. - msdn.microsoft.com/

9. RSDN [Электронный ресурс]. – Режим доступа к сайту: http://www.rsdn.ru/ свободный.

10.Электронный ресурс. - http://programmersclub.ru/21/

11. Дополнительная литература 1 Основы высокопроизводительных вычислений: учебное пособие / К.Е. Афанасьев, С.Ю. Завозкин, С.Н. Трофимов, А.Ю. Власенко. - Кемерово: Кемеровский государственный университет, 2011. - Т. 1. Высокопроизводительные вычислительные системы. - 246 с. [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=232203

12. Основы высокопроизводительных вычислений: учебное пособие / К.Е. Афанасьев, С.В. Стуколов, В.В. Малышенко и др. - Кемерово: Кемеровский государственный универси-

тет, 2012. - Т. 2. Технологии параллельного программирования. - 412 с. [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=232204

13. Подбельский, В.В. Язык C#. Базовый курс: учебное пособие / В.В. Подбельский. - М.: Финансы и статистика, 2011. - 382 с. [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=86074

5 Технология программирования: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, М.П. Беляев, Ю.В. Минин; - Там- бов: Издательство ФГБОУ ВПО «ТГТУ», 2013. - 173 с.: [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=277802