**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
«Национальный исследовательский университет   
«Высшая школа экономики»**

Факультет компьютерных наук  
Департамент программной инженерии

Утверждаю  
Академический руководитель

образовательной программы

по направлению 09.03.04

«Программная инженерия»

В.В. Шилов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Программа дисциплины «Введение в программирование»

для направления 09.03.04 «Программная инженерия»   
подготовки бакалавра

Автор программы:  
**Максименкова О.В.**

Одобрена на заседании Департамента программной инженерии «\_\_\_» \_\_\_\_\_\_ 2017  г.

Руководитель Департамента \_\_\_\_\_\_\_\_\_\_\_\_\_\_С.М. Авдошин

Рекомендована Академическим советом образовательной программы

«Программная инженерия» «\_\_\_»\_\_\_\_\_\_\_\_ 2017 г.

Менеджер Департамента программной инженерии \_\_\_\_\_\_\_\_\_\_\_\_\_\_Т.В. Климова

Москва, 2016

*Настоящая программа не может быть использована другими подразделениями   
университета и другими вузами без разрешения департамента-разработчика программы.*

|  |  |
| --- | --- |
|  | Национальный исследовательский университет – Высшая школа экономики Программа дисциплины **«Введение в программирование»** для направления 09.03.04  «Программная инженерия» подготовки бакалавра |

# Область применения и нормативные ссылки

Настоящая программа учебной дисциплины "Введение в программирование" устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления 09.03.04 «Программная инженерия» подготовки бакалавра, изучающих дисциплину «Введение в программирование».

Программа разработана в соответствии с:

* Образовательным стандартом ФГАОУ ВПО «Национальный исследовательский университет «Высшая школа экономики»;
* Образовательной программой направления 09.03.04 «Программная инженерия» подготовки бакалавра.
* Рабочим учебным планом по направлению 09.03.04 «Программная инженерия» подготовки бакалавра, утвержденным в 2015 г.

# Цели освоения дисциплины

Цели освоения дисциплины "Введение в программирование":

* обеспечить студентов базовыми знаниями по информатике, алгоритмизации и программированию;
* заложить основы для последующих курсов, связанных с программированием;
* привить студентам навыки исследовательской работы, предполагающей самостоятельное изучение специфических инструментов и средств, необходимых для решения именно той конкретной проблемы, которая в качестве задачи поставлена перед ним.

# Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

* Знать:
  + - * + процедурную парадигму создания программных продуктов;
        + основные алгоритмические конструкции;
        + методы отладки структурных программ;
        + подходы к решению некоторых алгоритмических задач;
        + базовый синтаксис и семантику языка С#;
* Уметь:
  + - * + разрабатывать алгоритмы с использованием базовых алгоритмических конструкций;
        + проводить структурную декомпозицию задач;
        + составлять программный С# код, отвечающий заданному или разработанному алгоритму;
* Иметь навыки (приобрести опыт):
  + - * + разработки алгоритмов с использованием базовых алгоритмических конструкций и некоторых алгоритмов;
        + в решении типовых задач программирования с применением современного языка программирования и передовых инструментальных средств;

 В результате изучения дисциплины студент осваивает следующие компетенции:

1. Способен работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения научных и профессиональных задач (в том числе на основе системного подхода) (УК-5/СК-Б6).
2. Способен применять основные концепции, принципы, теории и факты, связанные с информатикой при решении научно-исследовательских задач (ПК-1/ИК-1).
3. Способен использовать методы и инструментальные средства исследования объектов профессиональной деятельности (ПК-3/ИК-3)
4. Способен читать, понимать и выделять главную идею прочитанного исходного кода, документации (ПК-11/ИК-11).
5. Способен моделировать, анализировать и использовать формальные методы конструирования программного обеспечения (ПК-12/ИК-12).
6. Способен применять основные методы и инструменты разработки программного обеспечения (ПК-17/ИК-17)

# Место дисциплины в структуре образовательной программы

Настоящая дисциплина является факультативной.

Изучение данной дисциплины базируется на знаниях студентами математики, основ информатики и алгоритмизации в рамках учебной программы средней школы базового уровня, умении применять математический аппарат при выборе метода решения задачи.

# Тематический план учебной дисциплины

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название темы | Всего часов по дисциплине | Аудиторные часы | | Самостоятельная работа |
| Лекции | Семинары и практические занятия |
| 1 | Алгоритмы и программы. Программы на языке C#. Комментирование кода. Типы данных в C#. Переменные. Операции. Организация ввода/вывода в консоль. | 10 | 2 | 2 | 6 |
| 2 | Основные алгоритмические конструкции. Линейный и ветвящийся алгоритм. Логические выражения. Условный оператор в C#, использование логических выражений в C# | 10 | 2 | 2 | 6 |
| 3 | Основные алгоритмические конструкции. Циклические алгоритмы. Реализация цикла с предусловием, цикла с постусловием и универсального цикла в C#. | 10 | 2 | 2 | 6 |
| 4 | Вложенные циклы. Вычисление конечных и бесконечных сумм. Решение задач с использованием циклов в C#. | 12 | 2 | 2 | 6 |
| 5 | Структурная декомпозиция. Статические методы в C#. Передача параметров в методы. Рекуррентные формулы. Рекурсия. | 12 | 2 | 2 | 8 |
| 6 | Массивы. Одномерные массивы. Передача массивов в методы. Решение задач с использованием одномерных массивов C#. | 12 | 4 | 2 | 6 |
| 7 | Массивы. Многомерные массивы и массивы массивов в С#. Решение задач с многомерными массивами. | 12 | 2 | 4 | 8 |
| 8 | Принципы разработки. Основы объектно-ориентированного программирования. Классы и объекты. Инкапсуляция. Полиморфизм. | 12 | 4 | 2 | 6 |
| 9 | Отношения между классами. Наследование. Виды ассоциативных связей. | 14 | 2 | 4 | 8 |
| 10 | Абстрактные типы данных. Последовательности. Списки. | 10 | 2 | 2 | 6 |
|  | **Итого по дисциплине** | 114 | 24 | 24 | 66 |

# Формы контроля знаний студентов

По дисциплине «Введение в программирование» предусмотрены следующие формы контроля:

* текущий контроль
  + лабораторные работы (Л).
* итоговый контроль
  + итоговая (экзаменационная) контрольная работа (ИК), выполняемая письменно или на компьютере, продолжительностью 80 минут.

Распределение контрольных мероприятий по модулям:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1 нед** | **2 нед** | **3 нед** | **4 нед** | **5 нед** | **6 нед** | **7 нед** | **8 нед** |
| **Модуль 1** |  |  |  |  |  | Л1 |  |  |
| **Модуль 2** |  | Л2 |  |  | Л3 |  |  | ИК |

## Методика формирования оценки за программную реализацию задачи

Оценка за письменную работу, выполняемую на компьютере в течении 60 минут, формируется в соответствии с критериями пункта 6.2.

## Критерии выставления оценки за программную реализацию задачи

При выполнении задачи на ЭВМ критерии делятся на две группы: основные и дополнительные. Основные критерии определяют нижний предел оценки по десятибалльной шкале в рамках соответствующей оценки по пятибалльной шкале. Дополнительные критерии определяют возможность повышения десятибалльной оценки

**1. Основные критерии**

**“ОТЛИЧНО”:**

***8 баллов***

1. Программа работает и полностью соответствует спецификации.

2. Студент может обосновать принятое решение.

3. Исходный текст документирован: назначение программы (условие задачи), номер учебной группы, фамилия и инициалы студента, дата выполнения, назначение используемых переменных, назначение и параметры определяемых программистом методов.

4. Программа остается работоспособной при вводе неверных исходных данных.

5. Предусмотрено повторное решение задачи без повторного запуска программы.

***9 баллов***

1. Программа соответствует критериям получения оценки 8 баллов.

2. Программа в целом соответствует дополнительным критериям.

***10 баллов***

1. Программа соответствует критериям получения оценки 8 баллов.

2. Программа полностью соответствует дополнительным критериям.

3. Студент может выполнить анализ альтернативных вариантов решения задачи.

**“ХОРОШО”:**

***6 баллов***

1. Программа в основном соответствует спецификации. Отклонения от спецификации допущены при реализации второстепенных подзадач.

2. Студент может объяснить принятое решение.

3. Исходный текст документирован.

***7 баллов***

1. Программа соответствует критериям получения оценки 6 баллов.

2. Программа в целом соответствует дополнительным критериям.

**“УДОВЛЕТВОРИТЕЛЬНО”:**

***4 балла***

1. Программа имеет отклонения от спецификации при решении основной задачи.

2. Студент может объяснить функционирование программы по ее исходному тексту.

3. Исходный текст документирован.

***5 баллов***

1. Программа соответствует критериям получения оценки 4 балла.

2. Программа соответствует отдельным дополнительным критериям.

**“НЕУДОВЛЕТВОРИТЕЛЬНО”:**

***1 балл:***

1. Разработка программы не завершена.

2. Программа имеет синтаксические ошибки.

***2 балла:***

1. Программа не соответствует спецификации.
2. Cтудент не может объяснить функционирование программы по ее исходному тексту.

***3 балла:***

1. Программа в основном соответствует спецификации.

2. Программа завершается аварийно при некоторых вариантах исходных данных.

3. Студент может объяснить функционирование программы по ее исходному тексту.

**2. Дополнительные критерии**

1. Исходный текст структурирован.

2. Предусмотрена обработка всех исключительных ситуаций.

3. Программа имеет средства изменения размерности и формы представления данных без изменения исходного текста или при минимальных изменениях (динамическое выделение памяти, именованные константы, и т.п.).

4. Программа реализована по модульному принципу и декомпозирована по реализуемым функциям.

5. В коде программы обоснованно использованы конструкции, изученные студентом самостоятельно, и их применение не противоречит основным требованиям, предъявляемым к решаемой задаче.

6. Предусмотрено само-документирование программы.

## Порядок формирования оценок по дисциплине

По всем видам работ выставляется десятибалльная оценка.

Оценка за лабораторную работу выставляется как средняя результатов взаимного оценивания. Одиночный результат взаимного оценивания формируется согласно оценочному листу для каждой лабораторной работы.

Итоговая оценка (ИО) по дисциплине «Введение в программирование» вычисляется по формуле:

**ИО = 0,6\*Н + 0,4\*ЭК**, где **Н**– оценка, накопленная за лабораторные работы первого и второго модулей, **ЭК** – оценка за итоговую (экзаменационную) контрольную работу.

**Н = 0,2\*Л1 + 0,4\*Л2 + 0,4\*Л3**

*Округление оценок при вычислениях осуществляется до ближайшего целого.*

# Содержание дисциплины

## Содержание лекций

**Тема 1.**

Понятие типа данных. Типы данных в C#. Переменные. Приведение типов. Операторы ввода и вывода в консольное окно.

Литература по теме 1:

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. С# 5.0 Language specification. Version 5.0 : Microsoft Corporation, 2012

**Тема 2.**

Логический тип данных. Логические выражения. Ветвящиеся алгоритмы. Условный оператор в C#. Тернарная операция.

Литература по теме 2:

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 3.**

Трассировочные таблицы первого и второго видов. Циклические алгоритмы. Циклы с предусловием и постусловием в C#.

Литература по теме 3

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 4.**

Понятие о рекуррентных последовательностях. Рекуррентные формулы. Универсальный цикл. Последовательность Фибоначчи.

Литература по теме 4

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 5.**

Алгоритмы со вложенными циклическими конструкциями. Понятие о конечной и бесконечной суммах.

Литература по теме 5

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 6.**

Операторы break и continue. Понятие о массиве. Одномерные массивы в C#. Создание массивов, заполнение массивов, обработка элементов массивов, преобразование массивов.

Литература по теме 6

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 7.**

Массивы символов. Строки в C#. Методы обработки строк. Посимвольная обработка строк. Сравнение строк.

Литература по теме 7

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 8.**

Понятие о матрицах. Свойства индексов элементов диагоналей квадратных матриц. Массивы массивов в С#. Многомерные массивы в С#. Представление матриц многомерными массивами и массивами массивов.

Литература по теме 8

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

**Тема 9.**

Понятие об абстракции и инкапсуляции. Понятие структурной декомпозиции и структурного проектирования программных систем. Подпрограммы. Статические методы C#. Параметры методов.

Литература по теме 9

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
3. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
4. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012
5. J. Farrell. Microsoft Visual C# 2010: An introduction to object-oriented programming. 4-th edition, 2011

**Тема 10.**

Понятие рекурсивных функций в математике и программировании.

Литература по теме 10

1. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989
2. Головешкин В.А., ульянов М.В. Теория рекурсии для программистов. – М.:Физматлит, 2006
3. Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011
4. J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
5. K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012

## Содержание практических занятий

Примерный перечень тем практических занятий:

1. Подходы к решению задач, связанных с вычислением конечных и бесконечных сумм и произведений.
2. Задачи, связанные с программированием рекуррентных формул.
3. Особенности решения задач, вычисляющих суммы.
4. Особенности программирования задач, связанных со знакопеременными рядами и суммами.
5. Подходы к решению задач, связанных с рекурсией.
6. Подходы к решению задач, требующих применения массивов.
7. Структурная декомпозиция.
8. Объектовая декомпозиция.

# Образовательные технологии

В курсе реализованы следующие технологии активного подхода к обучению:

* формирующий контроль [*formative assessment*] и использование результатов текущего контроля для воздействие на учёбу и настройку учебного курса. Предоставление информативной обратной связи [*feedback*] с указанием сильных и слабых сторон работы студента.
* взаимное оценивание [*peer assessment*] – оценивание учащимися лабораторных работ друг друга для выработки навыков чтения и анализа чужого кода.

Лекционные занятия проводятся во фронтальном режиме.

Работа на семинарах представляет собой комбинацию образовательных технологий. В зависимости от восприятия аудиторий учебного материала применяется фронтальная работа, самостоятельная индивидуальная работа студентов, самостоятельная групповая работа студентов.

# Оценочные средства для текущего контроля аттестации студента

## Тематика заданий текущего контроля

### Лабораторные работы

Лабораторные работы проводится по материалам нескольких лекций и семинаров и включают в себя дополнительный материал для самостоятельного освоения студентами.

Каждая лабораторная работа состоит из задания или набора заданий, которые в анонимизированном виде оцениваются несколькими однокурсниками.

*Примерные задания*

1. Задачи из книги Меньшиков Ф.В. Олимпиадные задачи по программированию. – Спб.: Питер, 2006. – 315 с.):
   1. Пользователь с клавиатуры вводит два целых неотрицательных числа *М* и *N* (0 ≤ *M*, *N* ≤ 1030 000)*.* Найти их сумму.
   2. Пользователь с клавиатуры вводит целое неотрицательное число *N* – количество прямоугольников со сторонами, параллельными осям координат. Требуется определить площадь фигуры, образованной объединением данных прямоугольников.
2. Написать программу, используя шаблон консольного приложения в Visual Studio 2015 и выше.

*Цель программы – последовательная обработка массива целочисленных массивов при помощи набора методов*.

**В программе используется ровно один массив**.

В методе **Main()** сформировать массив целочисленных массивов, представляющий квадратную матрицу размером **NxN** (целое число **N<20** получить от пользователя). Заполнить матрицу случайными числами из диапазона [0;10).

Программа последовательно запускает методы обработки матрицы и выводит результаты. Перед переходом к следующему шагу программа просит пользователя нажать клавишу Enter для продолжения.

Вывод результатов на экран осуществляется при помощи метода

* **void Print(arr)** – выводит массив массивов **arr** в табличной форме.

Операции над массивом массивов определяются следующими методами:

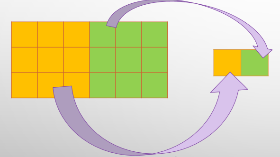
* **bool IsSparce(arr)** – проверяет является ли квадратная матрица, представленная массивом массивов **arr**, разряжённой, то есть количество нулей **K** в ней больше или равно (**N\*N/2).**
* **void DeleteZeros(arr)** – удаляет из каждого целочисленного массива в **arr** нулевые элементы со сдвигом влево, количество элементов при этом уменьшается. Если все элементы нули, то ссылка на соответствующий массив в **arr** заменяется на **null**.

## Итоговая контрольная работа

Итоговая контрольная работа состоит в самостоятельном написании программного кода. Задачи могут включать в себя темы первого модуля.

*Примерное задание контрольной работы*:

Создать квадратную целочисленную матрицу размера **3x6** и заполнить ее случайными числами из диапазона [**0; 20**]. По матрице сформировать целочисленный массив из двух элементов, значениями которых являются определитель левой и правой подматриц размера **3x3**, соответственно (см. рисунок).



## Тематика заданий итогового контроля

Итоговая контрольная работа состоит в самостоятельном написании программного кода. Задачи могут включать в себя любые темы курса.

**Примерное задание контрольной работы**:

Метод **ArrayForm()** формирования массива неотрицательных целых чисел и заполнения его «сериями». В качестве параметров передаются границы диапазона заполнителей и количество значений в каждой серии, отрицательное число или нулевое число означает отсутствие значений данной серии.

Например, выполнение **ArrayForm(1,4,2,2,-1,3)** должно формировать массив: **1 1 2 2 4 4 4**, а выполнение **ArrayForm(10,7,2,2,-1,3)** должно формировать массив: **10 10 9 9 7 7 7**

Метод вывода на экран массива целых неотрицательных чисел.

В основной программе получить от пользователя границы диапазона значений и строку с количествами элементов каждой серии, формат строки: числа, разделённые символом точка с запятой. Сформировать массив и вывести его на экран.

*Предусмотреть обработку корректности ввода данных и цикл повторения решения*.

# Учебно-методическое и информационное обеспечение дисциплины

**10.1 Базовый учебник –** Подбельский В.В. Язык C#. Базовый курс. М.: Финансы и статистика, 2011.

## Основная литература:

* Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке С#. – СПб.: Питер, 2016.
* C# 5.0. Language Specification. Version 5.0.: Microsoft Corporation, 2012.
* H. Abelson, G.J. Sussman, J. Sussman, Structure and Interpretation of Computer Programs, MIT Press, 2008
* J. Albahari, B. Albahari, C# 5.0 in a nutshell, 5-th edition, 2012
* J. Sharp, Microsoft Visual C# 2012. Step by step, 2012
* K. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel. Beginning Visual C# 2012. Programming, 2012
* J. Farrell. Microsoft Visual C# 2010: An introduction to object-oriented programming. 4-th edition, 2011
* Троелсен Э. Язык программирования С# 5.0 и платформа .NET 4.5. М.: Вильямс, 2014.
* Шилдт Г. С# 4.0. Полное руководство. М.: Издательский дом "Вильямс", 2011.
* Гросс Кр. C# и платформа .NET 3.5 Framework: базовое руководство.2-е изд. М.: ООО "И. Д. Вильямс", 2009.
* Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. – М.:Физматлит, 2006

## Дополнительная литература и источники

* Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989.
* Пауэрс Л., Снелл М. Microsoft Visual Studio 2008. СПб.: БХВ-Петербург, 2009.
* Richter J. CLR via C#. 4-th Edition. Redmond: Microsoft Press, 2012.
* Нейгел К. и др. C# 4.0 и платформа .NET 4.0 для профессионалов. М. : ООО "И.Д. Вильямс", 2011.
* Бердж В. Методы рекурсивного программирования. – М.: Машиностроение, 1983
* Баррон д. Рекурсивные методы в программировании. – М.: «МИР», 1974.
* Меньшиков Ф.В. Олимпиадные задачи по программированию. – Спб.: Питер, 2006. – 315 с.
* Jesse Dunietz, Geza Kovacs, and John Marrero. *6.096 Introduction to C++, January IAP 2011*. (Massachusetts Institute of Technology: MIT OpenCourseWare),[http://ocw.mit.edu](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011) (Accessed 21 Jun, 2015). License: [Creative Commons BY-NC-SA](http://creativecommons.org/licenses/by-nc-sa/4.0/)
* Tom Lieber, Kyle Murray, and Frank Li. *6.S096 Introduction to C and C++, January IAP 2013*. (Massachusetts Institute of Technology: MIT OpenCourseWare), [http://ocw.mit.edu](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-s096-introduction-to-c-and-c-january-iap-2013) (Accessed 21 Jun, 2015). License: [Creative Commons BY-NC-SA](http://creativecommons.org/licenses/by-nc-sa/4.0/)

## Справочники, словари, энциклопедии

* MSDN (URL: http://msdn.microsoft.com/en-US/)

## Программные средства

* Microsoft Visual Studio 2015 и выше или Visual Studio Teams Services (VSTS).
* Microsoft Office 365 Education

# Материально-техническое обеспечение дисциплины

* Проектор для проведения лекций и семинаров
* Классы для семинаров с компьютерами, на которых установлена IDE Microsoft Visual Studio 2015 и выше, есть возможность доступа в Интернет.