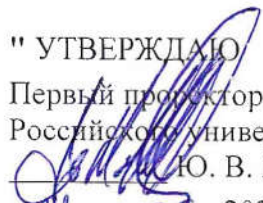


Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

" УТВЕРЖДАЮ

Первый проректор Белорусско-
Российского университета

 О. В. Машин

« 22 » 06 2021 г.

Регистрационный № УД-090301/Б.1.0.15/

ПРОГРАММИРОВАНИЕ

(наименование дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) Автоматизированные системы обработки информации

Квалификация Бакалавр

	Форма обучения
	Очная
Курс	1
Семестр	1, 2
Лекции, часы	68
Лабораторные занятия, часы	68
Курсовой проект, семестр	2
Экзамен, семестр	1, 2
Контактная работа по учебным занятиям, часы	136
Самостоятельная работа, часы	116
Всего часов / зачетных единиц	252/7

Кафедра-разработчик программы: Программное обеспечение информационных техноло-
гий

(наименование кафедры)

Составители: Сергиенко О.В., ст. препод., Н.Н. Горбатенко, канд. техн. наук, доц.

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

Могилев, 2021

Рабочая программа составлена в соответствии с Федеральным Государственным образовательным стандартом высшего профессионального образования по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденным приказом № 929 от 19.09.2017 г., учебным планом рег. № 090301-4, утвержденным 27.12.2019г.

Рассмотрена и рекомендована к утверждению в качестве рабочего варианта на заседании кафедры «Автоматизированные системы управления»
13.05.2021 г., протокол № 9.

Зав. кафедрой ПОИТ

 В. В. Кутузов

Одобрена и рекомендована к утверждению научно-методического совета
Белорусско-Российского университета

16.06.2021 г., протокол № 7.

Зам. председателя
научно-методического совета

 С.А. Сухоцкий

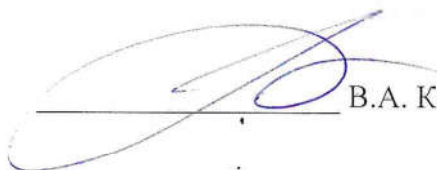
Рецензент:
начальник управления информационных технологий ОАО «Лента» С.В. Миренков

Рабочая программа согласована:

Ведущий библиотекарь

 Е.Н. Киселева

Начальник учебно-методического
отдела

 В.А. Кемова

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:

- Проектирование программного обеспечения;
- 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 проектно-технологической 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-9	Способен осваивать методики использования программных средств для решения практических задач

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	Вводное занятие	Предварительные сведения технологиях программирования и методах алгоритмизации. Обзор современных языков и систем программирования. История развития языков программирования.	ОПК-2, ОПК-9
2	Структура простейшей программы	Линейный алгоритм исполнения программы. Система основных численных и символьных типов языка.	ОПК-2, ОПК-9
3	Литералы	Переменные. Константы. Инициализация переменных. Динамическая инициализация. Жизненный цикл переменной. Область видимости переменной.	ОПК-2, ОПК-6
4	Операторы	Арифметические операторы. Операторы отношений и логические операторы. Оператор присваивания. Поразрядные операторы. Операторы сдвига. Приоритет операторов. Выражение, использование выражений.	ОПК-2, ОПК-9
5	Консольный ввод-вывод данных	Управляющие последовательности символов. Встроенные математические функции.	ОПК-2, ОПК-9
6	Инструкции управления	Инструкция if. Инструкция switch.	ОПК-2, ОПК-9
7	Цикл for	Назначение, формат записи цикла for. Работа инструкции for и ее модификации.	ОПК-2, ОПК-9
8	Циклы while и do-while	Назначение, формат записи инструкций while, do-while. Инструкции break и continue. Вложенные циклы.	ОПК-2, ОПК-9
9	Одномерные массивы	Структура одномерного массива. Инициализация. Консольный ввод-вывод. Доступ к отдельным элементам одномерного массива.	ОПК-2, ОПК-9
10	Основные алгоритмы обработки массивов	Алгоритм вычисления суммы элементов массива. Алгоритм вычисления произведения элементов массива. Алгоритм удаления элементов из массива.	ОПК-2, ОПК-9

11	Сортировка данных	Основные алгоритмы сортировки. Использование встроенных возможностей для сортировки данных.	ОПК-2, ОПК-9
12	Поиск данных в массивах	Двоичный (бинарный) поиск. Метод бисекции.	ОПК-2, ОПК-9
13	Двумерные массивы	Структура двумерного массива. Инициализация. Консольный ввод-вывод. Доступ к отдельным элементам двумерного массива.	ОПК-2, ОПК-9
14	Ступенчатые массивы	Использование цикла foreach. Ступенчатые массивы. Многомерные массивы и массивы массивов	ОПК-2, ОПК-9
15	Подпрограммы (методы)	Метод. Сигнатура метода. Список формальных параметров. Последовательность обмена информацией метода с вызвавшим его кодом. Способы передачи параметров. Основные свойства методов.	ОПК-2, ОПК-9
16	Параметры методов	Функции с переменным числом аргументов. Перегруженные функции. Функции с аргументами по умолчанию. Модификаторы out и ref. Передача массива в функцию в качестве аргумента.	ОПК-2, ОПК-9
17	Рекурсия	Рекурсивные методы. Организация вызова рекурсивных методов.	ОПК-2, ОПК-9

2 семестр

Номер темы	Наименование тем	Содержание	Коды формируемых компетенций
18	Методы и инструменты отладки кода в Visual Studio	Основные виды ошибок в программировании. Синтаксические. Логические. Ошибки, возникающие во время выполнения программы. Использование отладчика Visual Studio для поиска логических ошибок в программе. Точки останова: назучение, установка и удаление. Запуск отладчика. Пошаговая отладка. Команды Step Into, Step Over и Step Out. Просмотр значений переменных. Информация отображаемая в окнах Debug-Windows. Завершение работы с отладчиком.	ОПК-2, ОПК-9
19	Классы и объекты	Понятие класса как типа данных, определяемого пользователем. Формат определения класса. Члены класса: поля, методы, свойства, конструкторы. Модификаторы доступа к членам класса: private, public, protected. Создание объектов класса. Операция new. Действия выполняемые CLR при создании объекта класса. Объекты класса и ссылочный тип данных. Операция доступа к членам объекта. Ключевое слово this. Объекты классов как поля других классов. Объекты классов как параметры методов.	ОПК-2, ОПК-9
20	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 членам структуры. Массивы структур. Передача в метод и возврат из метода структур и массивов структур.	ОПК-2, ОПК-9
21	Перечисления. Кортежи	Понятие перечисления. Формат объявления перечисления. Доступ к членам перечисления. Инициализация перечисления. Указание базового типа	ОПК-2, ОПК-9

		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.	
22	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-9
23	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-9
24	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-9
25	Работа с каталогами. Сериализация	Классы для работа с каталогами. Работа с ZIP-архивами. Сериализация и десериализация объектов.	ОПК-2, ОПК-9
26	Коллекции	Назначение коллекций. Различие между коллекциями и массивами. Типы коллекций. Необобщенные и обобщенные коллекции. Пространства имен с коллекциями. Классы ArrayList, 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-9
27	Этапы разработки программного проекта	Основные этапы разработки программного проекта. Анализ проблемы, определение требований. Проектирование. Реализация (кодирование). Тестирование. Эксплуатация и сопровождение. Процедурный стиль разработки программы. Сущность стиля. Пошаговая детализация, восходящий и нисходящий методы проектирования программ. Методы нисходящего проектирования. Методы восходящего проектирования. Структурная схема программы. Проектирование представления данных. Выбор типа для скалярного данного. Выбор представления наборов данных. Представления наборов, данных в памяти. Хранение наборов данных.	ОПК-2, ОПК-9
28	Юнит-тесты	Назначение юнит-тестов. Этапы создания юнит-теста в среде Visual Studio. Примеры использования юнит-тестов при программировании.	ОПК-2, ОПК-9
29	Windows-Forms приложения	Суть событийно-управляемого программирования и его отличия от классических программ. Структура программы, управляемой событиями.	ОПК-2, ОПК-9

		Основные элементы Windows-Forms приложения. Форма. Окно свойств. Панель инструментов. Обозреватель решений. Окно текста программы. Переключение между формой и тестом программы. Настройка формы. Свойства и события элементов формы. Добавление, установление свойств и обработчиков событий. Элементы управления. Label, TextBox, Button. ListBox, RadioButton, CheckBox, ComboBox, MenuStrip, DataGridView, Chart. Свойства и события элементов управления. Добавление, установление свойств и обработчиков событий. Использование нескольких Windows Forms. Взаимодействие между формами. Способы организации взаимодействия между родительской и дочерней формами.	
30	Алгоритмы поиска в линейных структурах	Постановка задачи поиска в ранее созданном множестве данных. Ключ поиска. Общий алгоритм поиска данных. Алгоритм последовательного поиска.. Схема алгоритма. Реализация алгоритма. Время выполнения алгоритма. Достоинство и недостаток алгоритма. Алгоритм поиска с барьером. Схема алгоритма. Реализация алгоритма. Время выполнения алгоритма. Достоинство и недостаток алгоритма. Алгоритм бинарного (двоичного) поиска. Схема алгоритма. Реализация алгоритма. Время выполнения алгоритма. Достоинство и недостаток алгоритма.	ОПК-2, ОПК-9
31	Алгоритмы сортировки данных	Назначение сортировки. Классификация алгоритмов сортировки. Ключ сортировки. Алгоритм метода пузырька. Схема и реализация алгоритма. Алгоритм шейкер-сортировки. Схема и реализация алгоритма. Алгоритм прямой выборки. Схема и реализация алгоритма. Алгоритм простыми вставками. Схема и реализация алгоритма. Алгоритм Шелла. Схема и реализация алгоритма. Алгоритм бинарной сортировки. Схема и реализация алгоритма.	ОПК-2, ОПК-9

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	1. Вводное занятие	2	Лаб. р. 1. Алгоритмы	2		ЗИЗ	4
2	2. Структура простейшей программы	2	Лаб. р. 1. Алгоритмы	2		ЗИЗ	4
3	3. Литералы	2	Лаб. р. 2. Программирование линейных алгоритмов	2		ЗИЗ	4

4	4. Операторы	2	Лаб. р. 3. Программирование разветвляющихся алгоритмов	2		ЗИЗ	4
5	5. Консольный ввод-вывод данных	2	Лаб. р. 3. Программирование разветвляющихся алгоритмов	2		ЗИЗ	4
6	6. Инструкции управления	2	Лаб. р. 4. Оператор цикла for	2		ЗИЗ	5
7	7. Цикл for	2	Лаб. р. 4. Оператор цикла for	2		ЗИЗ	5
8	8. Циклы while и do-while	2	Лаб. р. 5. Операторы while и do...while	2		ПКУ	30
9	9. Одномерные массивы	2	Лаб. р. 5. Операторы while и do...while	2		ЗИЗ	4
10	10. Основные алгоритмы обработки массивов	2	Лаб. р. 6. Одномерные массивы	2		ЗИЗ	4
11	11 Сортировка данных	2	Лаб. р. 6. Одномерные массивы	2		ЗИЗ	4
12	12 Поиск данных в массивах	2	Лаб. р. 7. Двумерные массивы	2		ЗИЗ	4
13	13 Двумерные массивы	2	Лаб. р. 7. Двумерные массивы	2		ЗИЗ	4
14	14 Ступенчатые массивы	2	Лаб. р. 8. Сортировка и поиск	2		ЗИЗ	4
15	15 Подпрограммы (методы)	2	Лаб. р. 8. Сортировка и поиск	2		ЗИЗ	3
16	16 Параметры методов	2	Лаб. р. 9. Статические методы	2	2	ЗИЗ	3
17	17 Рекурсия	2	Лаб. р. 9. Статические методы	2	2	ПКУ	30
18-20					36	ПА (экзамен)	40
	Итого	34		34	40		100

2 семестр

№ недели	Лекции (наименование тем)	Часы	Лабораторные занятия	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
Модуль 1							
1	18 Методы и инструменты отладки кода в Visual Studio	2	Лаб. р. 10. Разработка программ с использованием классов и объектов	2		ЗИЗ	4
2	19 Классы и объекты	2	Лаб. р. 10. Разработка программ с использованием классов и объектов	2		ЗИЗ	4
3	19 Классы и объекты	2	Лаб. р. 11. Разработка программ с использованием структур	2		ЗИЗ	4
4	20 Структуры	2	Лаб. р. 12. Разработка программ с использованием перечислений. Создание меню пользователя	2		ЗИЗ	4
5	21 Перечисления. Кортежи	2	Лаб. р. 13. Разработка программ с использованием кортежей	2		ЗИЗ	4
6	22 Строки и символы	2	Лаб. р. 14. Работа с символами и строками	2		ЗИЗ	5
7	23 Обработка исключений	2	Лаб. р. 15. Разработка программ с использованием исключений	2		ЗИЗ	5
8	24 Файлы и потоки	2	Лаб. р. 16. Разработка программ с использованием файлов	2		ПКУ	30
Модуль 2							
9	25 Работа с каталогами. Сериализация	2	Лаб. р. 17. Сериализация объектов	2		ЗИЗ	4
10	26 Коллекции	2	Лаб. р. 18. Разработка программ с использованием коллекций	2		ЗИЗ	4
11	27 Этапы разработки программного проекта	2	Лаб. р. 18. Разработка программ с использованием коллекций	2		ЗИЗ	4
12	28 Юнит-тесты	2	Лаб. р. 19. Разработка юнит-тестов в Visual-Studio	2		ЗИЗ	4
13	29 Windows-Forms приложения	2	Лаб. р. 20. Разработка Windows-Forms приложений	2		ЗИЗ	4
14	29 Windows-Forms приложения	2	Лаб. р. 21. Разработка Windows-Forms приложения для работы с табличными данными	2		ЗИЗ	4
15	29 Windows-Forms приложения	2	Лаб. р. 22. Разработка Windows-Forms приложения с несколькими формами	2		ЗИЗ	3

16	30 Алгоритмы поиска в линейных структурах	2	Лаб. р. 23. Разработка программ с использованием алгоритмов поиска в линейных структурах	2	2	ЗИЗ	3
17	31 Алгоритмы сортировки данных	2	Лаб. р. 24. Разработка программ с использованием алгоритмов сортировки данных	2	2	ПКУ	30
1-17	Выполнение курсового проекта				36		
18-20					36	ПА (экзамен)	40
	Итого	34		34	76		100

Принятые обозначения:

Текущий контроль –

ЗИЗ – защита индивидуального задания;

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 курсовому проекту

Целями курсового проекта 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 курсового проекта 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 проектов представлена в приложении 1.

На выполнение курсового проекта (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	Описание предметной области	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 курсового проекта 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	
2	Мультимедиа	Темы 1-31			68
7	С использованием ЭВМ			Лаб.р. №1-24	68
8	Расчетные				
9	...				
	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	Контрольные задания для проведения рейтинг-контроля	4
4	Вопросы к зачетным индивидуальным заданиям	15
5	Тематика курсового проекта	1

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>Индикатор достижения ОПК-2.1. Способен использовать современные программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.</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
<i>ОПК-9 Способен осваивать методики использования программных средств для решения практических задач</i> <i>Индикатор достижения ОПК-9.1. Использует классификацию программных средств и методики их применения для решения практических задач.</i>			
1	Пороговый уровень	Знает современные инструментальные средства разработки программного обеспечения, методики использования программных средств для решения практических задач.	Выполнение лабораторных работ 1-24 и курсового проекта
2	Продвинутый уровень	Умеет применять современные инструментальные средства при разработке программного обеспечения.	Выполнение лабораторных работ 1-24 и курсового проекта
3	Высокий уровень	Способен ставить задачу и разработать алгоритм ее решения, используя прикладные системы программирования, включая объектно-ориентированные.	Выполнение лабораторных работ 1-24 и курсового проекта

5.2 Методика оценки знаний, умений и навыков студентов

Результаты обучения	Оценочные средства*
<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>	
1 Пороговый уровень. Знает методы разработки схем алгоритмов, понимает основные принципы создания прикладных задач	Контрольные вопросы к зачетным индивидуальным заданиям
2 Продвинутый уровень. Способен создавать и отлаживать программный код пользуясь консультацией преподавателя	Контрольные вопросы к зачетным индивидуальным заданиям
3 Высокий уровень. Способен самостоятельно создавать и отлаживать программы, отвечающие требованию задания	Контрольные вопросы к зачетным индивидуальным заданиям
<i>ОПК-9 Способен осваивать методики использования программных средств для решения практических задач</i>	
1 Пороговый уровень. Знает принципы и методы разработки прикладных программ в среде Microsoft Visual Studio.	Контрольные вопросы к зачетным индивидуальным заданиям
2 Продвинутый уровень. Способен решать задачи из лабораторного практикума под руководством преподавателя.	Контрольные вопросы к зачетным индивидуальным заданиям
3 Высокий уровень. Владеет методами объектно-ориентированного и визуального программирования.	Контрольные вопросы к зачетным индивидуальным заданиям

5.3 Критерии оценки лабораторных работ

Каждая лабораторная работа оценивается от 2 до 5 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 индивидуальных заданий

Каждое индивидуальное задание оценивается от 2 до 5 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.

5.5 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.

5.6 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.
1	С#. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. - СПб.: Питер, 2019. - 432с. - (Учебник для вузов).	Доп. МО и науки РФ в качестве учебника для студентов вузов	9
2	С# 2.0. Полное руководство: [Пер. с англ.] / Шилдт Герберт. - М.: ЭКОМ, 2019. - 976с.	-	10

7.2 Дополнительная литература

№п/п	Автор, название, место издания, издательство, год издания учебной литературы	Гриф	Кол. экз.
1	Языки программирования: лабораторный практикум для вузов / П. В. Казаков. - Брянск: БГТУ, 2011. - 84с. - (Информационные системы и технологии).	—	1
2	С# 2005 для профессионалов: [Пер. с англ.] / Нейгел Кристиан, Ивсен Билл, Глинн Джей и др. - М.: Диалектика, 2007. - 1376с.	—	1
3	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

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 Горбатенко Н.Н. Методические рекомендации к выполнению лабораторных работ по дисциплине «Программирование», БРУ, 2017 г.

2 Овсянников К.В. Методические рекомендации к курсовому проектированию по дисциплине «Программирование» для студентов специальности 09.03.01 «Информатика и вычислительная техника», БРУ – Могилев 2017, 27 с.

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 - 20.