

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**УТВЕРЖДАЮ**

Проректор по учебной работе и  
образовательным инновациям

О.Н. Здрок

2020 г.

Регистрационный № УД- 8565/уч.



**Язык С# и базовые технологии .NET**

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности:**

**1-31 03 08 – Математика и информационные технологии (по направлениям)**

**Направление специальности:**

**1-31 03 08-01 – Веб-программирование и интернет-технологии**

**1-31 03 08-02 – Математическое и программное обеспечение мобильных устройств**

2020 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 03 08-2014 от 09.07.2014 и учебных планов № G313-197/уч., G313-199/уч., G313-198/уч., G313-200/уч. от 30.05.2014 г.

**СОСТАВИТЕЛИ:**

А.И. Кравчук, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент.

**РЕЦЕНЗЕНТЫ:**

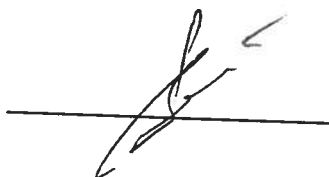
О.Г. Смолякова, доцент кафедры программного обеспечения и информационных технологий БГУИР, кандидат технических, доцент.

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой веб-технологий и компьютерного моделирования БГУ  
(протокол № 9 от 20.05.2020 г.)

Научно-методическим Советом БГУ  
(протокол № 5 от 17.06.2020 г.)

Заведующий кафедрой



В.М. Волков

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. Язык программирования C# был создан специально для работы с платформой .NET, которая

- обеспечивает возможность взаимодействия с существующим программным кодом;
- позволяет поддерживать многочисленные языки программирования (C#, Visual Basic, F#, S# и т.д.);
- имеет общий исполняющий механизм, используемый всеми поддерживающими .NET языками;
- характеризуется полной интеграций языков – поддерживается межъязыковое наследование, обработка исключений и отладка кода;
- имеет обширную библиотеку базовых классов, позволяющую избегать сложностей, связанных с выполнением прямых вызовов к API-интерфейсу, и предлагает согласованную объектную модель, которую могут использовать все поддерживающие .NET языки.

Наличие перечисленных функциональных возможностей делает платформу .NET сверхмощным инструментом для создания различного рода приложений: от небольших десктопных до мобильных кроссплатформенных приложений, а также крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей. Разработчики программного обеспечения на основе платформы .NET в настоящий момент являются востребованными, что делает платформу и язык C# актуальными для изучения студентами.

### **Цели и задачи учебной дисциплины**

**Цель** учебной дисциплины – изучение языка C# и базовых .NET технологий необходимых для разработки различных приложений корпоративного уровня.

### **Задачи учебной дисциплины:**

1. дать теоретическую подготовку по языку C#;
2. дать теоретическую подготовку по платформе .NET;
3. дать практические навыки создания программного обеспечения на примере разработки библиотек классов и консольных приложений;
4. познакомить с принципами модульного тестирования разрабатываемого программного продукта;
5. научить правильному построению архитектуры разрабатываемых приложений в соответствии с принципами объектно-ориентированного программирования;
6. научить пользоваться сторонними программами и инструментами в процессе разработки реальных программных продуктов.

**Место учебной дисциплины** в системе подготовки специалиста с высшим образованием.

Дисциплина «Язык С# и базовые технологии .NET» относится к циклу дисциплин специализации компонента учреждения высшего образования.

**Связи** с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

Изучение дисциплины «Язык С# и базовые технологии .NET» базируется на знаниях, полученных при изучении университетских курсов по компьютерным сетям, базам данных и современных языков программирования (C++, Java).

#### **Требования к компетенциям**

Освоение учебной дисциплины «Язык С# и базовые технологии .NET» должно обеспечить формирование следующих академических, социально-личностных и профессиональных компетенций:

*академические* компетенции:

- АК-4. Уметь работать самостоятельно.
- АК-5. Быть способным порождать новые идеи (обладать креативностью).
- АК-6. Владеть междисциплинарным подходом при решении проблем.
- АК-7. Иметь навыки, связанные с использованием технических устройств, управлением информацией и работой с компьютером.
- АК-8. Обладать навыками устной и письменной коммуникации.
- АК-9. Уметь учиться, повышать свою квалификацию в течение всей жизни.

*профессиональные* компетенции:

- ПК-1. Разрабатывать практические рекомендации по использованию научных исследований, планировать и проводить экспериментальные исследования, исследовать патентоспособность и показатели технического уровня разработок программного обеспечения информационных систем.
- ПК-2. Владеть основными методами, способами и средствами получения, хранения, переработки информации. Применять современные методы проектирования информационных систем, использовать веб-сервисы, оформлять техническую документацию.
- ПК-22. Работать с научной, технической и патентной литературой.

В результате освоения дисциплины студент должен:

**знать:**

- язык программирования С#;

- основные концепции платформы .NET и ее базовой части BCL;

**уметь:**

- создавать и вызывать методы;
- перехватывать, обрабатывать и выбрасывать исключения;
- создавать и использовать новые типы (перечисления, классы и структуры) и понимать различия между ссылочными и значимыми типами данных;
- выполнять простые операции для взаимодействия с файловой системой;
- контролировать область видимости и время жизни членов типа;
- использовать наследование для разработки новых ссылочных типов данных;
- управлять временем жизни объектов и контролировать использование ресурсов;
- определять свойства и индексы для инкапсулирования данных и определять операторы для этих данных;
- отделять действие от метода, реализующего это действие, и использовать отделенные операции для обработки асинхронных событий;
- использовать коллекции для объединения данных и использовать обобщения для реализации типизированных классов, структур, интерфейсов и методов;
- реализовывать собственные классы коллекций, поддерживающие возможность перебора элементов;
- создавать запросы к данным, находящимся в памяти, с использованием LINQ;
- выполнять первичное тестирование проекта на этапах разработки;
- планировать и внедрять архитектуру проекта в соответствии с принципами ООП;
- выполнять тестирование проекта на этапах разработки;
- планировать и внедрять архитектуру проекта в соответствии с принципами объектно-ориентированного программирования.

**владеть:**

- навыками работы в IDE Microsoft Visual Studio 2019/JetBrains Rider 2020.

**Структура учебной дисциплины**

Дисциплина «Язык C# и базовые технологии .NET» изучается в 8-9 семестрах заочной формы получения высшего образования. Всего на

изучение учебной дисциплины отведено 70 часов, из них 10 аудиторных часов, в том числе 6 часов лекций, 4 часа лабораторных занятий.

Трудоемкость учебной дисциплины составляет 1 зачетная единица. Форма текущей аттестации по учебной дисциплине – зачет.

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

**Тема 1. Введение в платформу .NET. Основные программные конструкции языка C#.** Знакомство с платформой .NET. Создание проектов с Visual Studio 2019. Типы проектов Visual Studio 2019. Отладка исходного кода, документирование приложения. Объявление переменных и присваивание значений. Использование выражений и операций языка в C#. Операторы. Создание и использование массивов.

**Тема 2. Создание новых типов.** Определение и вызов методов. Спецификация необязательных и выходных параметров. Создание и использование перечислений. Создание и использование классов. Создание и использование структур. Сравнение ссылочных типов с типами значений.

**Тема 3. Инкапсуляция. Наследование классов. Интерфейсы и абстрактные классы.** Управление видимостью членов типа. Свойства и индексаторы. Статические методы и данные. Перегрузка операций. Использование наследования для определения новых ссылочных типов. Определение и реализация интерфейсов. Определение абстрактных классов.

**Тема 4. Обработка исключений.** Обработка исключений. Возникновение исключений.

**Тема 5. Делегаты и события.** Объявление и использование делегатов. Использование лямбда-выражений.

**Тема 6. Коллекции и обобщенные типы.** Использование коллекций. Создание и использование обобщенных типов. Определение обобщенных интерфейсов. Использование обобщенных методов и делегатов.

# УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

## Заочная (полная и сокращенная) форма получения образования

| Номер раздела, темы | Название раздела, темы                                                  | Количество аудиторных часов |                      |                     |                      |      |                      |            | Литература                            | Форма контроля знаний |
|---------------------|-------------------------------------------------------------------------|-----------------------------|----------------------|---------------------|----------------------|------|----------------------|------------|---------------------------------------|-----------------------|
|                     |                                                                         | Лекции                      | Практические занятия | Семинарские занятия | Лабораторные занятия | Иное | Количество часов УРС |            |                                       |                       |
| 1                   | 2                                                                       | 3                           | 4                    | 5                   | 6                    | 7    | 8                    | 9          | 10                                    |                       |
| 1                   | Введение в платформу .NET. Основные программные конструкции языка C#    |                             |                      |                     |                      |      |                      | [1-5]      | Опрос, отчет по лабораторным заданиям |                       |
| 2                   | Создание новых типов                                                    | 1                           |                      |                     | 1                    |      |                      | [1-7]      | Опрос, отчет по лабораторным заданиям |                       |
| 3                   | Инкапсуляция<br>Наследование классов.<br>Интерфейсы, абстрактные классы | 2                           |                      |                     | 1                    |      |                      | [1-5]      | Опрос, отчет по лабораторным заданиям |                       |
| 4                   | Обработка исключений                                                    | 1                           |                      |                     |                      |      |                      | [1-5], [6] | Опрос                                 |                       |
| 5                   | Делегаты и события                                                      | 1                           |                      |                     | 1                    |      |                      | [1-5], [6] | Опрос, отчет по лабораторным заданиям |                       |
| 6                   | Коллекции и обобщенные типы                                             | 1                           |                      |                     | 1                    |      |                      | [1-5]      | Опрос, отчет по лабораторным заданиям |                       |
|                     | ИТОГО:                                                                  | 6                           |                      |                     | 4                    |      |                      |            |                                       |                       |

## **ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**

### **Основная литература**

1. C# documentation. Microsoft Documentation (<https://docs.microsoft.com/en-us/dotnet/csharp/>).
2. Полное руководство по языку программирования C# 8.0 и платформе .NET Core 3 (<https://metanit.com/sharp/tutorial/>).
3. C# Notes for Professionals book. Download pdf (<https://books.goalkicker.com/CSharpBook/>).
4. Programming C# 8.0. by Ian Griffiths. O'Reilly Media. 2019 – 801 p.
5. C# 8.0 in a Nutshell. by Joseph Albahari, Eric Johanssen. 2020. – 1104 p.

### **Дополнительная литература**

6. C# in Depth. by Jon Skeet. Fourth Edition. 2019 – 528.
7. Essential C# 7.0, by Mark Michaelis. Sixth Edition. Addison-Wesley Professional. 2018 – 900 p.

### **Перечень рекомендуемых средств диагностики и методика формирования итоговой оценки**

Оценка за ответы на лекциях (опрос) и лабораторных занятиях может включать в себя полноту ответа, наличие аргументов, примеров из практики и т.д.

При оценке лабораторных работ студента необходимо обращать внимание на: актуальность проекта, содержание и степень развития проекта, оригинальность подхода к решению проблемы, корректность используемых методов, практикоориентированность полученных результатов, самостоятельность и аргументированность действий, свидетельства качества работ студента (отзывы, рецензии, оценки, показатели аналитика проекта т.д.), наличие самооценки студентом собственных работ (рефлексивные замечания, реплики, комментарии, проекты изменений).

Формой текущей аттестации по дисциплине «Язык C# и базовые технологии .NET» учебным планом предусмотрен зачет.

При формировании итоговой оценки используется рейтинговая оценка знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая оценка предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

**Примерные** весовые коэффициенты, определяющие вклад текущего контроля знаний и текущей аттестации в рейтинговую оценку:

Формирование оценки за текущую успеваемость:

- ответы на аудиторных занятиях – 10 %;
- каждая из лабораторных работ по темам имеет равный вес, суммарный вес всех лабораторных работ равен 90 %.

Рейтинговая оценка по дисциплине рассчитывается на основе оценки текущей успеваемости и оценки на зачете с учетом их весовых коэффициентов. Вес оценки по текущей успеваемости составляет 40 %, оценка на зачете – 60 %.

### **Примерная тематика лабораторных занятий**

#### **Лабораторное занятие 1. Создание новых типов**

Студентам рекомендуется выполнить следующие задания:

*Задание 1.*

- Реализовать класс Transformer, экземплярный метод TransformToWords которого выполняет преобразование любого вещественного числа (System.Double) в его «словесный формат».
- Разработать модульные тесты. Примерные тест-кейсы  
[TestCase(double.NaN, ExpectedResult = "Not a number")]  
[TestCase(double.NegativeInfinity, ExpectedResult = "Negative infinity")]  
[TestCase(double.PositiveInfinity, ExpectedResult = "Positive infinity")]  
[TestCase(-0.0d, ExpectedResult = "zero")]  
[TestCase(0.0d, ExpectedResult = "zero")]  
[TestCase(0.1d, ExpectedResult = "zero point one")]  
[TestCase(-23.809d, ExpectedResult = "minus two three point eight zero nine")]  
[TestCase(-0.123456789d, ExpectedResult = "minus zero point one two three four five six seven eight nine")]  
[TestCase(1.23333e308d, ExpectedResult = "one point two three three three three E plus three zero eight")]  
[TestCase(double.Epsilon, ExpectedResult = "four point nine four zero six five six four five eight four one two four seven E minus three two four")]  
и т.д. для double.MaxValue, double.MinValue.

- Проанализировать полученное решение на предмет возможности его использования для получения "словесного формата" на другом (русском, немецком и т.д.) языке.

#### *Задание 2.*

- Разработать неизменяемый класс `Polynomial` (полином) для работы с многочленами  $n$ -ой степени от одной переменной вещественного типа (в качестве внутренней структуры для хранения коэффициентов использовать `sz`-массив).
- Для разработанного класса реализовать протокол эквивалентности по значению, перегрузить операции (включая `"=="` и `"!="`), допустимые для работы с многочленами (исключая деление многочлена на многочлен).
- Разработать модульные тесты для тестирования методов класса.

#### *Задание 3.*

- Разработать систему типов для моделирования инфраструктуры для онлайн-чтения книг.
- Разработать модульные тесты для проверки функционирования разработанной системы типов.

### **Лабораторное занятие 2. Инкапсуляция Наследование классов. Интерфейсы, абстрактные классы**

Студентам рекомендуется выполнить следующие задания:

#### *Задание 1.*

- Разработать систему типов для моделирования работы колл-центра с тремя уровнями сотрудников – оператора, менеджера и директора.
- Разработать модульные тесты для проверки функционирования разработанной системы типов.

#### *Задание 2.*

- Класс `Book` должен иметь следующие приватные поля:
  - `published` (`System.Boolean`) И `datePublished` (`System.DateTime`); поле `published` имеет значение `true` когда `datePublished` связано с датой публикации, `false` – в противном случае
  - поле `totalPages` (`System.Int32`) содержит количество страниц в книге;
- Класс `Book` должен иметь два конструктора.
- Первый конструктор с тремя параметрами `author`, `title`, `publisher`.
- Параметры `author`, `title` И `publisher` используются для неизменяемых свойств `Author`, `Title` and `Publisher` соответственно.

- Второй конструктор включает параметр `isbn`, который связан с автоматическим свойством `ISBN`.
- Второй конструктор использует ключевое слово `this` для вызова первого.
  - Класс `Book` должен содержать следующие свойства.
- `Title (System.String)` свойство только для чтения, чье значение устанавливается в конструкторе.
- `Pages (System.Int32)` свойство, которое указывает количество страниц в книге. Значение хранится в приватном поле `totalPages`. Свойство должно выбрасывать `ArgumentOutOfRangeException` в случае не положительного параметра.
- `Publisher (System.String)` свойство только для чтения, чье значение устанавливается в конструкторе.
- `ISBN (System.String)` свойство только для чтения, которое возвращает для объекта книга International Standard Book Number. Значение хранится в приватном поле автоматического свойства.
- `Author (System.String)` свойство только для чтения, чье значение устанавливается в конструкторе.
- `Price` и `Currency` свойства только для чтения, значения которых устанавливаются в методе `SetPrice`. Свойство `Currency` является трехсимвольным ISO символом валюты (например, `USD` для `U.S.` доллара).
  - Класс `Book` должен
- Иметь метод `Publish`, устанавливающий значение полей `published` и `datePublished`.
- Иметь метод `GetPublicationDate`, возвращающий строку `"NYP"` когда `published = false` и дату публикации `datePublished` в противном случае.
- Иметь метод `SetPrice`, который устанавливает значения свойств `Price` и `Currency`.
- Переопределять метод `ToString` (производный от `System.Object`), возвращающий информацию об объекте в виде автора и названия книги.
- Переопределять методы `Object.Equals` and `GetHashCode` (производные от `System.Object`), принимая за равные два объекта с одинаковым значением `isbn`.

- Разработать модульные тесты.

### **Лабораторное занятие 3. Делегаты и события**

Студентам рекомендуется выполнить следующие задания:

#### *Задание 1.*

- Метеостанция `WeatherStation` работает на базе запатентованного объекта `WeatherData`, отслеживающего текущие погодные условия (температура (`Temperature`), влажность (`Humidity`), атмосферное давление (`Pressure`)).
- Разработать систему классов для создания текущей сводки (`CurrentConditionsReport`) и статистики (`StatisticReport`). Все данные должны обновляться в режиме реального времени, по мере того, как объект `WeatherData` получает данные последних изменений. Предложить два варианта решения (используя интерфейсы и события (`event`)).
- Выполнить моделирование работы метеостанции в консольном приложении.

#### *Задание 2.*

- Разработать класс для имитации часов с обратным отсчетом (через `event`), реализующий возможность по истечении назначенного времени (время ожидания предоставляется классу пользователем) передавать сообщение и дополнительную информацию о событии любому подписавшемуся на событие типу.
- Продемонстрировать работу класса в консольном приложении.

### **Лабораторное занятие 4. Коллекции и обобщенные типы**

Студентам рекомендуется выполнить следующие задания:

#### *Задание 1.*

- Реализуйте обобщенную коллекцию «last-in-first-out» (LIFO) указанного типа `T`, которая реализует интерфейс `IEnumerable<T>`. Использование стандартных коллекций (кроме `sz`-массивов) в качестве внутренней структуры не допускается.
- Разработать модульные тесты.

#### *Задание 2.*

- Разработать обобщенно-типизированный класс-коллекцию `BinarySearchTree` (бинарное дерево поиска).
- Предусмотреть для класса возможности использования подключаемого интерфейса для реализации отношения порядка для его элементов.
- Реализовать три способа обхода дерева (для реализации обходов допускается использование блок-итератора (`yield`)):
  - прямой (`preorder`),

- поперечный (inorder),
  - обратный (postorder).
- Протестировать разработанный класс, используя следующие типы:
  - System.Int32 (использовать сравнение по умолчанию и подключаемый компаратор);
  - System.String (использовать сравнение по умолчанию и подключаемый компаратор);
  - пользовательский класс, для объектов которого реализовано отношения порядка (использовать сравнение по умолчанию и подключаемый компаратор);
  - пользовательскую структуру, для объектов которого не реализовано отношения порядка (использовать подключаемый компаратор).

### **Описание инновационных подходов и методов к преподаванию учебной дисциплины**

При организации образовательного процесса используются следующие инновационные подходы:

***Практико-ориентированный подход***, который предполагает:

- освоение содержание образования через решения практических задач;
- приобретение навыков эффективного выполнения разных видов профессиональной деятельности;
- ориентацию на генерирование идей, развитие предпринимательской культуры;
- использованию процедур, способов оценивания, фиксирующих сформированность профессиональных компетенций.

***Метод анализа конкретных ситуаций (кейс-метод)***, который предполагает:

- приобретение студентом знаний и умений для решения практических задач;
- анализ ситуации, используя профессиональные знания, собственный опыт, дополнительную литературу и иные источники.

## **Методические рекомендации по организации самостоятельной работы обучающихся**

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- поиск (подбор) и обзор литературы и электронных источников по индивидуально заданной проблеме курса;
- подготовка к лабораторным занятиям;
- работы, предусматривающие выполнение заданий, выдаваемых на лабораторных занятиях;
- изучение материала, вынесенного на самостоятельную проработку;
- анализ статистических и фактических материалов по заданной теме.

Для организации самостоятельной работы студентов по учебной дисциплине рекомендуется использовать современные информационные ресурсы, размещенные на образовательном портале <https://edummf.bsu.by> смешанного и дистанционного обучения БГУ и содержащие учебные материалы для электронного сопровождения изучаемой дисциплины.

### **Примерный перечень вопросов к зачету**

1. Объявление переменных и присваивание значений.
2. Использование выражений и операций языка в C#.
3. Использование операторов языка C#.
4. Создание и использование массивов.
5. Определение и вызов методов.
6. Спецификация необязательных и выходных параметров.
7. Взаимодействие типов, объектов, стека потока и управляемой кучи во время выполнения.
8. Создание и использование перечислений.
9. Создание и использование классов.
10. Создание и использование структур.
11. Сравнение ссылочных типов с типами значений.
12. Управление видимостью членов типа.
13. Свойства и индексаторы.
14. Статические методы и данные.
15. Перегрузка операций.
16. Использование наследования для определения новых ссылочных типов.
17. Определение и реализация интерфейсов.
18. Определение абстрактных классов.
19. Обработка исключений.
20. Возникновение исключений.
21. Объявление и использование делегатов.
22. Использование лямбда-выражений.

23. Обработка событий.
24. Использование коллекций.
25. Создание и использование обобщенных типов.
26. Определение обобщенных интерфейсов и понимание вариантности.
27. Использование обобщенных методов и делегатов.
28. Реализация пользовательского класса коллекции.

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

| Название дисциплины, с которой требуется согласование | Название кафедры                             | Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине | Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола) |
|-------------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Технологии Java EE                                    | Веб технологий и компьютерного моделирования | Нет                                                                                                           | Вносить изменения не требуется (протокол № 9 от 20.05.2020г.)                                     |

## ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ

на \_\_\_\_\_ / \_\_\_\_\_ учебный год

| №<br>п/п | Дополнения и изменения | Основание |
|----------|------------------------|-----------|
|          |                        |           |

Учебная программа пересмотрена и одобрена на заседании кафедры Веб-технологий и компьютерного моделирования (протокол № \_\_\_\_ от \_\_\_\_\_ 201\_\_ г.)

Заведующий кафедрой

доктор физ.-мат. наук, доцент \_\_\_\_\_

В.М. Волков

УТВЕРЖДАЮ

Декан факультета

доктор физ.-мат. наук, доцент \_\_\_\_\_

С.М. Босяков