Темы урока

[**Знакомство**](#_1nbaqgkx9q55) **1**

[**О курсе**](#_znkic6381v5) **3**

[Программа курса](#_ip4j4ltzl7x8) 3

[План уроков](#_c7n9hheqwqdq) 3

[Домашние задания](#_9uos4zm9iy56) 3

[Перспективы после окончания](#_cbup0pzjw85) 3

[**C# и .NET**](#_rn40trywlsby) **3**

[**Средства разработки**](#_rr3wnobhwntw) **4**

[**Стадии разработки ПО**](#_83y1n421vqtu) **4**

[**Простые команды ввода/вывода**](#_6pzi3spl5blz) **4**

[Работа с консолью.](#_mtkqnv6szxtb) 4

[**Самостоятельная работа**](#_9g6kclk62jmp) **4**

[**Введение в Git, GitHub**](#_bw70t4bs63lu) **6**

[Инструкция](#_wpd2lprjvefc) 6

[В конце урока](#_h0cmqdvy9be0) 6

[Дома](#_pqpodhn75ky) 6

[**Домашнее задание**](#_h5sagbglk9hz) **7**

# Знакомство

Представиться, рассказать о себе.

Меня зовут Андрей.

Обладаю 15-летним опытом работы в сфере IT и программирования на языках C#, SQL, JavaScript.

С 2006 работаю в компании CNET Content Solutions – филиале компании CBS Interactive, входящей в международный медиа-холдинг CBS.

Прошёл карьерный путь в этой компании от разработчика до руководителя сектора разработки.

Участвовал в десятке крупных проектов. Сейчас управляю несколькими командами по разработке корпоративных продуктов.

Слежу за развитием методик и инструментов для разработки и управления циклом разработки программных решений, посещаю профильные конференции.

Попросить представиться студентов, рассказать об ожиданиях от курса.

# О курсе

4 месяца (36 занятий или 144 часа)

Обучение инструментам и методам разработки с нуля

Широкий спектр практик для решения различных задач

Каждые два месяца завершаются выполнением проекта

Сертификат и уверенность при поиске работы

## Программа курса

* Синтаксис С#
* Алгоритмы
* Объектно-ориентированное программирование
* Архитектура
* Современный C#
* ASP.NET Web API 2
* Проектирование баз данных (Microsoft SQL Server)
* Облачная инфраструктура Microsoft Azure

## План уроков

Уроки будут делиться на 2 части - примерно по полтора часа. В течение урока я буду читать теоретическую часть, показывать примеры, студенты будет выполнять самостоятельную работу для закрепления материала.

## Домашние задания

Я буду просить выкладывать домашние задания на GitHub чтобы было удобно смотреть исправления, сделанные после замечаний.

## Перспективы после окончания

После прохождения курса студенты будут достаточно подкованы для прохождения собеседований на позицию Junior Software Developer C# / .NET

# C# и .NET

* Язык C#, CLI, другие .NET языки
* Платформы .NET, .NET Framework и .NET Core

# 

# Средства разработки

* Visual Studio
* VisualStudio Code

# Стадии разработки ПО

Постановка задачи

Проектирование

Кодирование

Отладка / тестирование

Сопровождение

# Базовые команды ввода/вывода

## Работа с консолью.

Самые простые команды в любом языке программирования обычно касаются работы с консолью.

В .NET для это существует класс System.Console (на самом деле это статический класс, но что это такое, мы обязательно узнаем чуть позже; пока можно запомнить статические классы, как объекты у которых есть полезные функции).

|  |
| --- |
| Console.WriteLine("Вывести на экран текст и перейти на новую строку.");  Console.WriteLine("Вывести на экран текст.");  // прочитать строку, введенную пользователем и сохранить в переменной string line = Console.ReadLine(); |

## Проблемы с русскими символами

Если есть проблемы с вводом/выводом русских символов, можно воспользоваться командами предварительной установки правильной кодировки для консоли:

|  |
| --- |
| Console.InputEncoding = Encoding.Unicode; Console.OutputEncoding = Encoding.Unicode; |

Небольшой ликбез на тему “почему так?”

Если у вас международная или американская версия Windows, возможно, по-умолчанию для консоли выставлена кодировка 437 - English (United States).

это можно проверить используя команду **chcp** (Change code page).

# Самостоятельная работа

Разбор всех стадий на примере задачи со следующими условиями:

* Необходимо узнать имя, а потом вывести ему приветствие.
* Перед вводом приветствия должно пройти 5 секунд
* Программа должна завершиться, когда пользователь подтвердит прочтение приветствия нажатием любой клавиши на клавиатуре.

Базовые элементы алгоритмических блок-схем:

* Терминатор
* Процесс
* Ввод-вывод

# 

# Введение в Git, GitHub

Рассказать про распределенную систему контроля версий

## Инструкция

### В конце урока

* Регистрируемся на GitHub <https://github.com>
* Создаем публичный репозиторий nordic-it-cs-q1
* Ставим локальный Git для Windows <https://gitforwindows.org>
* Запускаем Git bash
* git clone URL-на-репозиторий.git
* Создаем структуру папок для уроков:
  + 01 (02, 03, и т.д.)
    - ClassWork
    - HomeWork
* Копируем в корневую папку репозитория (nordic-it-cs-q1) файл .gitignore для Visual Studio / C# <https://github.com/github/gitignore/blob/master/VisualStudio.gitignore>
* Добавляем файлы с урока в папку 01/ClassWork
* git status
* git add --all
* git status
* git commit -m “Class work of the 1st lesson added”
* git push

### Дома

* Делаем git clone URL-на-репозиторий.git
* Создаем новый солюшн в папке 01/HomeWork/
* Выполняем там домашнюю работу
* git add --all
* git status
* git commit -m “Home work of the 1st lesson added”
* git push
* В личном кабинете присылаете мне ссылку на на свой GitHub-репозиторий, а также возможные комментарии и вопросы.
* Если что-то не получится, присылайте вопросы, **в самом крайнем случае** просто zip-файл с кодом решения.

# Домашнее задание

1. Модифицировать программу таким образом, чтобы после вывода приветствия программа ожидала ещё 5 секунд и выводила прощание, а уже потом ожидала нажатия клавиши и завершалась.
2. Запушить изменения на GitHub.