

Учреждение образования  
«Белорусский государственный педагогический университет  
имени Максима Танка»

УТВЕРЖДАЮ



Проректор по учебной работе БГПУ

А.В.Маковчик

2019 г.

Регистрационный № УД-24-2-138-2019 /уч.

**ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ  
И МЕТОДЫ АЛГОРИТМИЗАЦИИ**

Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности  
1-02 05 01 Математика и информатика

Учебная программа составлена на основе Образовательного стандарта высшего образования первая ступень специальность 1-02 05 01 Математика и информатика (ОСВО 1 02 05 01-2013), утвержден и введен в действие постановлением Министерства образования РБ 30.08.2013 г. № 87 и учебного плана специальности

**СОСТАВИТЕЛИ:**

С.И. Зенько, доцент кафедры информатики и методики преподавания информатики учреждения образования «Белорусский государственный педагогический университет имени Максима Танка», кандидат пед. наук, доцент;

А.З. Кутыш, старший преподаватель кафедры информатики и методики преподавания информатики учреждения образования «Белорусский государственный педагогический университет имени Максима Танка»

**РЕЦЕНЗЕНТЫ:**

В.М. Котов, заведующий кафедрой дискретной математики и алгоритмики факультета прикладной математики и информатики Белорусского государственного университета, доктор физ.-мат. наук, профессор;

Т.С. Жилинская, заведующий кафедрой информационных технологий в культуре Учреждения образования «Белорусский государственный университет культуры и искусств», кандидат педагогических наук.

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой информатики и методики преподавания информатики (протокол № 4 от 25.11.2019 г.);

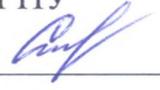
Заведующий кафедрой  С.В.Вабищевич

Научно-методическим советом БГПУ

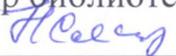
(протокол № 2 от 17.12.2019 г.).

Оформление учебной программы и сопровождающих ее материалов действующим требованиям Министерства образования Республики Беларусь соответствует

Методист учебно-методического  
отдела БГПУ

 С.А.Стародуб

Директор библиотеки

 Н.П.Сятковская

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Технологии программирования и методы алгоритмизации» предназначена для студентов учреждений высшего образования, обучающихся по специальности 1-02 05 01 «Математика и информатика». Учебная программа по дисциплине «Технологии программирования и методы алгоритмизации» составлена в соответствии с требованиями образовательного стандарта Республики Беларусь и типового учебного плана этой специальности.

На протяжении истории развития информационных технологий эволюция языков программирования означала изменение вычислительной среды, способа мышления и самих подходов к программированию: от структурного к объектно-ориентированному и компонентному программированию. Практико-ориентированность учебной дисциплины «Технологии программирования и методы алгоритмизации» очевидна. В рамках учебной дисциплины студенты знакомятся с основными принципами современных технологий программирования, изучают не менее двух языков программирования. Это позволяет им получить качественную подготовку, способствующую, в дальнейшем эффективно реализовывать процесс обучения учащихся в школе основам алгоритмизации и программирования как на базовом, так и на профильном уровнях.

Учебная дисциплина преподается на 1 курсе во 2-м семестре и на 2 курсе в 3-м и 4-м семестрах. Освоение системы знаний и умений по учебной дисциплине «Технологии программирования и методы алгоритмизации» позволит студентам продуктивно осваивать содержание учебных дисциплин как специального цикла («Информационные системы и сети», «Методика преподавания информатики», «Практикум по решению задач по информатике»), так и общепрофессионального цикла («Информационные технологии в образовании»).

*Цель учебной дисциплины* – формирование профессиональных компетенций учителя математики и информатики в области технологий программирования и методов алгоритмизации.

*Задачи изучения учебной дисциплины:*

- формирование знаний теоретических основ алгоритмизации;
- овладение понятийным аппаратом языков и технологий программирования;
- изучение основных принципов структурного, объектно-ориентированного, обобщенного и компонентно-ориентированного программирования;
- приобретение умений разработки алгоритмов, классов, обобщенных наборов данных, компонентов и программ их реализации.

### **Требования к освоению учебной дисциплины**

В результате изучения дисциплины обучаемый должен:

**знать:**

- специальную терминологию современных технологий программирования;
- основные структуры данных и базовые структуры алгоритмов;
- методы алгоритмизации;
- сущность и особенности структурно-модульного и объектно-ориентированного программирования;
- основные приемы (визуальное проектирование, событийное программирование и др.) использования современных технологий программирования при разработке приложений.

**уметь:**

- работать в современных средах программирования;
- использовать основные конструкции языка программирования высокого уровня при проектировании и отладке алгоритмов;
- применять методы алгоритмизации при разработке программ на языке высокого уровня;
- разрабатывать объектные модели в различных предметных областях;
- создавать приложения прикладного характера с помощью современных технологий программирования.

**владеть:**

- способами проектирования алгоритмов и их реализацией с помощью современных средств программирования;
- приемами разработки приложений на основе объектных моделей и их тестирования;
- методами и средствами современных парадигм программирования: объектно-ориентированного, обобщенного, компонентно-ориентированного и др.

### **Требования к освоению учебной дисциплины в соответствии с образовательным стандартом**

Согласно образовательному стандарту высшего образования ОСВО 1-02 05 01-2013 по специальности 1-02 05 01 Математика и информатика изучение учебной дисциплины «Технологии программирования и методы алгоритмизации» должно обеспечить формирование у студентов академических, социально-личностных и профессиональных компетенций.

#### ***Требования к академическим компетенциям***

Специалист должен:

- АК-7. Иметь навыки, связанные с использованием технических устройств, управлением информацией и работой с компьютером.

#### ***Требования к социально-личностным компетенциям специалиста***

Специалист должен:

– СЛК-7. Быть способным осуществлять самообразование и совершенствовать профессиональную деятельность.

### ***Требования к профессиональным компетенциям специалиста***

Специалист должен быть способен:

Обучающая деятельность

– ПК-2. Использовать оптимальные методы, формы и средства обучения.

Воспитательная деятельность

– ПК-8. Формировать базовые компоненты культуры личности обучающегося.

Развивающая деятельность

– ПК-12. Развивать навыки самостоятельной работы обучающихся с учебной, справочной, научной литературой и др. источниками информации.

Ценностно-ориентационная деятельность

– ПК-16. Оценивать учебные достижения обучающихся, а также уровни их воспитанности и развития.

– ПК-17. Осуществлять профессиональное самообразование и самовоспитание с целью совершенствования профессиональной деятельности.

### **Распределение общего количества часов по семестрам**

Трудоёмкость учебной дисциплины составляет 8,5 зачетных единиц.

Дневная форма получения образования 1 курс 2 семестр (3,5 зачетные единицы), 2 курс 3 семестр (2 зачетные единицы) и 2 курс 4 семестр (3 зачетные единицы).

Всего на учебную дисциплину «Технологии программирования и методы алгоритмизации» учебным планом предусмотрено 318 часа, из них аудиторных занятий – 150 часов (58 лекционных и 92 лабораторных) и самостоятельной работы – 168 часа (в том числе 36 часов – подготовка к экзамену).

Текущая аттестация проводится в соответствии с учебным планом специальности в форме зачета (2, 3 семестры) и экзамена (4 семестр).

Распределение количества аудиторных часов по семестрам:

2 семестр – 68 часов аудиторные занятия (28 часов – лекции, 40 часов – лабораторные работы), 62 часа – самостоятельная работа; текущая аттестация – зачет;

3 семестр – 42 часа аудиторные занятия (16 часов – лекции, 26 часов – лабораторные работы), 34 часа – самостоятельная работа; текущая аттестация – зачет;

4 семестр – 40 часов аудиторные занятия (14 часов – лекции, 26 часов – лабораторные работы), 72 часа – самостоятельная работа (в том числе 36 часов – подготовка к экзамену), текущая аттестация – экзамен.

### **Структура содержания учебной дисциплины**

Тематика учебной дисциплины «Технологии программирования и методы алгоритмизации» способствует выявлению и развитию у студентов системных подходов к конструированию объектных моделей из разных предметных

областей, логического и объектно-ориентированного мышления, подготовке студентов к активной профессиональной деятельности в условиях современного общества. Учебная дисциплина содержит три раздела. Первые два раздела являются пропедевтикой к изучению основ современных технологий программирования.

Первый раздел учебной дисциплины предназначен для знакомства студентов с основными понятиями теории и практики алгоритмизации и языков программирования. Раскрываются парадигмы программирования.

Второй раздел учебной дисциплины посвящен структурному программированию, методам алгоритмизации (поиск в массиве, упорядочивание элементов массива, циклические перестановки элементов массива) и их реализации.

Третий раздел учебной дисциплины посвящен изучению принципов и средств реализации современных парадигм программирования (объектно-ориентированного, обобщенного, компонентно-ориентированного программирования и др.), формированию умений визуального проектирования графического интерфейса приложений, приобретению опыта разработки Windows-приложений прикладного характера.

Практическая реализация подготовки осуществлять за счет разработки алгоритмов, проектов и Windows-приложений на языках программирования Pascal и C#.

### **Методы обучения**

На лекционных и лабораторных занятиях методическая система обучения студентов базируется на деятельностно-семантическом подходе и подходе взаимосвязанного изучения технологий программирования. Приветствуются использование методов активного обучения.

На лекциях особое внимание следует уделять использованию цифровых и мультимедийных технологий для демонстрации особенностей и возможностей изучаемых языков и технологий программирования, а также для оперативного получения обратной связи.

Лабораторные занятия направлены на закрепление лекционного теоретического материала и на формирование универсальных умений и навыков практического использования полученных знаний при выполнении конкретных учебных и практико-ориентированных заданий. Организация лабораторных работ предполагает использование лично-ориентированных методов обучения, что способствует развитию индивидуально-творческих способностей каждого студента и приобретению навыков самостоятельной работы.

Для управления самостоятельной работой рекомендуется использовать интерактивные учебные пособия, тренажеры, тестирующие программы и др. Текущий контроль осуществляется при выполнении и сдаче лабораторных работ. Для промежуточной аттестации студентов предлагается тематический контроль (тестирование, коллоквиум).

# СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

## РАЗДЕЛ I. ОСНОВЫ ТЕОРИИ И ПРАКТИКИ АЛГОРИТМИЗАЦИИ И ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

### Тема 1.1. Основы алгоритмизации

Этапы решения задач на компьютере.

Понятие алгоритма. Свойства алгоритма. Способы представления алгоритмов.

Блок-схемы алгоритмов, основные элементы. Базовые структуры.

### Тема 1.2. Основы теории языков программирования

Понятие языка программирования. Область применения языков программирования. Критерии эффективности языков программирования.

Пути эволюции языков и технологий программирования. Парадигмы программирования. Виды языков программирования. Парадигма структурного программирования.

### Тема 1.3. Системы и среды программирования

Понятия систем и сред программирования. Назначение, основные функции и состав системы программирования.

## РАЗДЕЛ II. СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

### Тема 2.1. Основные элементы языка программирования

Основные понятия языка программирования: символ, слова, величина.

Скалярные (простые) типы данных: перечисляемый, целый, вещественный, логический, символьный и диапазонный.

Операции и выражения. Унарные и бинарные операции. Приоритетность операций. Арифметические и логические выражения. Совместимость типов операндов выражения.

Основные арифметические функции.

Структура программы.

### Тема 2.2. Операторы языка программирования. Базовые алгоритмические конструкции

Оператор присваивания. Комментарии в программе. Процедуры ввода и вывода. Форматы вывода числовых данных.

Понятие структурного оператора.

Алгоритмическая конструкция «Следование». Составной оператор.

Алгоритмическая конструкция «Ветвление». Оператор альтернативы (условный). Полная и неполная формы оператора альтернативы. Оператор выбора.

Алгоритмическая конструкция «Повторение». Понятия «цикл», «тело цикла», «условие цикла». Циклы с определяемым количеством повторений. Оператор цикла *while*. Оператор цикла *repeat*. Цикл с известным количеством повторений. Оператор цикла *for*. Вложенные циклы.

### **Тема 2.3. Структурированные (составные, сложные) типы данных. Методы алгоритмизации**

Понятие массива. Одномерные и двумерные массивы: описание массивов, способы ввода и вывода элементов массива.

Методы алгоритмизации: поиск в массиве, упорядочивание (сортировка) элементов массива, циклические перестановки элементов массива.

Методы поиска элемента массива: поиск перебором, бинарный поиск и др. Поиск элементов в одномерных и двумерных массивах с заданными свойствами (свойства элементов, лежащих на (под, над) главной и побочной диагоналями квадратной матрицы и др.).

Методы упорядочивания элементов массива: сортировка выбором, сортировка обменом (пузырьковая) и др.

Циклические перестановки элементов массива.

Анализ алгоритмов. Сложность алгоритмов.

Понятие строковой величины. Объявление строковых величин. Операции, процедуры и функции работы со строковыми величинами.

Понятие множества. Описание множеств. Операции над множествами: объединение, пересечение, разность, операции сравнения, операция *in*.

Понятия записи, поля записи. Описание записей. Массивы записей. Оператор *with*.

### **Тема 2.4. Подпрограммы**

Понятие подпрограммы. Виды подпрограмм. Локальные и глобальные переменные.

Процедуры пользователя. Структура процедуры пользователя. Организация вызова процедуры пользователя. Типы параметров (фактические и формальные, параметры-значения и параметры-переменные). Виды процедур пользователя: процедуры без параметров, процедуры с параметрами-значениями, процедуры с параметрами-значениями и параметрами-переменными.

Функции пользователя. Структура функции пользователя. Организация вызова функции пользователя. Отличия функции пользователя от процедуры пользователя.

Понятие итерации. Понятие рекурсии. Рекурсивные подпрограммы.

### **Тема 2.5. Основы работы с файлами**

Понятие файлов. Виды файлов. Типизированные файлы. Работа с типизированными файлами: создание файла, использование данных из файла, дополнение файла новыми данными.

### **Тема 2.6. Основы программирования графики и звука**

Основы работы с графикой. Модуль *Graph*. Графические примитивы. Работа с цветом. Работа с пером и кистью. Действия со шрифтами. Имитация движения графических объектов. Построение графиков элементарных функций. Деловая графика (столбчатые и круговые диаграммы).

Основы работы со звуком. Процедуры и функции для работы со звуком.

## РАЗДЕЛ III. ОСНОВЫ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

### Тема 3.1. Парадигма объектно-ориентированного программирования. Платформа .NET

Объектно-ориентированные языки и среды программирования. Философия и технология .NET. Компоненты платформы .NET. Инфраструктура .NET Framework, строительные блоки .NET – CLR, CTS, CLS. Базовые классы .NET.

Модель выполнения приложений в среде .NET Framework. Промежуточный язык MSIL (*Microsoft Intermediate Language*).

### Тема 3.2. Основы объектно-ориентированного и компонентно-ориентированного языка программирования C#

Синтаксис языка.

Встроенные типы и системные типы. Пространство имен *System*. Директива *using System*. Встроенная утилита *Object Browser*, просмотр содержимого пространства имен *System*.

Класс *System.Console*. Операции ввода и вывода. Чтение строки символов. Преобразование строки символов в число, метод *Parse()*. Формат вывода. Класс *System.Math*, его методы и константы.

Операции, приоритеты операций. Операторы и выражения. Преобразование типов в выражениях.

Массивы. Класс *System.Random*, применение его методов для создания массива случайных чисел. Класс *System.Array*, его методы и свойство *Length*. Цикл *foreach*, перебор элементов массивов.

Структуры. Создание структур. Работа со структурами.

### Тема 3.3. Объектно-ориентированное проектирование и программирование

Понятие объектно-ориентированного программирования. Объектные модели, принципы их построения, особенности реализации.

Классы в C#. Форма определения класса. Члены класса. Спецификаторы доступа к членам класса.

Конструкторы. Параметризованные конструкторы. Назначение оператора *new*. Создание экземпляра класса.

Типы значений. Ссылочные типы. Динамическое использование памяти: стеки и кучи. Сбор «мусора» и использование деструкторов.

Определение пользовательских методов класса. Спецификаторы доступа к методам класса. Модификаторы *ref*, *out*, *params* параметров методов.

Возвращение методами объектов класса. Возвращение методами массивов.

Общие (статические) члены класса. Доступ к общим членам класса. Ограничения на *static*-методы.

Область видимости и время существования переменных. Преобразование и приведение типов.

Обработка исключительных ситуаций.

### Тема 3.4. Принципы объектно-ориентированного программирования

Перегрузка методов и конструкторов как механизм реализации полиморфизма.

Перегрузка операторов. Перегрузка бинарных, унарных и операторов отношений.

Инкапсуляция и свойства. Формат записи свойств и правила их использования.

Индексаторы. Одномерные индексаторы.

Наследование, базовые и производные классы. Доступы к членам базовых классов. Конструкторы и наследование. Виртуальные методы. Абстрактные классы. Переопределение виртуальных и абстрактных методов переопределение как механизм реализации полиморфизма.

Интерфейсы и множественное наследование. Определение. Реализация интерфейсов.

### Тема 3.5. Принципы обобщенного программирования и их реализация

Понятие обобщенного программирования. Принципы обобщенного программирования (абстрагирование, иерархия, типизация).

Коллекции и наборы. Обзор коллекций. Интерфейсы коллекций. Классы коллекций общего назначения.

Классы *ArrayList*, *Queue*, *Stack*, другие типы наборов данных из пространства имен *System.Collections*. Методы работы с различными структурами данных.

Динамическая идентификация типов. Проверка типа с помощью ключевого слова *is*. Использование операторов *as*, *typeof*.

Обобщения. Разработка обобщенных наборов, методов и классов.

### Тема 3.6. Основы компонентно-ориентированного программирования

Принципы работы *Windows*-приложений.

Делегаты. Назначение делегатов. События и сообщения, их обработка. Объявление и генерация событий. Реализация обработчиков событий.

Разработка объектной модели приложения.

Графический интерфейс приложений. Технология *Windows Forms*. Пространство имен *System.WinForms*, класс *Form*. Добавление форм к проекту. Методы и события формы.

Элементы управления. Создание обработчиков событий элементов управления.

Понятие компонента и компонентной модели. Разработка программных компонентов. Приложение-сервер, приложение-контейнер (клиент). Интерфейс *System.ComponentModel.IComponent* и класс *System.ComponentModel.Component*.

Использование собственных программных компонентов в *Windows*-приложениях прикладного характера.

## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов		Самостоятельная работа студента	Материальное обеспечение (наглядные, методические пособия и др.)	Литература	Форма контроля знаний
		Лекции	Лабораторные занятия				
1	2	3	4	5	6	7	8
<b>2 семестр</b>							
<b>1.</b>	<b>ОСНОВЫ ТЕОРИИ И ПРАКТИКИ АЛГОРИТМИЗАЦИИ И ЯЗЫКОВ ПРОГРАММИРОВАНИЯ</b>	<b>4</b>		<b>8</b>			
<b>1.1.</b>	<b>Основы алгоритмизации</b>	<b>2</b>		<b>4</b>			
1.1.1	<i>Основы алгоритмизации</i> 1. Этапы решения задач на компьютере. 2. Понятие алгоритма. Свойства алгоритма. Способы представления алгоритмов. 3. Блок-схемы алгоритмов, основные элементы. Базовые структуры.	2		4	Лекционный материал в электронном виде	1, 3; 1, 8	Устный опрос.
<b>1.2.</b>	<b>Основы теории языков программирования</b>	<b>1</b>		<b>2</b>			
1.2.1	<i>Основы теории языков программирования</i> 1. Понятие языка программирования. Область применения языков программирования. 2. Критерии эффективности языков программирования. 3. Пути эволюции языков и технологий программирования. Парадигмы программирования. Виды языков программирования. 4. Парадигма структурного программирования.	1		2	Лекционный материал в электронном виде	1, 3; 1, 6, 8	Устный опрос.

1	2	3	4	5	6	7	8
<b>1.3.</b>	<b>Системы и среды программирования</b>	<b>1</b>		<b>2</b>			
1.3.1	<i>Системы и среды программирования</i> 1. Понятия систем и сред программирования. 2. Назначение, основные функции и состав системы программирования.	1		2	Лекционный материал в электронном виде	1–3; 1, 6, 8	Устный опрос.
<b>2.</b>	<b>СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ</b>	<b>24</b>	<b>40</b>	<b>54</b>			
<b>2.1.</b>	<b>Основные элементы языка программирования</b>	<b>2</b>	<b>2</b>	<b>4</b>			
2.1.1	<i>Основные элементы языка программирования</i> 1. Основные понятия языка программирования: символ, слова, величина. 2. Скалярные (простые) типы данных: перечисляемый, целый, вещественный, логический, символьный и диапазонный. 3. Операции и выражения. Унарные и бинарные операции. Приоритетность операций. Арифметические и логические выражения. Совместимость типов операндов выражения. 4. Основные арифметические функции. 5. Структура программы.	2		4	Лекционный материал в электронном виде	2–4; 1, 2, 8	Устный опрос.
2.1.2	<i>Типы переменных, выражения</i> Указание типов переменных в программе. Определение типов констант. Арифметические выражения и их типы. Логические выражения. Запись математических выражений в программе.		2		PascalABC. NET C#	2–4; 2, 7	Проверка лабораторной работы
<b>2.2.</b>	<b>Операторы языка программирования. Базовые алгоритмические конструкции</b>	<b>6</b>	<b>10</b>	<b>12</b>			
2.2.1	<i>Операторы языка программирования</i> 1. Оператор присваивания. Комментарии в программе. 2. Процедуры ввода и вывода. 3. Форматы вывода числовых данных. 4. Понятие структурного оператора. 5. Алгоритмическая конструкция «Следование». Составной оператор.	2		4	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос.
2.2.2	<i>Составление программ с использованием алгоритмической конструкции «Следование»</i> Составление блок-схем решения задач с использованием алгоритмической конструкции «Следование».		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы

1	2	3	4	5	6	7	8
	Объявление констант и переменных в программе. Процедуры ввода и вывода в программе и их геометрическое представление. Оператор присваивания в программе и его геометрическое представление.						
2.2.3	<b>Алгоритмическая конструкция «Ветвление»</b> 1. Алгоритмическая конструкция «Ветвление». 2. Оператор альтернативы (условный). 3. Полная и неполная формы оператора альтернативы. 4. Оператор выбора.	2		4	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос.
2.2.4	<b>Составление программ с использованием алгоритмической конструкции «Ветвление»</b> Полная и неполная формы оператора альтернативы в программах и их геометрическое представление. Полная и неполная формы оператора выбора в программах и их геометрическое представление.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.2.5	<b>Алгоритмическая конструкция «Повторение»</b> 1. Алгоритмическая конструкция «Повторение». 2. Понятия «цикл», «тело цикла», «условие цикла». 3. Циклы с определяемым количеством повторений. Оператор цикла while. Оператор цикла repeat. 4. Цикл с известным количеством повторений. Оператор цикла for. 5. Вложенные циклы.	2		4	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос.
2.2.6	<b>Составление программ с использованием алгоритмической конструкции «Повторение»</b> Оператор цикла с предусловием в программах и его геометрическое представление в блок-схемах. Оператор цикла с постусловием в программах и его геометрическое представление в блок-схемах.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
	Оператор цикла с параметром в программах и две формы его представления в блок-схемах. Составление программ с использованием операторов цикла и их представление в виде блок-схем.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.2.7	<b>Составление программа с использованием комбинаций алгоритмических конструкций</b> Понятие внешнего и внутреннего циклов и их представление в виде блок-схем. Количество повторений для внешнего и внутреннего циклов.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы

1	2	3	4	5	6	7	8
	Сочетание разных операторов циклов в программах с использованием вложенных циклов. Использование совместно различных алгоритмических конструкций						
<b>2.3.</b>	<b>Структурированные (составные, сложные) типы данных. Методы алгоритмизации</b>	<b>8</b>	<b>14</b>	<b>22</b>			
2.3.1	<b>Массивы</b> 1. Понятие массива. 2. Одномерные и двумерные массивы: описание массивов, способы ввода и вывода элементов массива. 3. Главная и побочная диагонали двумерного массива.	1		3	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.3.2	<b>Составление программ по формированию массивов</b> Ввод и вывод элементов одномерного массива. Составление программ с использованием одномерных массивов. Ввод и вывод элементов двумерного массива. Составление программ с использованием свойств главной и побочной диагоналей двумерных массивов.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.3.3	<b>Методы алгоритмизации</b> 1. Поиск в массиве. 2. Упорядочивание (сортировка) элементов массива. 3. Циклические перестановки элементов массива: сущность.	1		3	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.3.4	<b>Методы поиска элемента массива</b> 1. Поиск перебором. 2. Бинарный поиск. 3. Поиск элементов в одномерных и двумерных массивах с заданными свойствами: свойства элементов, лежащих на (под, над) главной и побочной диагоналями квадратной матрицы.	1		3	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.3.5	<b>Составление программ с использованием алгоритмов поиска элементов в массивах</b> Методы поиска наименьшего и наибольшего элемента в одномерном массиве. Поиск элементов с заданными свойствами в одномерном массиве. Поиск наименьшего и наибольшего элементов по строкам и столбцам в двумерном массиве. Поиск элементов с заданными свойствами на главной и побочной диагоналях двумерного массива.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы

1	2	3	4	5	6	7	8
2.3.6	<b>Методы упорядочивания и перестановки элементов массива</b> 1. Сортировка выбором. 2. Сортировка обменом. 3. Циклические перестановки элементов массива. 4. Анализ сложности алгоритмов.	1		3	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.3.7	<b>Составление программ с использованием алгоритмов сортировки и перестановки элементов массива</b> Составление программ с использованием методов сортировки элементов одномерного массива. Составление программ с использованием сортировки элементов двумерного массива.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.3.8	<b>Составление программ с использованием алгоритмов перестановки элементов массива</b> Составление программ с использованием методов сдвига элементов массива одномерного массива. Составление программ с использованием алгоритмов перестановки строк (столбцов) двумерного массива.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.3.9	<b>Строковые величины</b> 1. Понятие строковой величины. 2. Объявление строковых величин. 3. Операции над строковыми величинами. 4. Процедуры и функции работы со строковыми величинами.	2		4	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.3.10	<b>Составление программ с использованием строковых величин</b> Составление программ с использованием процедур и функций работы с отдельными словами. Составление программ для обработки текстовых предложений. Составление программ на основе перевода строковых переменных в числовые и наоборот.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.3.11	<b>Множества</b> 1. Понятие множества. 2. Описание множеств. 3. Операции над множествами: объединение, пересечение, разность, операции сравнения, операция in.	1		3	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос

1	2	3	4	5	6	7	8
2.3.12	<b>Составление программ с использованием операций над множествами</b> Составление программ с использованием операций объединения и пересечения множеств. Составление программ с использованием операции in.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.3.13	<b>Записи</b> 1. Понятия записи, поля записи. Описание записей. 2. Массивы записей. 3. Оператор with.	1		3	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.3.14	<b>Составление программ с использованием записей</b> Составление программ с использованием отдельных записей. Составление программ с использованием массивов записей.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
<b>2.4.</b>	<b>Подпрограммы</b>	<b>4</b>	<b>6</b>	<b>8</b>			
2.4.1	<b>Подпрограммы</b> 1. Понятие подпрограммы. 2. Виды подпрограмм. 3. Локальные и глобальные переменные.	1		2	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос.
2.4.2	<b>Процедуры и функции пользователя</b> 1. Процедуры пользователя. Структура процедуры пользователя. Организация вызова процедуры пользователя. 2. Типы параметров (фактические и формальные, параметры-значения и параметры-переменные). 3. Виды процедур пользователя: процедуры без параметров, процедуры с параметрами-значениями, процедуры с параметрами-значениями и параметрами-переменными. 4. Функции пользователя. Структура функции пользователя. Организация вызова функции пользователя. 5. Отличия функции пользователя от процедуры пользователя.	2		4	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос.
2.4.3	<b>Составление программ с использованием процедур пользователя</b> Составление программ с использованием процедуры пользователя без параметров. Составление программ с использованием процедуры пользователя без параметров-переменных. Составление программ с использованием процедуры пользователя общего вида.		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы

1	2	3	4	5	6	7	8
2.4.4	<b>Составление программ с использованием функций пользователя</b> Составление программ с использованием функций пользователя без глобальных параметров. Составление программ с использованием функций пользователя с глобальными параметрами. Составление программ с использованием функций пользователя общего вида		2		PascalABC. NET C#	3, 4; 2, 7	Проверка лабораторной работы
2.4.5	<b>Понятия итерации и рекурсии</b> 1. Понятие итерации. 2. Понятие рекурсии. 3. Рекурсивные подпрограммы.	1		2	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос.
2.4.6	<b>Рекурсивные подпрограммы</b> Составление программ с использованием рекурсивных функций пользователя. Составление программ с использованием рекурсивных процедур пользователя.		2		PascalABC. NET C#	3, 4; 7	Проверка лабораторной работы
<b>2.5.</b>	<b>Основы работы с файлами</b>	<b>2</b>	<b>4</b>	<b>4</b>			
2.5.1	<b>Файлы</b> 1. Понятие файлов. 2. Виды файлов. 3. Типизированные файлы. 4. Работа с типизированными файлами: создание файла, использование данных из файла, дополнение файла новыми данными.	2		4	Лекционный материал в электронном виде	3, 4; 1, 2, 8	Устный опрос
2.5.2	<b>Составление программ с использованием файлов</b> Составление программ записи данных в типизированный файл.		2		PascalABC. NET, C#	3, 4; 7	Проверка лаб. работы
	Составление программ с использованием данных из типизированных файлов.		2		PascalABC. NET, C#	3, 4; 7	Проверка лаб. работы
<b>2.6.</b>	<b>Основы программирования графики и звука</b>	<b>2</b>	<b>4</b>	<b>4</b>			
2.6.1	<b>Основы программирования графики и звука</b> 1. Основы работы с графикой. Модуль Graph. Графические примитивы. Работа с цветом. Работа с пером и кистью. Действия со шрифтами. 2. Имитация движения графических объектов. 3. Построение графиков элементарных функций. 4. Деловая графика (столбчатые и круговые диаграммы). 5. Основы работы со звуком. Процедуры и функции для работы со звуком.	2		4	Лекционный материал в электронном виде	1, 3, 4; 1, 2, 8	Устный опрос

1	2	3	4	5	6	7	8
2.6.2	<b>Составление программ с использованием графики</b> Составление программ составления мозаики. Составление программ с движением графических объектов. Составление программ рисования графиков элементарных функций.		2		PascalABC. NET C#	1, 3, 4; 7	Проверка лабораторной работы
2.6.3	<b>Составление программ с использованием диаграмм</b> Составление программ рисования столбчатых диаграмм. Составление программ рисования круговых диаграмм.		1		PascalABC. NET C#	1, 3, 4; 7	Проверка лабораторной работы
2.6.4	<b>Составление программ с использованием звуковых эффектов</b> Составление программ с использованием кодов нот и включения звук. Составление программ с подключением готовых звуковых файлов.		1		PascalABC. NET C#	1, 3, 4; 7	Проверка лабораторной работы
<b>Итого за 2-й семестр</b>		<b>28</b>	<b>40</b>	<b>62</b>			<b>Зачет</b>
<b>3 семестр</b>							
<b>3.</b>	<b>ОСНОВЫ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ</b>	<b>30</b>	<b>52</b>	<b>70</b>			
<b>3.1.</b>	<b>Парадигма объектно-ориентированного программирования. Платформа .NET</b>	<b>4</b>		<b>8</b>			
3.1.1	<b>Объектно-ориентированные языки и среды программирования</b> 1. Объектно-ориентированные языки. 2. Объектно-ориентированные среды программирования	1		2	Лекционный материал в электронном виде	1–3; 3–6	Устный опрос
3.1.2	<b>Философия и технология .NET.</b> 1. Компоненты платформы .NET. 2. Инфраструктура .NET Framework, строительные блоки .NET – CLR, CTS, CLS. 3. Базовые классы .NET.	2		4	Лекционный материал в электронном виде	1–3; 3–6	Устный опрос
3.1.3	<b>Модель выполнения приложений в среде .NET Framework.</b> 1. Промежуточный язык MSIL (Microsoft Intermediate Language). 2. Генерация управляемого кода. Сборка.	1		2	Лекционный материал в электронном виде	1–3; 3–6	Устный опрос

1	2	3	4	5	6	7	8
<b>3.2.</b>	<b>Основы объектно-ориентированного и компонентно-ориентированного языка программирования C#</b>		<b>8</b>				
3.2.1	<b>Синтаксис объектно-ориентированного языка C#.</b> Синтаксис языка. Встроенные типы и системные типы. Пространство имен System. Директива using System. Встроенная утилита Object Browser, просмотр содержимого пространства имен System.		1		C#	2, 3; 6	Проверка лабораторной работы
3.2.2	<b>Класс System.Console</b> Операции ввода и вывода. Чтение строки символов. Преобразование строки символов в число, метод Parse(). Формат вывода. Класс System.Math, его методы и константы.		1		C#	2, 3; 6	Проверка лабораторной работы
3.2.3	<b>Операции и операторы языка программирования</b> Операции, приоритеты операций. Операторы и выражения. Преобразование типов в выражениях.		2		C#	2, 3; 6	Проверка лабораторной работы
3.2.4	<b>Одномерные массивы</b> Одномерные массивы. Способы создания элементов массива. Класс System.Random, применение его методов для создания массива случайных чисел. Класс System.Array, его методы и свойство Length.		2		C#	2, 3; 6	Проверка лабораторной работы
3.2.5	<b>Структуры</b> Структуры. Создание структуры. Работа со структурами. Массивы структур.		2		C#	2, 3; 6	Проверка лабораторной работы
<b>3.3.</b>	<b>Объектно-ориентированное проектирование и программирование</b>	<b>10</b>	<b>8</b>	<b>22</b>			
3.3.1	<b>Понятие объектно-ориентированного программирования. Объектные модели, принципы их построения, особенности реализации</b> 1. Понятия объекта, объектной модели. 2. Понятие объектно-ориентированного программирования. 3. Характеристики объектно-ориентированного программирования.	1		2	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.3.2	<b>Классы в C#</b> 1. Форма определения класса. 2. Члены класса. 3. Спецификаторы доступа к членам класса.	1		2	Лекционный материал в электронном виде	2, 3; 6	Устный опрос

1	2	3	4	5	6	7	8
3.3.3	<b>Конструкторы</b> 1. Параметризованные конструкторы. 2. Назначение оператора new. 3. Создание экземпляра класса.	1		3	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.3.4	<b>Класс «Натуральные числа»</b> Разработка класса «Натуральные числа». Определение членов-переменных класса. Разработка конструкторов и методов класса и их пошаговая отладка.		2		C#	2, 3; 6	Проверка лабораторной работы
3.3.5	<b>Типы значений. Ссылочные типы</b> 1. Динамическое использование памяти: стеки и кучи. 2. Сбор «мусора» и использование деструкторов.	1		3	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.3.6	<b>Определение пользовательских методов класса</b> 1. Спецификаторы доступа к методам класса. 2. Модификаторы ref, out, params параметров методов. 3. Возвращение методами объектов класса. 4. Возвращение методами массивов.	2		4	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.3.7	<b>Использование модификаторов параметров методов</b> Использование модификаторов ref, out, params параметров методов в классе «Натуральные числа».		2		C#	2, 3; 6	Проверка лабораторной работы
3.3.8	<b>Общие (статические) члены класса</b> 1. Доступ к общим членам класса. 2. Ограничения на static-методы.	2		4	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.3.9	<b>Класс «Числовой массив»</b> Разработка класса «Числовой массив». Определение статических и нестатических членов-переменных класса. Разработка статических и нестатических методов класса.		2		C#	2, 3; 6	Проверка лабораторной работы
3.3.10	<b>Решение задач с использованием класса «Числовой массив»</b> Решение задач с использованием класса «Числовой массив». Разработка и тестирование методов класса, необходимых для решения задач.		2		C#	2, 3; 6	Рейтинговая контрольная работа № 1

1	2	3	4	5	6	7	8
3.3.11	<b>Цикл «жизни» переменных</b> 1. Область видимости и время существования переменных. 2. Преобразование и приведение типов.	1		2	Лекционный материал в эл. виде	2, 3; 6	Устный опрос
3.3.12	<b>Исключительные ситуации</b> Обработка исключительных ситуаций.	1		2	Лекционный материал в эл. виде	2, 3; 6	Устный опрос
<b>3.4.</b>	<b>Принципы объектно-ориентированного программирования</b>	<b>12</b>	<b>24</b>	<b>28</b>			
3.4.1	<b>Перегрузка методов и конструкторов как механизм реализации полиморфизма</b> 1. Перегрузка методов. 2. Перегрузка конструкторов. 3. Понятие стека. 4. Класс «Стек из символов».	2		4	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.4.2	<b>Класс «Стек»</b> Класс «Стек» из данных встроенного типа, соответствующего заданию. Перегрузка конструкторов и методов класса.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.3	<b>Класс «Point»</b> Разработка класса «Point». Перегрузка конструкторов, конструктор копий. Перегрузка арифметических операций (бинарных, унарных) и операций отношений для класса Point.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.4	<b>Инкапсуляция и свойства</b> Инкапсуляция и свойства. Формат записи свойств и правила их использования. Разработка свойств для полей класса Point и решение задач с их использованием.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.5	<b>Одномерные индексы</b> Одномерные индексы. Разработка одномерного индекса для класса «Числовой массив», для класса «Стек».		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.6	<b>Класс Figure</b> Наследование. Базовый класс Figure и его производные классы: Rectangle, Triangle, Kwadrat, Circle.		2		C#	2, 3; 6	Проверка лабораторной работы
<b>Итого за 3-й семестр</b>		<b>16</b>	<b>26</b>	<b>34</b>			<b>Зачет</b>

4 семестр							
1	2	3	4	5	6	7	8
3.4.7	<b>Перегрузка операторов (операций)</b> 1. Перегрузка бинарных операций. 2. Перегрузка унарных операций и операций отношений.	2		4	Лекционный материал в эл. виде	2, 3; 6	Устный опрос
3.4.8	<b>Индексаторы. Одномерные индексаторы</b> 1. Определение одномерного индексатора. 2. Класс «Безопасный массив». 3. Виртуальный массив.	2		6	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.4.9	<b>Наследование, базовые и производные классы</b> 1. Доступы к членам базовых классов. 2. Конструкторы и наследование. 3. Виртуальные методы и их переопределение как механизм реализации полиморфизма. 4. Абстрактные классы, переопределение абстрактных методов.	4		8	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.4.10	<b>Проект «Планиметрия»</b> Разработка проекта «Планиметрия». Базовый класс Figure и производный класс Point.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.11	<b>Проект «Планиметрия»</b> Разработка проекта «Планиметрия». Производный класс Otrezok. Решение задач планиметрии с использованием классов Point, Otrezok.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.12	<b>Проект «Планиметрия»</b> Разработка проекта «Планиметрия». Производный класс Prjamaja. Решение задач планиметрии с использованием класса Prjamaj.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.13	<b>Проект «Планиметрия»</b> Разработка проекта «Планиметрия». Производный класс Circle. Решение задач планиметрии с использованием класса Circle.		2		C#	2, 3; 6	Проверка лабораторной работы
3.4.14	<b>Проект «Планиметрия»</b> Разработка проекта «Планиметрия». Производный класс Triangle. Решение задач планиметрии с использованием класса Triangle.		4		C#	2, 3; 6	Рейтинговая контрольная работа № 2
3.4.15	<b>Интерфейсы и множественное наследование</b> 1. Определение интерфейса. 2. Реализация интерфейсов. 3. Некоторые стандартные интерфейсы среды .NET Framework.	2		6	Лекционный материал в электронном виде	2, 3; 6	Устный опрос

1	2	3	4	5	6	7	8
3.4.16	<b>Проект «Наследование»</b> Создание интерфейса IТор для классов проекта «Наследование» и его реализация в соответствующих классах. Реализация интерфейсов ICloneable, IComparable в классе Point и их использование.		2		С#	2, 3; 6	Проверка лабораторной работы
<b>3.5.</b>	<b>Принципы обобщенного программирования и их реализация</b>	<b>4</b>	<b>4</b>	<b>12</b>			
3.5.1	<b>Понятие обобщенного программирования</b> 1. Принципы обобщенного программирования (абстрагирование, иерархия, типизация).	1		3	Лекционный материал в эл. виде	2, 3; 6	Устный опрос
3.5.2	<b>Коллекции и наборы</b> 1. Обзор коллекций. 2. Интерфейсы коллекций. 3. Классы коллекций общего назначения.	1		3	Лекционный материал в электронном виде	2, 3; 6	Устный опрос
3.5.3	<b>Методы работы с различными структурами данных</b> Классы ArrayList, Queue, Stack, другие типы наборов данных из пространства имен System.Collections. Методы работы с различными структурами данных.		1		С#	2, 3; 6	Проверка лабораторной работы
3.5.4	<b>Динамическая идентификация типов</b> 1. Проверка типа с помощью ключевого слова is. 2. Использование операторов as, typeof.	1		3	Лекционный материал в эл. виде	2, 3; 6	Устный опрос
3.5.5	<b>Проект «Наследование»</b> Динамическая идентификация типов для экземпляров классов проекта «Наследование»		1		С#	2, 3; 6	Проверка лабораторной работы
3.5.6	<b>Обобщения</b> 1. Обобщения. Обобщенные наборы, методы и классы.	1		3	Лекционный материал в эл. виде	2, 3; 6	
3.5.7	<b>Разработка обобщений</b> Разработка обобщенных наборов, методов и классов.		2		С#	2, 3; 6	Проверка лаб. Работы
<b>3.6.</b>	<b>Основы компонентно-ориентированного программирования</b>		<b>8</b>				
3.6.1	<b>Принципы работы Windows-приложений</b> Делегаты. Назначение делегатов. События и сообщения, их обработка. Объявление и генерация событий. Реализация обработчиков событий.		1		С#	2, 3; 6	Проверка лабораторной работы

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
3.6.2	<b><i>Разработка объектной модели приложения</i></b> Графический интерфейс приложений. Технология Windows Forms. Пространство имен System.WinForms, класс Form. Добавление форм к проекту. Методы и события формы.		1		С#	2, 3; 6	Проверка лабораторной работы
3.6.3	<b><i>Элементы управления</i></b> Элементы управления. Создание обработчиков событий элементов управления.		2		С#	2, 3; 6	Рейтинговая контрольная работа № 3
3.6.4	<b><i>Компонент и компонентная модель</i></b> Понятие компонента и компонентной модели. Разработка программных компонентов. Приложение-сервер, приложение-контейнер (клиент). Интерфейс System.ComponentModel.IComponent и класс System.ComponentModel.Component.		2		С#	2, 3; 6	Проверка лабораторной работы
3.6.5	<b><i>Windows-приложения прикладного характера</i></b> Использование собственных программных компонентов в Windows-приложениях прикладного характера.		2		С#	2, 3; 6	Проверка лабораторной работы
<b>Итого за 4-й семестр</b>		<b>14</b>	<b>26</b>	<b>36</b>			<b>Экзамен</b>
<b>ИТОГО</b>		<b>58</b>	<b>92</b>	<b>132</b>			

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### ЛИТЕРАТУРА

#### Основная литература

1. Городко, С. И. Современные технологии программирования : учеб.-метод. пособие для студентов / С. И. Городко, С. В. Снисаренко ; М-во образования Респ. Беларусь, Белорус. гос. ун-т информатики и радиоэлектроники, Фак. компьютер. проектирования. – Минск : БГУИР, 2017. – 67 с.
2. Демидович, В. Е. С#. Консольные приложения : справочник / В. Е. Демидович, Е. М. Демидович. – Минск : Бестпринт, 2017. – 658 с.
3. Зенько, С. И. Реализация современных образовательных технологий при обучении программированию будущих учителей информатики : пособие для студентов учреждений высш. образования / С. И. Зенько, А. З. Кутыш ; М-во образования Респ. Беларусь, Белорус. гос. пед. ун-т. – Минск : БГПУ, 2019. – 319 с.
4. Кутыш, А. З. Взаимосвязное обучение технологиям программирования : практикум для студентов учреждений высш. образования, обучающихся по специальности 1-02 05 01 Математика и информатика / А. З. Кутыш ; М-во образования Респ. Беларусь, Белорус. гос. пед. ун-т. – Минск : БГПУ, 2017. – 200 с.

#### Дополнительная литература

1. Буславский, А. А. Начальный уровень обучения программированию на языке Pascal / А. А. Буславский ; Мин. обл. ин-т развития образования. – Минск : МОИРО, 2010. – 89 с.
2. Вабищевич, С. В. Рабочая тетрадь по программированию на языке Паскаль / С. В. Вабищевич ; М-во образования Респ. Беларусь, Белорус. гос. пед. ун-т. – Минск : БГПУ, 2018. – 42 с.
3. Заборовский, Г. А. Изучение объектно-ориентированного программирования в среде Pascal ABC / Г. А. Заборовский // Вес. БДПУ. Сер. 3, Фізика. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – 2015. – № 2. – С. 47–50.
4. Заборовский, Г. А. Структурно-модульное и событийно-ориентированное программирование в среде Pascal ABC / Г. А. Заборовский // Вес. БДПУ. Сер. 3, Фізика. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – 2014. – № 2. – С. 63–65.
5. Зенько, С. И. Формирование информационной компетентности будущего учителя информатики при обучении технологии объектно-ориентированного программирования / С. И. Зенько, А. З. Кутыш // Вес. БДПУ. Сер. 3, Фізика. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – 2015. – № 2. – С. 51–57.
6. Павловская, Т. А. С#. Программирование на языке высокого уровня / Т. А. Павловская. – СПб. : Питер : Мир книг, 2013. – 432 с.
7. Расолько, Г. А. Система тестов для самоподготовки по курсу «Методы программирования и информатика». Язык Pascal: пособие для студентов

механико-математического факультета специальности 1-31 03 01-02 «Математика (научно-педагогическая деятельность)» / Г. А. Расолько, Е. В. Кремень, Ю. А. Кремень. – Минск : Белорус. гос. ун-т, 2013. – 70 с.

8. Таборовец, В. В. Основы алгоритмизации и программирования : учеб.-метод. комплекс / В. В. Таборовец, Т. В. Русак ; Мин. ин-т упр. – 2-е изд. – Минск : МИУ, 2010. – 436 с.

## ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование раздела, темы	Всего	Лекции	Лабораторные
<b>1.</b>	<b>Основы теории и практики алгоритмизации и языков программирования</b>	<b>4</b>	<b>4</b>	
1.1.	Основы алгоритмизации	2	2	
1.2.	Основы теории языков программирования	1	1	
1.3.	Системы и среды программирования	1	1	
<b>2.</b>	<b>Структурное программирование</b>	<b>64</b>	<b>24</b>	<b>40</b>
2.1.	Основные элементы языка программирования	4	2	2
2.2.	Операторы языка программирования. Базовые алгоритмические конструкции	16	6	10
2.3.	Структурированные (составные, сложные) типы данных. Методы алгоритмизации	22	8	14
2.4.	Подпрограммы	10	4	6
2.5.	Основы работы с файлами	6	2	4
2.6.	Основы программирования графики и звука	6	2	4
<b>3.</b>	<b>Основы современных технологий программирования</b>	<b>82</b>	<b>30</b>	<b>52</b>
3.1.	Парадигма объектно-ориентированного программирования. Платформа .NET	4	4	
3.2.	Основы объектно-ориентированного и компонентно-ориентированного языка программирования C#	8		8
3.3.	Объектно-ориентированное проектирование и программирование	18	10	8
3.4.	Принципы объектно-ориентированного программирования	36	12	24
3.5.	Принципы обобщенного программирования и их реализация	8	4	4
3.6.	Основы компонентно-ориентированного программирования	8		8
	<b>Всего</b>	<b>150</b>	<b>58</b>	<b>92</b>

## **Методические рекомендации по организации и выполнению самостоятельной работы студентов**

Содержание и формы контролируемой самостоятельной работы студентов разрабатываются кафедрами в соответствии с целями и задачами подготовки специалиста.

Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы.

Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

## ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

№ п.п	Название раздела, темы	К-во часов на СРС	Задание	Форма выполнения
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
<b>1.</b>	<b>Основы теории и практики алгоритмизации и языков программирования</b>	<b>8</b>		
<b>1.1.</b>	<b>Основы алгоритмизации</b>	<b>4</b>		
1.1.1	Основы алгоритмизации	4	Блок-схемы алгоритмов, основные элементы. Базовые структуры.	Отчет
<b>1.2.</b>	<b>Основы теории языков программирования</b>	<b>2</b>		
1.2.1	Основы теории языков программирования	2	Парадигма структурного программирования.	Доклад
<b>1.3.</b>	<b>Системы и среды программирования</b>	<b>2</b>		
1.3.1	Системы и среды программирования	2	Назначение, основные функции и состав системы программирования.	Презентация
<b>2.</b>	<b>Структурное программирование</b>	<b>54</b>		
<b>2.1.</b>	<b>Основные элементы языка программирования</b>	<b>4</b>		
2.1.1	Основные элементы языка программирования	4	Структура программы.	Схема
<b>2.2.</b>	<b>Операторы языка программирования. Базовые алгоритмические конструкции</b>	<b>12</b>		
2.2.1	Операторы языка программирования	4	Алгоритмическая конструкция «Следование». Составной оператор.	Презентация
2.2.3	Алгоритмическая конструкция «Ветвление»	4	Оператор выбора	Презентация
2.2.5	Алгоритмическая конструкция «Повторение»	4	Вложенные циклы	Презентация
<b>2.3.</b>	<b>Структурированные (составные, сложные) типы данных. Методы алгоритмизации</b>	<b>22</b>		
2.3.1	Массивы	3	Главная и побочная диагонали двумерного массива.	Презентация
2.3.3	Методы алгоритмизации	3	Циклические перестановки элементов массива: сущность.	Доклад
2.3.4	Методы поиска элемента массива	3	Бинарный поиск.	Доклад

1	2	3	4	5
2.3.6	Методы упорядочивания и перестановки элементов массива	3	Циклические перестановки элементов массива.	Презентация
2.3.9	Строковые величины	4	Процедуры и функции работы со строковыми величинами.	Доклад
2.3.11	Множества	3	Описание множеств.	Презентация
2.3.13	Записи	3	Оператор with.	Презентация
<b>2.4.</b>	<b>Подпрограммы</b>	<b>8</b>		
2.4.1	Подпрограммы	2	Локальные и глобальные переменные	Доклад
2.4.2	Процедуры и функции пользователя	4	Функции пользователя. Структура функции пользователя. Организация вызова функции пользователя.	Презентация
2.4.5	Понятия итерации и рекурсии	2	Рекурсивные подпрограммы	Доклад
<b>2.5.</b>	<b>Основы работы с файлами</b>	<b>4</b>		
2.5.1	Файлы	4	Виды файлов	Презентация
<b>2.6.</b>	<b>Основы программирования графики и звука</b>	<b>4</b>		
2.6.1	Основы программирования графики и звука	4	Процедуры и функции для работы со звуком.	Доклад
<b>2 семестр</b>		<b>62</b>		
<b>3.</b>	<b>Основы современных технологий программирования</b>	<b>70</b>		
<b>3.1.</b>	<b>Парадигма объектно-ориентированного программирования. Платформа .NET</b>	<b>8</b>		
3.1.1	Объектно-ориентированные языки и среды программирования	2	Объектно-ориентированные среды программирования	Доклад
3.1.2	Философия и технология .NET.	4	Базовые классы .NET.	Доклад
3.1.3	Модель выполнения приложений в среде .NET Framework.	2	Сборка	Презентация
<b>3.3.</b>	<b>Объектно-ориентированное проектирование и программирование</b>	<b>22</b>		
3.3.1	Понятие объектно-ориентированного программирования. Объектные модели, принципы их построения, особенности реализации	2	Характеристики объектно-ориентированного программирования	Доклад

1	2	3	4	5
3.3.2	Классы в C#	2	Спецификаторы доступа к членам класса.	Отчет
3.3.3	Конструкторы	3	Создание экземпляра класса.	Отчет
3.3.5	Типы значений. Ссылочные типы	3	Сбор «мусора» и использование деструкторов.	Презентация
3.3.6	Определение пользовательских методов класса	4	Возвращение методами массивов.	Отчет
3.3.8	Общие (статические) члены класса	4	Ограничения на static-методы.	Отчет
3.3.11	Цикл «жизни» переменных	2	Преобразование и приведение типов.	Доклад
3.3.12	Исключительные ситуации	2	Виды ситуаций	Обзор
<b>3.4.</b>	<b>Принципы объектно-ориентированного программирования</b>	<b>28</b>		
3.4.1	Перегрузка методов и конструкторов как механизм реализации полиморфизма	4	Класс «Стек из символов».	Презентация
<b>3 семестр</b>		<b>34</b>		
3.4.7	Перегрузка операторов (операций)	4	Перегрузка унарных операций	Отчет
3.4.8	Индексаторы. Одномерные индексаторы	6	Виртуальный массив.	Доклад
3.4.9	Наследование, базовые и производные классы	8	Абстрактные классы, переопределение абстрактных методов.	Доклад
3.4.15	Интерфейсы и множественное наследование	6	Некоторые стандартные интерфейсы среды .NET Framework.	Доклад
<b>3.5.</b>	<b>Принципы обобщенного программирования и их реализация</b>	<b>12</b>		
3.5.1	Понятие обобщенного программирования	3	Принцип абстрагирования	Доклад
3.5.2	Коллекции и наборы	3	Классы коллекций общего назначения.	Обзор
3.5.4	Динамическая идентификация типов	3	Использование операторов as, typeof.	Отчет
3.5.6	Обобщения	3	Обобщения	Доклад
<b>4 семестр</b>		<b>36</b>		
<b>Итого:</b>		<b>132</b>		

## ПЕРЕЧЕНЬ СРЕДСТВ ДИАГНОСТИКИ

Основным средством диагностики усвоения знаний, умений и овладения необходимыми навыками по учебной дисциплине являются:

– *фронтальный опрос* на лекционных занятиях, направлен систематизацию знаний студентов, определение уровня готовности аудитории к восприятию нового материала, а также на формирование у преподавателя представление об усвоении студентами основополагающих понятий и фактов изучаемой учебной дисциплины;

– *проверка заданий* разнообразного типа (рецептивные, репродуктивные, продуктивные, творческие), выполняемых в рамках часов, отводимых на учебные занятия (практические, семинарские, лабораторные), представляет собой диагностику систематичности подготовки студентов к занятиям, уровень усвоения ими практико-ориентированного содержания программного материала учебной дисциплины;

– *групповые и индивидуальные консультации студентов* предназначены для диагностики уровня овладения определенными знаниями, умениями и навыками, как теоретического материала, так и практического; устранения типичных ошибок и пробелов в знаниях обучающихся;

– *самостоятельные работы* используются для определения индивидуальных особенностей, темпа продвижения студентов и усвоения ими необходимых знаний;

– *компьютерное тестирование* позволяет относительно быстро провести диагностику усвоения студентами учебного материала как по отдельным темам и разделам учебной дисциплины, так и по учебной дисциплине в целом;

– *контрольная работа* используется для осуществления промежуточной диагностики усвоения учащимися содержания учебной дисциплины и, как правило содержит материалы из различных тем, направленные для определения имеющегося уровня знаний, умений и навыков студентов;

– *зачет и экзамен* используется для осуществления итоговой диагностики усвоения студентами содержания учебной дисциплины за учебный семестр и за весь период обучения в соответствии с критериями оценки результатов учебной деятельности обучающихся в учреждениях высшего образования по десятибалльной шкале.

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

Название учебной 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)
Информационные технологии в образовании	Кафедра информатики и методов преподавания информатики	Предусмотреть в тематике тем при создании электронных средств обучения темы, связанные с технологиями программирования и методами алгоритмизации	Протокол № 4 от 25.11.2019
Методика преподавания информатики	Кафедра информатики и методов преподавания информатики	Развивать компетенции, полученные во время изучения технологий программирования в русле преемственности со школьными особенностями изучения содержательной линии «Основы алгоритмизации и программирования»	Протокол № 4 от 25.11.2019
Практикум по решению задач по информатике	Кафедра информатики и методов преподавания информатики	Развивать компетенции, полученные во время изучения технологий программирования и методов алгоритмизации	Протокол № 4 от 25.11.2019
Информационные системы и сети	Кафедра информатики и методов преподавания информатики	Предусмотреть при рассмотрении алгоритмических конструкций анализ взаимосвязей их реализации на различных языках и при создании запросов	Протокол № 4 от 25.11.2019