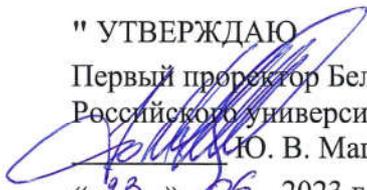


Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет»

" УТВЕРЖДАЮ

Первый проректор Белорусско-  
Российского университета

 Ю. В. Машин

« 23 » 06 2023 г.

Регистрационный № УД-090304/Б.1.06/

**ПРОГРАММИРОВАНИЕ**

(наименование дисциплины)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Направление подготовки** 09.03.04 Программная инженерия

**Направленность (профиль)** Разработка программно-информационных систем

**Квалификация** Бакалавр

	Форма обучения
	Очная
Курс	1
Семестр	1, 2
Лекции, часы	68
Лабораторные занятия, часы	68
Курсовая работа, семестр	2
Экзамен, семестр	1, 2
Контактная работа по учебным занятиям, часы	136
Самостоятельная работа, часы	188
Всего часов / зачетных единиц	324/9

Кафедра-разработчик программы: Программное обеспечение информационных технологий

(название кафедры)

Составители: Сергиенко О.В.

(И.О. Фамилия, ученая степень, ученое звание)

Могилев, 2023

Рабочая программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования - бакалавриат по направлению подготовки 09.03.04 Программная инженерия рег. № 920 от 19.09.2017, учебным планом рег. № 090304-2.1 от 28.04.2023

Рассмотрена и рекомендована к утверждению в качестве рабочего варианта на заседании кафедры «Программное обеспечение информационных технологий» «31» мая 2023 г., протокол № 11.

Зав. кафедрой ПОИТ

 В. В. Кутузов

Одобрена и рекомендована к утверждению Научно-методическим советом Белорусско-Российского университета

« 21 » июня 2023 г., протокол № 6.

Зам. председателя  
Научно-методического совета

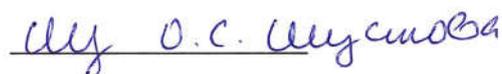
 С.А. Сухоцкий

Рецензент:

С.В. Миренков, начальник управления информационных технологий ОАО «Лента»  
(И.О. Фамилия, должность, ученая степень, ученое звание рецензента)

Рабочая программа согласована:

Ведущий библиотекарь



Начальник учебно-методического  
отдела

 О.Е. Печковская

# 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## 1.1 Цель учебной дисциплины

Целью учебной дисциплины является формирование у студентов базовых знаний программирования, привитие студентам навыков постановки, подготовки и решения задачи на языке высокого уровня, необходимых в качестве фундаментальной основы для изучения последующих дисциплин.

## 1.2 Планируемые результаты изучения дисциплины

В результате освоения учебной дисциплины студент должен

**знать:**

- основы алгоритмизации
- основные конструкции языков высокого уровня;
- терминологию;
- принципы создания программного обеспечения;
- технологии разработки программного обеспечения;
- основные структуры данных;
- базовые концепции объектно-ориентированного программирования;

**уметь:**

- составлять алгоритмы;
- использовать основные конструкции языков высокого уровня;
- реализовывать алгоритмы в виде программ на языке высокого уровня;

**владеть:**

- методами и инструментальными средствами создания программного обеспечения;
- навыками самостоятельной разработки, отладки, тестирования и документирования программы

## 1.3 Место учебной дисциплины в системе подготовки студента

Дисциплина относится к блоку 1 «Дисциплины (модули) (Обязательная часть блока 1)».

Перечень учебных дисциплин, которые будут опираться на данную дисциплину:

- Объектно-ориентированное программирование;
- Тестирование и отладка программного обеспечения.

Кроме того, знания, полученные при изучении дисциплины на лекциях и лабораторных работах будут применены при прохождении при прохождении ознакомительной и технологических практик, а также при подготовке выпускной квалификационной работы и дальнейшей профессиональной деятельности.

## 1.4 Требования к освоению учебной дисциплины

Освоение данной учебной дисциплины должно обеспечивать формирование следующих компетенций:

Коды формируемых компетенций	Наименования формируемых компетенций
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

## 2 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Вклад дисциплины в формирование результатов обучения выпускника (компетенций) и достижение обобщенных результатов обучения происходит путём освоения содержания обучения и достижения частных результатов обучения, описанных в данном разделе.

### 2.1 Содержание учебной дисциплины

1 семестр

Номер темы	Наименование тем	Содержание	Коды формируемых компетенций
1	Введение. Общие сведения об алгоритмах	Предварительные сведения технологиях программирования и методах алгоритмизации. Обзор современных языков и систем программирования. История развития языков программирования.	ОПК-2, ОПК-4, ОПК-6
2	Построение алгоритмов	Алгоритм и его свойства. Разновидности структур алгоритмов. Способы описания алгоритмов. Стандартизация графического представления алгоритмов. Методы разработки и анализа алгоритмов. Общие сведения о структурном программировании. Представление структурированных схем. Примеры вычислительных алгоритмов.	ОПК-2, ОПК-4, ОПК-6
3	Понятие и состав системы программирования	Назначение и состав системы программирования. Классификация языков программирования. Платформа .NET Среда Visual Studio.NET. Консольные приложения	ОПК-2, ОПК-6
4	Основные элементы языка программирования C#	Переменные. Константы. Инициализация переменных. Динамическая инициализация. Жизненный цикл переменной. Область видимости переменной. Арифметические операторы. Операторы отношений и логические операторы. Оператор присваивания. Поразрядные операторы. Операторы сдвига. Приоритет операторов. Выражение, использование выражений. Консольный ввод-вывод данных Управляющие последовательности символов. Встроенные математические функции.	ОПК-2, ОПК-4, ОПК-6
5	Конструкции принятия решений	Оператор if/else Операции отношения и равенства Условная операция Логические операции Оператор	ОПК-2, ОПК-6

		switch	
6	Цикл for	Назначение, формат записи цикла for. Работа инструкции for и ее модификации.	ОПК-2, ОПК-6
7	Циклы while и do-while	Назначение, формат записи инструкций while, do-while. Инструкции break и continue. Вложенные циклы.	ОПК-2, ОПК-6
8	Обработка исключений	Понятие исключения. Основные типы и принципы и обработки исключительных ситуаций. Перехват исключительных ситуаций. Операторы try, throw, catch, finally. Использование нескольких блоков catch и вложенный оператор try. Не перехваченные исключения. Встроенные исключения. Принципы создания и использования исключений.	ОПК-2, ОПК-6
9	Одномерные массивы	Типы значений и ссылочные типы. Структура одномерного массива. Инициализация. Консольный ввод-вывод. Доступ к отдельным элементам одномерного массива. Алгоритм вычисления произведения элементов массива. Алгоритм удаления элементов из массива	ОПК-2, ОПК-6
10	Сортировка данных	Основные алгоритмы сортировки. Использование встроенных возможностей для сортировки данных.	ОПК-2, ОПК-6
11	Двумерные массивы и ступенчатые массивы	Структура двумерного массива. Инициализация. Консольный ввод-вывод. Доступ к отдельным элементам двумерного массива. Использование цикла foreach. Ступенчатые массивы. Многомерные массивы и массивы массивов	ОПК-2, ОПК-6
12	Статические методы	Понятие метода. Сигнатура метода. Список формальных параметров. Последовательность обмена информацией метода с вызвавшим его кодом. Способы передачи параметров. Основные свойства методов. Рекурсивные методы. Организация вызова рекурсивных методов. Методы с переменным числом аргументов. Перегруженные методы. Методы с аргументами по умолчанию. Модификаторы out и ref. Передача массива в метод в качестве аргумента.	ОПК-2, ОПК-4, ОПК-6
13	Перечисления. Кортежи	Понятие перечисления. Формат объявления перечисления. Доступ к членам перечисления. Инициализация перечисления. Указание базового типа перечисления. Использование перечислений при создании интерфейса пользователя в окне консоли. Назначение кортежей. Формат объявления кортежей. Обращение к элементам кортежа. Передача в метод и возврат из метода кортежей.	ОПК-2, ОПК-6
14	Строки и символы	Класс String. Создание строк. Конструкторы класса String. Методы класса String: определение длины строки, копирование и клонирование строк, конкатенация строк, извлечение подстроки, вставка подстроки, удаление символов, разбор строки, сравнение строк. Недостаток объектов класса String. Класс StringBuilder. Создание и использование объектов класса StringBuilder. Основные методы класса StringBuilder. Отличие объектов класса StringBuilder от объектов класса String.	ОПК-2, ОПК-6
15	Структуры	Понятие структуры как типа данных, определяемого пользователем. Формат определения структуры. Члены структуры: поля, методы, свойства, конструкторы. Модификаторы доступа к членам структуры. Отличие структуры от класса. Назначение структур. Способы создания объектов структуры: без вызова конструктора; с помощью операции new. Инициализация полей структуры. Доступ к	ОПК-2, ОПК-6

		членам структуры. Массивы структур. Передача в метод и возврат из метода структур и массивов структур.	
16	Файлы и потоки	Понятие потока и файла. Классы для работы с файлами. Виды файлов (бинарные текстовые, байтовые). Чтение и запись текстовых файлов. Чтение и запись бинарных файлов.	ОПК-2, ОПК-6
17	Коллекции	Назначение коллекций. Различие между коллекциями и массивами. Типы коллекций. Необобщенные и обобщенные коллекции. Пространства имен с коллекциями. Классы ArrayLst, список List<T>, сортированный список SortedList<TKey, TValue>, двухсвязанный список LinkedList<T>, словарь Dictionary<TKey, TValue>, очередь Queue<T>, стек Stack<T>. Назначение перечисленных классов. Основные методы и свойства классов. Создание и использование объектов классов. Организация добавления, удаления, поиска, вставки, прохода по элементам коллекции.	ОПК-2, ОПК-6
18	Основные элементы языка C++	Основные понятия языка. Структура программы. Простые типы данных. Операции и их приоритет. Выражения. Основные операторы. Основные возможности организации ввода/вывода. Стандартные потоки ввода/вывода (ошибок). Примеры вычислительных алгоритмов.	ОПК-2, ОПК-4, ОПК-6
19	Сложные типы данных в языке C++	Массивы. Работа с массивами. Строки. Работа со строками. Структуры данных различного типа. Работа со структурами. Адреса и указатели. Основные возможности работы с динамической памятью. Динамические массивы. Примеры вычислительных алгоритмов. Задачи поиска и сортировки.	ОПК-2, ОПК-6
19	Подпрограммы в языке C++	Модульность в программировании. Понятие и структура подпрограммы. Описание подпрограмм в языках высокого уровня (процедуры, функции). Организация вызова подпрограммы. Типы параметров подпрограммы (функции), локальные и глобальные переменные. Организация многофункциональных программ. Внешние модули. Примеры вычислительных алгоритмов. Рекурсивные алгоритмы.	ОПК-2, ОПК-4, ОПК-6

## 2.2 Учебно-методическая карта учебной дисциплины

1 семестр

№ недели	Лекции (наименование тем)	Часы	Лабораторные занятия	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
Модуль 1							
1	Тема 1 Введение. Общие сведения об алгоритмах	2	Лаб. р. 1. Построение алгоритмов	2	3		
2	Тема 2 Построение алгоритмов	2	Лаб. р. 1. Построение алгоритмов	2	3	ЗЛР	6
3	Тема 3 Понятие и состав системы программирования	2	Лаб. р. 2. Программирование линейных алгоритмов	2	3	ЗЛР	8
4	Тема 4	2	Лаб. р. 3. Программирование раз-	2	3		

	Основные элементы языка программирования C#		ветвляющихся алгоритмов				
5	Тема 5 Конструкции принятия решений	2	Лаб. р. 3. Программирование разветвляющихся алгоритмов	2	3	ЗЛР	8
6	Тема 5 Конструкции принятия решений	2	Лаб. р. 4. Программирование с использованием оператора switch		3		
7	Тема 6 Цикл for	2	Лаб. р. 4. Программирование с использованием оператора switch	2	2	ЗЛР	8
8	Тема 7 Циклы while и do-while	2	Лаб. р. 5. Программирование циклических алгоритмов. Оператор цикла for	2	2	ПКУ	30
Модуль 2							
9	Тема 7 Циклы while и do-while	2	Лаб. р. 5. Программирование циклических алгоритмов. Оператор цикла for	2	2	ЗЛР	5
10	Тема 8 Обработка исключений	2	Лаб. р. 6. Программирование циклических алгоритмов. Операторы while и do...while	2	2	ЗЛР	5
11	Тема 9 Одномерные массивы	2	Лаб. р. 7. Программирование циклических алгоритмов. Вложенные циклы	2	2		
12	Тема 9 Одномерные массивы	2	Лаб. р. 7. Программирование циклических алгоритмов. Вложенные циклы	2	2	ЗЛР	5
13	Тема 10 Сортировка данных	2	Лаб. р. 8. Обработка одномерных массивов. Сортировка массивов	2	2		
14	Тема 10 Сортировка данных	2	Лаб. р. 8. Обработка одномерных массивов. Сортировка массивов	2	2	ЗЛР	5
15	Тема 11 Двумерные массивы и ступенчатые массивы	2	Лаб. р. 9. Обработка массивов как объектов	2	2	ЗЛР	5
16	Тема 11 Двумерные массивы и ступенчатые массивы	2	Лаб. р. 10. Обработка двумерных массивов.	2	2		
17	Тема 11 Двумерные массивы и ступенчатые массивы	2	Лаб. р. 10. Обработка двумерных массивов.	2	2	ЗЛР ПКУ	5 30
18-20					36	ПА (экзамен)	40
	Итого	34		34	76		100

## 2 семестр

№ недели	Лекции (наименование тем)	Часы	Лабораторные занятия	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
Модуль 1							
1	Тема 12 Статические методы	2	Лаб. р. 11. Статические методы	2	2		
2	Тема 12 Статические методы	2	Лаб. р. 11. Статические методы	2	2	ЗЛР	8
3	Тема 13 Перечисления. Кортежи	2	Лаб. р. 12. Массивы как параметры методов	2	2	ЗЛР	8
4	Тема 14 Строки и символы	2	Лаб. р. 13. Символьные и строковые типы	2	2		
5	Тема 14 Строки и символы	2	Лаб. р. 13. Символьные и строковые типы	2	2	ЗЛР	8
6	Тема 15	2	Лаб. р. 14. Решение задач с исполь-	2	2		

	Структуры		зованием структур				
7	Тема 15 Структуры	2	Лаб. р. 14. Решение задач с использованием структур	2	2	ЗЛР	6
8	Тема 16 Файлы и потоки	2	Лаб. р. 15. Решение задач с использованием текстовых файлов	2	2	ЗЛР ПКУ	30
Модуль 2							
9	Тема 16 Файлы и потоки	2	Лаб. р. 16. Решение задач с использованием бинарных файлов	2	2		
10	Тема 17 Коллекции	2	Лаб. р. 16. Решение задач с использованием бинарных файлов	2	2	ЗЛР	6
11	Тема 17 Коллекции	2	Лаб. р. 17. Решение задач с использованием списков	2	2	ЗЛР	6
12	Тема 18 Основные элементы языка C++	2	Лаб. р. 18. Программирование с использованием базовых структур языка C++	2	2		
13	Тема 18 Основные элементы языка C++	2	Лаб. р. 18. Программирование с использованием базовых структур языка C++	2	2	ЗЛР	6
14	Тема 19 Сложные типы данных в языке C++	2	Лаб. р. 19. Программирование с использованием строковых типов данных языка C++	2	2		
15	Тема 19 Сложные типы данных в языке C++	2	Лаб. р. 19. Программирование с использованием строковых типов данных языка C++	2	4	ЗЛР	6
16	Тема 20 Подпрограммы в языке C++	2	Лаб. р. 20. Передача массивов в функции	2	4		
17	Тема 20 Подпрограммы в языке C++	2	Лаб. р. 20. Передача массивов в функции	2	4	ЗЛР ПКУ	6 30
1-17	Выполнение курсового проекта				36		
18-20					36	ПА (экзамен)	40
	Итого	34		34	112		100

Принятые обозначения:

*Текущий контроль* –

ЗЛР – защита лабораторной работы;

ПКУ – промежуточный контроль успеваемости.

ПА - Промежуточная аттестация.

Итоговая оценка определяется как сумма текущего контроля и промежуточной аттестации и соответствует баллам:

Экзамен

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

### 2.3 Требования к курсовой работе

Целями курсовой работы являются:

- углубление, закрепление и систематизация теоретических знаний и практических умений, полученных в процессе обучения;
- оценка степени усвоения студентами основных положений дисциплины;
- приобретение навыков самостоятельной разработки программного обеспечения в интегрированной среде Microsoft Visual Studio.NET;
- получение навыков оформления программной документации и использования справочной литературы, ГОСТов при создании программных продуктов.

Содержание курсовой работы включает две части:

1) теоретическая часть – обзор по теме проектирования, исследование актуальных вопросов в данной области, постановка задач, обоснование принятого решения;

2) практическая – реализация программной системы и ее отладка.

Примерная тематика курсовых работ представлена в приложении 1.

На выполнение курсовой работы отводится 36 часов.

Разбивка этапов курсовой работы, определение количества минимальных и максимальных баллов за каждый из них производится преподавателем. Примерный перечень этапов выполнения курсовой работы и количества баллов за каждый из них представлен в таблице.

№	Этап выполнения	Минимум	Максимум
1	Составление технического задания	6	10
2	Проектирование программного модуля	6	10
3	Разработка документации программного модуля	9	15
4	Разработка программного модуля	9	15
5	Тестирование программного модуля	6	10
	Итого за выполнение курсового проекта	36	60
	Защита курсового проекта	15	40

Итоговая оценка курсовой работы представляет собой сумму баллов за его выполнение и защиту и выставляется в соответствии со шкалой:

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

### 3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изучении дисциплины используется модульно-рейтинговая система оценки знаний студентов. Применение форм и методов проведения занятий при изучении различных тем курса представлено в таблице.

№ п/п	Форма проведения занятия*	Вид аудиторных занятий**			Всего часов
		Лекции	Практические занятия	Лабораторные занятия	
1	Мультимедиа	Темы 1-20			68
2	С использованием ЭВМ			Лаб.р. №1-20	68
3	...				
	ИТОГО	68		68	136

## 4 ОЦЕНОЧНЫЕ СРЕДСТВА

Используемые оценочные средства по учебной дисциплине представлены в таблице и хранятся на кафедре.

№ п/п	Вид оценочных средств*	Количество комплектов
1	Вопросы к экзамену	2
2	Экзаменационные билеты	2
3	Вопросы к лабораторным работам	4
4	Тематика курсового проекта	20

## 5 МЕТОДИКА И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

### 5.1 Уровни сформированности компетенций

№ п/п	Уровни сформированности компетенции	Содержательное описание уровня*	Результаты обучения**
<i>ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности</i>			
<i>ИОПК-2.1. Способен использовать современные информационные технологии, в том числе отечественного производства, при разработке аппаратных и программных систем</i>			
1	Пороговый уровень	Знает современные инструментальные средства разработки программного обеспечения, методики использования программных средств для решения практических задач.	Выполнение лабораторных работ 1-20 и курсового проекта
2	Продвинутый уровень	Умеет применять современные инструментальные средства при разработке программного обеспечения.	Выполнение лабораторных работ 1-20 и курсового проекта
3	Высокий уровень	Способен ставить задачу и разработать алгоритм ее решения, используя прикладные системы программирования, включая объектно-ориентированные.	Выполнение лабораторных работ 1-20 и курсового проекта
<i>ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью</i>			
<i>ИОПК-4.1. Способен применять основные стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы</i>			
1	Пороговый уровень	Знает основы применения стандартов, норм и правил на различных стадиях жизненного цикла информационной системы	Знает названия стандартов оформления технической документации на различных стадиях жизненного цикла информационной системы
2	Продвинутый уровень	Умеет базовые действия по оформлению технической документации на различных стадиях жизненного цикла информационной системы	Способен частично оформлять техническую документацию на различных стадиях жизненного цикла информационной системы
3	Высокий уровень	Способен оформить и оценить результат, оформления технической документации на различных стадиях жизненного цикла информационной	Способен самостоятельно оформлять техническую документацию на различных стадиях жизненного

		системы	цикла информационной системы и оценивать результаты работы
<i>ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов</i>			
<i>ОПК-6.1. Применяет современные языки программирования для проектирования, конструирования и тестирования программных продуктов</i>			
1	Пороговый уровень	Знает основы алгоритмизации задач, понимает основные принципы создания программного обеспечения.	Знает методы разработки схем алгоритмов, понимает основные принципы создания прикладных задач
2	Продвинутый уровень	Умеет использовать базовые программные конструкции при решении прикладных задач.	Способен создавать и отлаживать программный код пользуясь консультацией преподавателя
3	Высокий уровень	Способен оценить результат, полученный в ходе разработки прикладной программы	Способен самостоятельно создавать и отлаживать программы, отвечающие требованию задания

## 5.2 Методика оценки знаний, умений и навыков студентов

Результаты обучения	Оценочные средства*
<i>ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности</i>	
1 Пороговый уровень. Знает принципы и методы разработки прикладных программ в среде Microsoft Visual Studio.	Экзаменационные билеты Вопросы к лабораторным работам
2 Продвинутый уровень. Способен решать задачи из лабораторного практикума под руководством преподавателя.	Экзаменационные билеты Вопросы к лабораторным работам
3 Высокий уровень. Владеет методами объектно-ориентированного и визуального программирования.	Экзаменационные билеты Вопросы к лабораторным работам
<i>ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью</i>	
1 Пороговый уровень. Знает названия стандартов оформления технической документации на различных стадиях жизненного цикла информационной системы	Экзаменационные билеты Вопросы к лабораторным работам
2 Продвинутый уровень. Способен частично оформлять техническую документацию на различных стадиях жизненного цикла информационной системы	Экзаменационные билеты Вопросы к лабораторным работам
3 Высокий уровень. Способен самостоятельно оформлять техническую документацию на различных стадиях жизненного цикла информационной системы и оценивать результаты работы	Экзаменационные билеты Вопросы к лабораторным работам
<i>ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и те-</i>	

<i>стированию программных продуктов</i>	
1 Пороговый уровень. Знает методы разработки схем алгоритмов, понимает основные принципы создания прикладных задач	Экзаменационные билеты Вопросы к лабораторным работам
2 Продвинутый уровень. Способен создавать и отлаживать программный код пользуясь консультацией преподавателя	Экзаменационные билеты Вопросы к лабораторным работам
3 Высокий уровень. Способен самостоятельно создавать и отлаживать программы, отвечающие требованию задания	Экзаменационные билеты Вопросы к лабораторным работам

### 5.3 Критерии оценки лабораторных работ

Каждая лабораторная работа оценивается от 5 до 8 баллов. В оценку включены следующие элементы: выполнение (2 бала), оформление и защита лабораторной работы (3 бала). При отсутствии любого элемента лабораторная работа переходит в разряд задолженности.

Промежуточный рейтинг по лабораторным работам определяется как произведение максимального количества баллов за этот вид занятий на средний балл, полученный при выполнении всех запланированных по модулю работ, деленный на 5.

### 5.4 Критерии оценки курсового проекта

Итоговая оценка курсового проекта представляет собой сумму баллов за его выполнение и защиту и выставляется в соответствии с таблицей, показанной в подразделе 2.3.

### 5.5 Критерии оценки экзамена

На экзамене знания, умения и навыки студентов оценивается по пяти бальной системе следующим образом:

«5» – за глубокое и полное овладение содержанием учебного материала, в котором студент легко ориентируется, владеет понятийным аппаратом, умеет связывать теорию с практикой, способен решать практические задачи, высказывать и обосновывать свои суждения. Отличная отметка предполагает грамотное, логическое изложение ответа (как в устной, так и в письменной форме), качественное внешнее оформление;

«4» – если студент полно освоил учебный материал, владеет понятийным аппаратом, ориентируется в изученном материале, осознанно применяет знания для решения практических задач, грамотно излагает ответ, но содержание, форма ответа имеют отдельные неточности;

«3» – если студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении знаний для решения практических задач, не умеет доказательно обосновать свои суждения;

«2» – если студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал, не может применять знания для решения практических задач;

«1» – за полное незнание и непонимание учебного материала или отказа отвечать.

## 6 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Самостоятельная работа студентов (СРС) направлена на закрепление и углубление освоения учебного материала, развитие практических умений. СРС включает следующие виды самостоятельной работы студентов:

- подготовка к выполнению лабораторных работ и их защите;
- выполнение курсового проекта;
- конспектирование лекций;
- выполнение тестовых заданий.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов приведен в приложении и хранится на кафедре.

Для СРС рекомендуется использовать источники, приведенные в п. 7.

## 7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 7.1 Основная литература

№п/п	Автор, название, место издания, издательство, год издания учебной литературы	Гриф	Кол. экз./URL
1	Бедердинова, О. И. Программирование на языках высокого уровня : учеб. пособие / О.И. Бедердинова, Т.А. Минеева. - М : ИН-ФРА-М, 2019. - 159 с.		<a href="https://znani.um.com/catalog/product/1044396">https://znani.um.com/catalog/product/1044396</a>

### 7.2 Дополнительная литература

№п/п	Автор, название, место издания, издательство, год издания учебной литературы	Гриф	Кол. экз.
1	С# 2005 для профессионалов: [Пер. с англ.] / Нейгел Кристиан, Ивьян Билл, Глинн Джей и др. - М.: Диалектика, 2007. - 1376с.	—	2
2	С#. Начала программирования: учебник / Э. А. Ишкова. - М.: Бином-Пресс, 2010. - 336с.	—	1

### 7.3 Перечень ресурсов сети Интернет по изучаемой дисциплине

[http://professorweb.ru/my/csharp/charp\\_theory/level1/index.php](http://professorweb.ru/my/csharp/charp_theory/level1/index.php)

## **7.4 Перечень наглядных и других пособий, методических рекомендаций по проведению учебных занятий, а также методических материалов к используемым в образовательном процессе техническим средствам**

### **7.4.1 Методические рекомендации**

1 Сергиенко О.В. Вайнилович Ю.В. Методические рекомендации к выполнению лабораторных работ по дисциплине «Программирование», для студентов специальностей 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» дневной формы обучения, БРУ – Могилев 2022.

2 Сергиенко О.В. Вайнилович Ю.В. Методические рекомендации к курсовому проектированию по дисциплине «Программирование» для студентов специальностей 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» дневной формы обучения, БРУ – Могилев 2022.

Электронные варианты методических рекомендаций оформлены в виде файлов и доступны во всех компьютерных классах университета, а также на сайте дистанционного обучения.

### **7.4.2 Информационные технологии**

Мультимедийные презентации по лекционному курсу.

### **7.4.3 Перечень программного обеспечения, используемого в образовательном процессе**

Используется свободно распространяемое программное обеспечение.

- 1 Microsoft Visual.NET.
- 2 Microsoft Word.
- 3 Microsoft Excel.
- 4 Microsoft Visio.

## **8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

Материально-техническое обеспечение дисциплины содержится в паспорте лаборатории «519/2», рег. номер ПУЛ-4 519/2 - 22.