

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Университет «Дубна»**

**Филиал «Протвино»
Кафедра «Информационные технологии»**

УТВЕРЖДАЮ
Директор

_____/Евсиков А.А./
подпись *Фамилия И.О.*

« ____ » _____ 2023 г.

Рабочая программа дисциплины

Программирование на языке высокого уровня

(наименование дисциплины)

Направление подготовки (специальность)

09.03.01 Информатика и вычислительная техника

код и наименование направления подготовки (специальности)

Уровень высшего образования

бакалавриат

бакалавриат, магистратура, специалитет

Направленность (профиль) программы (специализация)

«Программное обеспечение вычислительной техники и автоматизированных систем»

Форма обучения

очная, заочная

очная, очно-заочная, заочная

Протвино, 2023

Преподаватель (преподаватели):

Кульман Т.Н. доцент, к.т.н., кафедра информационных технологий

Фамилия И.О., должность, ученая степень, ученое звание, кафедра; подпись

Рабочая программа разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) высшего образования

09.03.01 Информатика и вычислительная техника

(код и наименование направления подготовки (специальности))

Программа рассмотрена на заседании кафедры информационных технологий
(название кафедры)

Протокол заседания № 11 от 20.06.23

Заведующий кафедрой _____ Нурматова Е.В.
(Фамилия И.О., подпись)

СОГЛАСОВАНО

Заведующий выпускающей кафедрой¹ _____
(Фамилия И.О., подпись)

«____» _____ 20__ г.

Эксперт (рецензент):

(Ф.И.О., ученая степень, ученое звание, место работы, должность; если текст рецензии не прилагается – подпись эксперта (рецензента), заверенная по месту работы)

¹ Для обеспечивающих кафедр.

Оглавление

1 Цели и задачи освоения дисциплины	4
2 Место дисциплины в структуре ОПОП	4
3 Планируемые результаты обучения по дисциплине	5
4 Объем дисциплины	6
5 Содержание дисциплины	7
6 Перечень учебно-методического обеспечения	14
7 Фонды оценочных средств по дисциплине	14
8 Ресурсное обеспечение	15
Приложение к рабочей программе дисциплины	19
Фонд оценочных средств	19
I курс, I семестр	25
Балльно-рейтинговая система (для очной формы)	25
Инновационные формы проведения учебных занятий (для очной формы)	26
Вопросы к экзамену по дисциплине «Программированию на языке высокого уровня», семестр I	26
Примеры заданий по теме «Линейные, условные алгоритмы, операторы цикла» для сам. выполнения	27
Примеры заданий по теме «Массивы» для сам. выполнения	28
Примеры заданий по теме «Классы» для сам. выполнения	28
I курс, II семестр	30
Балльно-рейтинговая система (для очной формы)	30
Инновационные формы проведения учебных занятий (для очной формы)	30
Вопросы к экзамену по дисциплине «Программированию на языке высокого уровня», II семестр	31
Примерная тематика курсовых работ	32
Фрагмент создания приложения «Тест»	33
Примеры заданий по теме «Символы и строки» для сам. выполнения	34

1 Цели и задачи освоения дисциплины

Цель преподавания дисциплины «Программирование на языке высокого уровня» является формирование у студентов профессиональных компетенций в соответствии с требованиями ФГОС ВО по направлению подготовки бакалавров 09.03.01 «Информатика и вычислительная техника» с учетом направленности бакалаврской программы – «Программное обеспечение вычислительной техники и автоматизированных систем».

Задачами курса являются:

1. изучение теоретических основ технологии программирования;
2. изучение одного из языков программирования высокого уровня;
3. развитие у студентов алгоритмического мышления, способности формулировать задачи;
4. формирование навыков программирования и создания простых приложений в различных предметных областях.

Особое внимание уделяется объектно-ориентированному подходу в программировании, как наиболее успешному в современном компьютерном мире. При обучении объектно-ориентированному языку программирования студенты осваивают новый способ мышления в программировании, сам подход и вычислительную среду.

В качестве языка программирования выбран язык C#.

Объектами профессиональной деятельности в рамках изучаемой дисциплины являются:

- программное обеспечение средств вычислительной техники и автоматизированных систем (программы, программные комплексы и системы).

2 Место дисциплины в структуре ОПОП

Дисциплина Б1.О.18.01 «Программирование на языке высокого уровня» входит в Блок 1 дисциплин обязательной части учебного плана.

Студенты изучают дисциплину «Программирование на языке высокого уровня» на первом курсе (**I и II семестры**), поэтому программа опирается на знания курса информатики (I семестр) и предметов математического цикла.

Данная дисциплина является предшествующей для следующих дисциплин:

- Интерфейсы информационных систем,
- Компьютерная графика,
- Информационные системы и технологии.

3 Планируемые результаты обучения по дисциплине

Раздел заполняется в соответствии с картами компетенций.

Формируемые компетенции: **ОПК-2; ОПК-8.**

Формируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и формулировка)	Планируемые результаты обучения по дисциплине (модулю)
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-2.1: Ориентируется в спектре операционных систем и программных средств отечественного производства на рынке информационных продуктов	Знает среды Visual Studio Microsoft
		Знает общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии
	ОПК-2.2: Устанавливает оптимальные критерии работы программных средств при решении задач профессиональной деятельности	Знает этапы процесса программирования, базовые алгоритмы
		Умеет реализовывать алгоритмы с применением объектно-ориентированного программирования
	ОПК-2.3: Использует современные информационные технологии сбора, обработки, хранения и поиска информации при решении задач профессиональной направленности	Умеет использовать современные информационные технологии и программные средства в части сбора, обработки, хранения и поиска информации
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1: Выбирает инструментальные средства, языки программирования и технологии обработки данных на начальном этапе разработки программного продукта	Знает алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения (C++, Microsoft Visual Studio)
	ОПК-8.2: Разрабатывает алгоритмы и программные коды программных модулей для практического применения	Умеет составлять алгоритмы и писать коды программ для практического применения
		Умеет работать со свойствами и методами элементов управления при программировании под Windows
	ОПК-8.3: Тестирует работоспособность программ и программных компонентов	Умеет отлаживать коды, тестировать работоспособность программы
		Владеет навыками отладки и тестирования написанной программы
	ОПК-8.4: Применяет языки программирования и современные программные среды разработки информационных систем для решения прикладных задач различных классов	Владеет навыками работы в интегрированной среде разработки приложений
		Владеет навыками программной реализации алгоритмов обработки данных на языке высокого уровня
		Владеет навыками разработки несложных приложений в режиме windows-программирования

Результат обучения сформулирован на основании требований профессиональных стандартов:

- 06.001 «Программист», обобщённая трудовая функция С5 - Интеграция программных модулей и компонент и проверка работоспособности выпусков программного продукта; трудовая функция С/02.5 - Осуществление интеграции программных модулей и компонент и верификации выпусков программного продукта; обобщённая трудовая функция D6 - Разработка требований и проектирование программного обеспечения; трудовая функция D/01.6 - Анализ требований к программному обеспечению;

4 Объем дисциплины

По очной форме обучения

Объем дисциплины (модуля) составляет 7 зачетных единицы, всего **252** часа.

Дисциплина читается на первом курсе 2 семестра (**I и II**):

103 часа составляет контактная работа обучающегося с преподавателем;

68 часов – самостоятельная работа обучающегося;

81 час отводится на контроль.

I семестр

(3 ЗЕ, всего 108 часов):

17 часов – лекционные занятия;

34 часа – практические занятия;

30 час – самостоятельная работа;

27 час – контроль.

Вид контроля – **экзамен**.

II семестр

(4 ЗЕ, всего 144 часа):

17 часов – лекционные занятия;

34 часа – практические занятия;

38 часов – самостоятельная работа;

54 часа – мероприятия текущего контроля успеваемости (**экзамен, курсовая работа**).

По заочной форме обучения

Объем дисциплины составляет **9** зачетных единицы, всего **324** часа.

Дисциплина читается 2 семестра (**I и II**):

14 часов составляет контактная работа обучающегося с преподавателем;

291 час составляет самостоятельная работа обучающегося.

18 часов – контроль.

I семестр:

(4 ЗЕ, всего 144 часа):

4 часа – практические занятия;

131 час – самостоятельная работа;

9 часов – промежуточный контроль – **экзамен**.

II семестр:

(5 ЗЕ, всего 180 часа):

10 часов – практические занятия;

160 часов – самостоятельная работа;

9 часов – мероприятия текущего контроля успеваемости (**экзамен, курсовая работа**).

5 Содержание дисциплины

Очная форма обучения

Лекции представлены в форме презентаций.

Наименование и краткое содержание разделов и тем дисциплины (модуля) Форма промежуточной аттестации по дисциплине (модулю)	Все го (часы)	В том числе:								
		Контактная работа					Самостоятельная работа			
		Лекционные занятия	Семинарские занятия	Практические занятия	Лабораторные занятия	Всего	Выполнение домашних заданий, подготовка к к/р	Подготовка курсовой работы	Всего	
I семестр										
Раздел 1. Введение в C# Цель и задачи курса. Изучение среды. Понятие переменной. Типы данных. Арифметические, логические операторы, операторы отношений. Этапы создания программы. Ввод/вывод на консоль. Форматирование строк.		2		2		4	15		15	
Раздел 2. Алгоритмы, операторы ветвления, циклов и switch Понятие алгоритма. Свойства алгоритма. Способы описания алгоритма (словесный, графический, программный). Описание алгоритма с помощью блок-схемы. Линейные, условные и циклические алгоритмы. Операторы if, switch, операторы циклов (while, do, for).		2		8		10				
Раздел 3. Массивы и их обработка Массивы: одномерные, двумерные, ступенчатые, многомерные. Индексы, доступ к элементам массива. Генератор случайных чисел. Массивы и вложенные циклы. Инициализация массивов. Цикл foreach. Основные элементы класса работы с массивами System.Array.		2		12		14				
Раздел 4. Введение в классы Основные понятия, описание класса, объекты класса, спецификаторы, состав класса. Создание объектов. Методы, свойства, конструкторы. Параметры и аргументы. Сигнатура метода.		2		4		6				

Передача параметров: по ссылке и по значению.									
Раздел 5. Основные концепции объектно-ориентированного программирования Интегрированная среда разработки приложений. Основные возможности интегрированных сред. Понятие оболочки Framework. Возможности этой оболочки в части многоязыкового программирования и переносимости кода. Понятие промежуточного языка. Инкапсуляция. Наследование. Полиморфизм.		2		1		3	15		15
Раздел 6. Символы и строки Подробно о типах char и string. Массивы символов. Методы и свойства классов char и string. Операции над строками. Методы Split(), Join(), Insert(), Remove(), Replace() и др. Класс StringBuilder – строитель строк. Отличие классов String и StringBuilder. Знакомство с классами регулярных выражений.		3		4		7			
Раздел 7. Введение в динамические структуры данных Понятие об абстрактных типах данных. Динамические структуры. Линейные списки. Стеки. Очереди. Деки. Способы реализации. Нелинейные динамические структуры (графы, деревья) – начальные сведения, определения, свойства и применение. Задача о Кенигсбергских мостах.		2		2		4			
Раздел 8. Технология проектирования программ Жизненный цикл программы Стадии жизненного цикла. Качество программного обеспечения (функциональность, надёжность, эффективность, переносимость и др.). Технология проектирования программ. Структурное программирование, его принципы. Теорема о структурировании. Эмерджентность. Модульное программирование.		2		1		3			
Промежуточная аттестация – экзамен (балльно-рейтинговая система) – 27ч.									
Всего:	108	17		34		51	30		30
II семестр									
Раздел 9. Введение в программирование под Windows Основные понятия. Событийно-управляемое программирование. Изучение свойств и методов элементов управления (для каждого элемента управления разрабатывается небольшое приложение).		2		16		18	8	30	38
Раздел 10. Построение интерфейса пользователя. Обработка исключительных ситуаций Объединение ранее изученных понятий на примере разработки приложения. Элементы интерфейса: формы, меню, элементы управления, диалоговые окна. Модальные и немодальные окна. Стандартные диалоговые элементы управления		2		6		8			

(ColorDialog, FontDialog). Класс Application: основные элементы класса. Обработка исключительных ситуаций. Операторы try, catch, finally.									
Раздел 11. Диалоговые формы. Потоки ввода/вывода Создание форм. SDI и MDI интерфейсы. Диалоговые формы и их свойства. Передача данных их одной формы в другую. Понятие потока. Классы для работы с потоками данных.		2		2		4			
Раздел 12. Классы и интерфейсы Понятие свойства. Примеры использования свойств. Наследование. Рассмотрение конкретных приложений.		2		2		4			
Раздел 13. Работа с файлами Пространства имен, связанные с объектами ввода-вывода. Классы для работы с файловой системой. Потоки данных. Примеры ввода/вывода в файлы различных типов. Обмен с внешними устройствами: BinaryReader, BinaryWriter, FileStream, StreamReader, StreamWriter и т.д.		3		2		5			
Раздел 14. Рекурсия Основные понятия, свойства, примеры.		1		2		3			
Раздел 15. Введение в графику Интерфейсы GDI и GDI+. Программное обеспечение видеосистем в Windows. Графический интерфейс пользователя. Изучение различных структур: Point, Size, Color. Взаимосвязь некоторых классов рисования. Контекст устройства. Кисти, перья и их разновидности. Геометрические фигуры и их изображения.		3		2		5			
Раздел 16. Графический интерфейс DataGridView Основные понятия, свойства и поведение DataGridView. Источники данных. Привязка к источнику данных через класс BindingSource. Рассмотрение примера.		2		2		4			
Курсовая работа	25 час. внеаудиторной контактной работы (вне расписания учебных занятий)								
Промежуточная аттестация – экзамен и курсовая работа, 54 часа (контроль) (балльно-рейтинговая система)									
Всего: 144 = 51 + 38 + 54 (контроль)	144	17		34		51	8	30	38
Итого: 252 = 34 + 68 + 102 + 81(контроль)	252	34		68		102	68		68

При реализации дисциплины (модуля) организуется практическая подготовка путем проведения практических занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью (68 часов).

Практическая подготовка также включает в себя отдельные занятия лекционного типа, которые предусматривают передачу учебной информации обучающимся, необходимой для последующего выполнения работ, связанных с будущей профессиональной деятельностью (34 часа).

Практическая подготовка при изучении дисциплины реализуется непосредственно в университете (филиале).

Заочная форма обучения

Лекционные занятия не предусмотрены

Наименование и краткое содержание разделов и тем дисциплины (модуля) Форма промежуточной аттестации по дисциплине (модулю)	Все го (часы)	В том числе:							
		Контактная работа					Самостоятельная работа		
		Лекционные занятия	Семинарские занятия	Практические занятия	Лабораторные занятия	Всего	Изучение теории, выполнение домашних заданий	Подготовка курсовой работы	Всего
I семестр									
Раздел 1 . Введение в C# Цель и задачи курса. Изучение среды. Понятие переменной. Типы данных. Арифметические, логические операторы, операторы отношений. Этапы создания программы. Ввод/вывод на консоль. Форматирование строк.							110		110
Раздел 2. Алгоритмы, операторы ветвления, циклов и switch Понятие алгоритма. Свойства алгоритма. Способы описания алгоритма (словесный, графический, программный). Описание алгоритма с помощью блок-схемы. Линейные, условные и циклические алгоритмы. Операторы if, switch, операторы циклов (while, do, for).				1		1			
Раздел 3. Массивы и их обработка Массивы: одномерные, двумерные, ступенчатые, многомерные. Индексы, доступ к элементам массива. Генератор случайных чисел. Массивы и вложенные циклы. Инициализация массивов. Цикл foreach. Основные элементы класса работы с массивами System.Array.				1		1			
Раздел 4. Введение в классы Основные понятия, описание класса, объекты класса, спецификаторы, состав класса. Создание объектов. Методы, свойства, конструкторы. Параметры и аргументы. Сигнатура метода. Передача параметров: по ссылке и по значению.				2		2			
Раздел 5. Основные концепции объектно-ориентированного программирования Интегрированная среда разработки приложений. Основные возможности							21		21

интегрированных сред. Понятие оболочки Framework. Возможности этой оболочки в части многоязыкового программирования и переносимости кода. Понятие промежуточного языка. Инкапсуляция. Наследование. Полиморфизм.									
Раздел 6. Введение в динамические структуры данных Понятие об абстрактных типах данных. Динамические структуры. Линейные списки. Стеки. Очереди. Деки. Способы реализации. Нелинейные динамические структуры (графы, деревья) – начальные сведения, определения, свойства и применение. Задача о Кенигсбергских мостах.									
Раздел 7. Технология проектирования программ Жизненный цикл программы Стадии жизненного цикла. Качество программного обеспечения (функциональность, надёжность, эффективность, переносимость и др.). Технология проектирования программ. Структурное программирование, его принципы. Теорема о структурировании. Модульное программирование.									
Промежуточная аттестация – экзамен 9 ч.									
Всего: 144= 4 + 131 + 9	144			4		4	131		131

II семестр									
Раздел 8. Введение в программирование под Windows Основные понятия. Событийно-управляемое программирование. Изучение свойств и методов элементов управления							80	80	160
Раздел 9. Построение интерфейса пользователя. Обработка исключительных ситуаций. Элементы интерфейса: формы, меню, элементы управления, диалоговые окна. Модальные и немодальные окна. (ColorDialog, FontDialog).				3		3			
Раздел 10. Диалоговые формы. Потоки ввода/вывода Создание форм. Диалоговые формы и их свойства. Передача данных их одной формы в другую. Понятие потока. Классы для работы с потоками данных.				3		3			
Раздел 11. Классы Наследование. Рассмотрение конкретных приложений.				1		1			
Раздел 12. Символы и строки Подробно о типах char и string. Массивы символов. Методы и свойства классов char и string. Операции над строками. Методы Split(), Join(), Insert(), Remove(), Replace() и др. Класс StringBuilder – построитель строк.				3		3			
Раздел 13. Работа с файлами Классы для работы с файловой системой. Потоки данных. Примеры ввода/вывода в файлы различных типов.									
Раздел 14. Графический интерфейс DataGridView Основные понятия, свойства и поведение DataGridView. Источники данных. Привязка к источнику данных через класс BindingSource.									
Раздел 15. Рекурсия Основные понятия, свойства, примеры.									
Контроль – экзамен, курсовая работа (9 часов)									
Всего: 180 = 10 + 160 + 9 (контроль) + 1	180			10		10	80	80	160
Итого: 324 = 144 + 180	324					14			191

6 Перечень учебно-методического обеспечения

Методические указания к практическим занятиям

1 семестр

1. Изучения среды программирования. Понятие алгоритма. Линейные, разветвляющиеся, циклические алгоритмы. Изучение панелей, меню, диалоговых окон. Рассмотрение простых программ.
2. Описание языка. Понятие переменной. Типа данных. Изучение основных операторов языка программирования. Арифметические операции. Ввод/вывод на консоль. Форматирование строк. Решение задач с условными и циклическими операторами.
3. Изучение классов, создание классов, создание объектов. Методы, свойства, конструкторы. Рассмотрение примеров.
4. Массивы и их обработка. Работа с массивами (одномерными, двумерными и динамическими). Индексы, доступ к элементам массива. Массивы и вложенные циклы.

2 семестр

5. Введение в программирование под Windows. Изучение основных элементов управления, реализация программного кода для этих элементов и разработка приложений, демонстрирующих их использование.
6. Построение интерфейса пользователя. Разработка приложений, таких как “Интернет-магазин”, “Тест” и др. Использование стандартных диалоговых элементов управления. Выбор тем для курсовых работ.
7. Символы и строки. Операции над строками. Методы Split(), Join(), Insert(), Remove(), Replace() и др. Решение задач.
8. Разбор и реализация приложения, содержащего много форм, на примере разработки приложения “Тест”.
9. Консультации по курсовой работе.

Методические указания по освоению дисциплины и для самостоятельной работы обучающихся

1. Автором подготовлен курс лекций по дисциплине и размещён на университетском сервере.
2. Подготовлено методическое пособие по написанию и оформлению курсовой работы.
3. На сервере выложены задачи по каждой теме, решения отдельных задач, а также проекты.

Применяемые образовательные технологии для различных видов учебных занятий и для контроля освоения обучающимися запланированных результатов обучения

Перечень обязательных видов учебной работы студента:

- самостоятельное изучение теории;
- ответы на теоретические вопросы на практических занятиях;
- решение практических задач и заданий на практических занятиях;
- выполнение домашних работ (выдаются индивидуально);
- выполнение курсовой работы.

7 Фонды оценочных средств по дисциплине

Для аттестации обучающихся на соответствие их персональных достижений поэтапным требованиям образовательной программы по дисциплине разработаны фонды оценочных средств, позволяющие оценить результаты обучения (знания, умения, навыки) и сформированные (формируемые) компетенции.

Эти фонды включают теоретические вопросы, типовые практические задания, контрольные работы, домашние работы, тесты и иные оценочные материалы, используемые при проведении процедур текущего контроля успеваемости и промежуточной аттестации.

Фонды оценочных средств представлены в приложении к рабочей программе.

При необходимости обучающиеся с ограниченными возможностями здоровья и инвалиды обеспечиваются оценочными материалами в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

8 Ресурсное обеспечение

– Перечень литературы

Основная учебная литература

1. Гуриков, С. Р. Введение в программирование на языке Visual C# : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 447 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-458-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1092167> (дата обращения: 21.04.2023). — Режим доступа: по подписке.
2. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/450868> (дата обращения: 15.03.2024).

Дополнительная учебная литература

1. Биллиг, В. А. Основы программирования на C#3.0: ядро языка : учебное пособие : [16+] / В. А. Биллиг. — 2-е изд., испр. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016. — 411 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=428947> (дата обращения: 17.04.2023). — ISBN 978-5-9963-0259-8. — Текст : электронный.
2. Биллиг, В. А. Объектное программирование в классах на C# 3.0 : [16+] / В. А. Биллиг. — 2-е изд., испр. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016. — 391 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=428945> (дата обращения: 21.04.2023). — Текст : электронный.
3. Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512404> (дата обращения: 16.04.2023).
4. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2020. — 200 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-680-3. - Текст : электронный.

- URL: <https://znanium.com/catalog/product/1069921> (дата обращения: 21.04.2023). – Режим доступа: по подписке.

5. Кульман, Т.Н. Подготовка и оформление курсовой работы по дисциплине «Программирование на языке высокого уровня»: учебно-методическое пособие / Т.Н. Кульман, М.М. Губаева, М.П. Астафьева, – Дубна: Междунар. ун-т природы, общества и человека «Дубна»; 2009. – 34с.
6. Кальтин, Н. Microsoft Visual C# в задачах и примерах. – СПб.: БХВ-Петербург, 2009. – 320 с.: ил. + CD-ROM
7. Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов. / Т.А. Павловская. – СПб.: Питер, 2010. – 432 с.: ил.

– Периодические издания

1. Информационные технологии и вычислительные системы / Учредитель Федеральное государственное учреждение "Федеральный исследовательский центр "Информатика и управление" РАН"; гл. ред. С.В. Емельянов, - М.: Федеральное государственное учреждение "Федеральный исследовательский центр "Информатика и управление" РАН". Год основания 1995 г. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: <https://elibrary.ru/contents.asp?titleid=8746>
2. Информация и безопасность / учредители: ФГБОУ Воронежский государственный технический университет; гл. ред. А.Г. Остапенко. – Воронеж.: Воронежский государственный технический университет. Журнал основан в 1998 году. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: <http://elibrary.ru/contents.asp?titleid=8748>
3. Программные продукты и системы / учредители: МНИИПУ (г.Москва), гл. редакция международного журнала «Проблемы теории и практики управления» (г. Москва), ЗАО НИИ «Центрпрограммсистем» (г. Тверь); гл. ред. С.В. Емельянов. – Тверь.: НИИ «Центрпрограммсистем». Журнал основан в 1995 году. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: <https://elibrary.ru/contents.asp?titleid=9834>; Сайт журнала www.swsys.ru
4. Системный администратор / учредитель и издатель: Общество с ограниченной ответственностью "Издательский дом "Положевец и партнеры" гл. ред. Г. Положевец. – М.: Общество с ограниченной ответственностью "Издательский дом "Положевец и партнеры" Журнал основан в 2002 году. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: https://elibrary.ru/title_about.asp?id=9973

– Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Электронно-библиотечные системы и базы данных

1. ЭБС «Znanium.com»: <http://znanium.com/>
2. ЭБС «Лань»: <https://e.lanbook.com/>
3. ЭБС «Юрайт»: <https://urait.ru/>
4. ЭБС «Университетская библиотека онлайн»: <http://biblioclub.ru/>
5. Научная электронная библиотека (ПУНЭБ) «eLIBRARY.RU»: <http://elibrary.ru>
6. Национальная электронная библиотека (НЭБ): <http://нэб.рф/>
7. Базы данных российских журналов компании «East View»: <https://dlib.eastview.com/>

Научные поисковые системы

1. ArXiv.org - научно-поисковая система, специализируется в областях: компьютерных наук, астрофизики, физики, математики, квантовой биологии. <http://arxiv.org/>

2. Google Scholar - поисковая система по научной литературе. Включает статьи крупных научных издательств, архивы препринтов, публикации на сайтах университетов, научных обществ и других научных организаций. <https://scholar.google.ru/>
3. SciGuide - навигатор по зарубежным научным электронным ресурсам открытого доступа. <http://www.prometeus.nsc.ru/sciguide/page0601.ssi>

Профессиональные ресурсы сети «Интернет»

1. Открытое образование <https://openedu.ru/>
2. Проект Инициативного Народного Фронта Образования - ИНФО-проект. Школа программирования Coding Craft <http://codingcraft.ru/>.
3. Портал Life-prog <http://life-prog.ru/>.
4. OpenNet www.opennet.ru.
5. Алгоритмы, методы, программы algotlist.manual.ru.

Необходимое программное обеспечение

Дисциплина обеспечена необходимым лицензионным программным обеспечением, которое находится в свободном доступе (Microsoft Visual Studio 2022).

Все лекции по дисциплине разработаны в форме презентаций и выложены на сервер для работы студентов.

Необходимое материально-техническое обеспечение

Проведение практических занятий по дисциплине предполагает использование специализированных аудиторий, оснащенных персональными компьютерами, объединенными в локальную сеть и имеющих доступ к ресурсам глобальной сети Интернет.

Для выполнения заданий самостоятельной подготовки обучающиеся обеспечиваются литературой, лекциями, электронными методическими материалами, подготовленными преподавателем, а также в определённом порядке могут получать доступ к информационным ресурсам Интернета.

Проектор для проведения лекций.

Компьютерный класс (15 ПК) (оборудование в собственности)

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья могут использовать специализированное программное и материально-техническое обеспечение:

- обучающиеся с нарушениями опорно-двигательного аппарата при необходимости могут использовать адаптивные технические средства: специально оборудованные джойстики, увеличенные выносные кнопки, клавиатуры с большими клавишами.
- обучающиеся с ограничениями по зрению могут прослушать доступный аудиоматериал или прочитать тексты, увеличив шрифт на экране монитора компьютера. Рекомендуются использовать экранную лупу и другие визуальные вспомогательные средства, чтобы изменить шрифт текста, межстрочный интервал, синхронизацию с речью и т.д., программы экранного доступа (скринридеры для прочтения текстовой информации через синтезированную речь) и/или включить функцию «экранного диктора» на персональном компьютере с операционной системой Windows 7, 8, 10.
- обучающиеся с ограничениями по слуху могут воспользоваться компьютерной аудиогарнитурой при прослушивании необходимой информации и портативной индукционной системой серии «ИСТОК».

При необходимости обучающиеся с ограниченными возможностями здоровья и инвалиды обеспечиваются печатными и (или) электронными образовательными ресурсами (образовательная программа, учебные пособия и др.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
 - в форме электронного документа,
- в форме аудиофайла.

Приложение к рабочей программе дисциплины

Фонд оценочных средств

В результате освоения дисциплины «Программирование на языке высокого уровня» программы бакалавриата по направлению подготовки 09.03.01 «Информатика и вычислительная техника» с учетом направленности программы – «Программное обеспечение вычислительной техники и автоматизированных систем» выпускник должен обладать следующими профессиональными компетенциями:

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-8: *Способен разрабатывать алгоритмы и программы, пригодные для практического применения*

Описание показателей и критериев оценивания компетенций, а также шкал оценивания

ИНДИКАТОР ДОСТИЖЕНИЯ КОМПЕТЕНЦИИ (код и наименование)	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ ШКАЛА оценивания				
	1	2	3	4	5
ОПК-2: <i>Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности</i>					
ОПК-2.1: Ориентируется в спектре операционных систем и программных средств отечественного производства на рынке информационных продуктов					
Знает среду Visual Studio Microsoft	Отсутс твие знаний	Не знает или слабо знает основные теоретические вопросы. Допускает множественные грубые ошибки	Удовлетворительно знает основные теоретические вопросы. Допускает достаточно серьезные ошибки.	Хорошо знает основные основные теоретические вопросы. Допускает отдельные негрубые ошибки	Хорошо знает основные теоретические вопросы. Не допускает ошибок

Знает общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии	Отсутствие знаний	Не знает или слабо знает общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии. Допускает множественные грубые ошибки	Удовлетворительно знает общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии. Допускает достаточно серьезные ошибки.	Хорошо знает общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии. Допускает отдельные негрубые ошибки	Хорошо знает общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии. Не допускает ошибок
ОПК-2.2: Устанавливает оптимальные критерии работы программных средств при решении задач профессиональной деятельности					
Знает этапы процесса программирования, базовые алгоритмы	Отсутствие умения	Не знает или слабо знает этапы процесса программирования, базовые алгоритмы. Допускает множественные грубые ошибки	Удовлетворительно знает этапы процесса программирования, базовые алгоритмы. Допускает достаточно серьезные ошибки.	Хорошо знает этапы процесса программирования, базовые алгоритмы. Допускает отдельные негрубые ошибки	Хорошо знает этапы процесса программирования, базовые алгоритмы. Не допускает ошибок
Умеет реализовывать алгоритмы с применением объектно-ориентированного программирования (ООП)	Отсутствие умения	Демонстрирует частичное умение реализовывать алгоритмы с применением ООП. Допускает множественные грубые ошибки	Демонстрирует удовлетворительное умение реализовывать алгоритмы с применением ООП. Допускает достаточно серьезные ошибки.	Демонстрирует достаточно устойчивое умение реализовывать алгоритмы с применением ООП. Допускает отдельные негрубые ошибки	Демонстрирует устойчивое умение реализовывать алгоритмы с применением ООП. Не допускает ошибок.
ОПК-2.3: Использует современные информационные технологии сбора, обработки, хранения и поиска информации при решении задач профессиональной направленности					

Умеет использовать современные информационные технологии и программные средства в части сбора, обработки, хранения и поиска информации	Отсутствие умения	Демонстрирует частичное умение использовать современные информационные технологии и программные средства. Допускает множественные грубые ошибки	Демонстрирует удовлетворительное умение использовать современные информационные технологии и программные средства. Допускает достаточно серьезные ошибки.	Демонстрирует достаточно устойчивое умение использовать современные информационные технологии и программные средства. Допускает отдельные негрубые ошибки	Демонстрирует устойчивое умение использовать современные информационные технологии и программные средства. Не допускает ошибок.
Владеет навыками применения современных информационных технологий и программных средств при решении задач программирования	Отсутствие владения	Не владеет или демонстрирует низкий уровень владения навыками применения современных информационных технологий и программных средств. Допускает множественные грубые ошибки	Демонстрирует удовлетворительный уровень владения навыками применения современных информационных технологий и программных средств. Допускает достаточно серьезные ошибки.	Демонстрирует хороший уровень владения навыками применения современных информационных технологий и программных средств. Допускает отдельные негрубые ошибки	Демонстрирует высокий уровень владения навыками применения современных информационных технологий и программных средств. Не допускает ошибок.
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения					
ОПК-8.1: Выбирает инструментальные средства, языки программирования и технологии обработки данных на начальном этапе разработки программного продукта					
Знает алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения (C++, Microsoft Visual Studio)	Отсутствие знаний	Не знает или слабо знает основные теоретические вопросы. Допускает множественные грубые ошибки	Удовлетворительно знает основные теоретические вопросы. Допускает достаточно серьезные ошибки.	Хорошо знает основные теоретические вопросы. Допускает отдельные негрубые ошибки	Хорошо знает основные теоретические вопросы. Не допускает ошибок

ОПК-8.2: Разрабатывает алгоритмы и программные коды программных модулей для практического применения					
Умеет составлять алгоритмы и писать коды программ для практического применения	Отсутствие умения	Демонстрирует частичное умение составлять алгоритмы и писать коды программ. Допускает множественные грубые ошибки	Демонстрирует удовлетворительное умение составлять алгоритмы и писать коды программ. Допускает достаточно серьезные ошибки.	Демонстрирует достаточно устойчивое умение составлять алгоритмы и писать коды программ. Допускает отдельные негрубые ошибки	Демонстрирует устойчивое умение составлять алгоритмы и писать коды программ. Не допускает ошибок.
Умеет работать со свойствами и методами элементов управления при программировании под Windows	Отсутствие умения	Демонстрирует частичное умение работать со свойствами и методами элементов управления. Допускает множественные грубые ошибки	Демонстрирует удовлетворительное умение работать со свойствами и методами элементов управления. Допускает достаточно серьезные ошибки.	Демонстрирует достаточно устойчивое умение работать со свойствами и методами элементов управления. Допускает отдельные негрубые ошибки	Демонстрирует устойчивое умение работать со свойствами и методами элементов управления. Не допускает ошибок.
ОПК-8.3: Тестирует работоспособность программ и программных компонентов					
Умеет отлаживать коды, тестировать работоспособность программы	Отсутствие умения	Демонстрирует частичное умение отлаживать коды, тестировать работоспособность программы. Допускает множественные грубые ошибки	Демонстрирует удовлетворительное умение отлаживать коды, тестировать работоспособность программы. Допускает достаточно серьезные ошибки.	Демонстрирует достаточно устойчивое умение отлаживать коды, тестировать работоспособность программы. Допускает отдельные негрубые ошибки	Демонстрирует устойчивое умение отлаживать коды, тестировать работоспособность программы. Не допускает ошибок.

Владеет навыками отладки и тестирования написанной программы	Отсутствие владения	Не владеет или демонстрирует низкий уровень владения навыками отладки и тестирования написанной программы. Допускает множественные грубые ошибки	Демонстрирует удовлетворительный уровень владения навыками отладки и тестирования написанной программы. Допускает достаточно серьезные ошибки.	Демонстрирует хороший уровень владения навыками отладки и тестирования написанной программы. Допускает отдельные негрубые ошибки	Демонстрирует высокий уровень владения навыками отладки и тестирования написанной программы. Не допускает ошибок.
ОПК-8.4: Применяет языки программирования и современные программные среды разработки информационных систем для решения прикладных задач различных классов					
Владеет навыками работы в интегрированной среде разработки приложений	Отсутствие владения	Не владеет или демонстрирует низкий уровень владения навыками работы в интегрированной среде разработки приложений. Допускает множественные грубые ошибки	Демонстрирует удовлетворительный уровень владения навыками работы в интегрированной среде разработки приложений. Допускает достаточно серьезные ошибки.	Демонстрирует хороший уровень владения навыками работы в интегрированной среде разработки приложений. Допускает отдельные негрубые ошибки	Демонстрирует высокий уровень владения навыками работы в интегрированной среде разработки приложений. Не допускает ошибок.
Владеет навыками программной реализации алгоритмов обработки данных на языке высокого уровня	Отсутствие владения	Не владеет или демонстрирует низкий уровень владения навыками программной реализации алгоритмов. Допускает множественные грубые ошибки	Демонстрирует удовлетворительный уровень владения навыками программной реализации алгоритмов. Допускает достаточно серьезные ошибки.	Демонстрирует хороший уровень владения навыками программной реализации алгоритмов. Допускает отдельные негрубые ошибки	Демонстрирует высокий уровень владения навыками программной реализации алгоритмов. Не допускает ошибок.

Владеет навыками разработки несложных приложений в режиме windows-forms	Отсутствие владения	Не владеет или демонстрирует низкий уровень владения навыками разработки несложных приложений в режиме windows-forms. Допускает множественные грубые ошибки	Демонстрирует удовлетворительный уровень владения навыками разработки несложных приложений в режиме windows-forms. Допускает достаточно серьезные ошибки.	Демонстрирует хороший уровень владения навыками разработки несложных приложений в режиме windows-forms. Допускает отдельные негрубые ошибки	Демонстрирует высокий уровень владения навыками разработки несложных приложений в режиме windows-forms. Не допускает ошибок.
---	---------------------	---	---	---	--

І курс, І семестр

Балльно-рейтинговая система (для очной формы)

І курс, І семестр (экзамен)

При балльно-рейтинговой системе все знания, умения и навыки, приобретаемые студентами в результате изучения дисциплины, оцениваются в баллах.

Оценка качества работы в рейтинговой системе является накопительной и используется для оценивания системной работы студентов в течение всего периода обучения.

І семестр

По итогам работы в семестре студент может получить максимально 70 баллов. Итоговой формой контроля в І семестре является экзамен. На экзамене студент может набрать максимально 30 баллов.

В течение І семестра студент может заработать баллы за следующие виды работ:

№	Вид работы	Сумма баллов
1	Работа на практических занятиях	17
2	Контрольная работа (ПР-2.1)	33
3	Контрольная работа (ПР-2.2)	33
4	Аудиторные занятия (посещение)	17
	Итого:	70

Если к моменту окончания семестра студент набирает от 51 до 70 баллов, то он получает допуск к экзамену.

Если студент к моменту окончания семестра набирает от 61 до 70 баллов, то он может получить автоматическую оценку «удовлетворительно». При желании повысить свою оценку, студент имеет право отказаться от автоматической оценки и сдать экзамен.

Если студент не набрал минимального числа баллов (51 балл), то он не получает допуск к экзамену.

Соответствие рейтинговых баллов и академических оценок

Общая сумма баллов за семестр	Итоговая оценка
86-100	Отлично
71-85	Хорошо
51-70	Допуск к экзамену
в том числе:	
61-70	Возможность получения автоматической оценки «удовлетворительно»
51-60	Только допуск к экзамену
0-50 *	Неудовлетворительно (студент не допущен к экзамену)

Текущий контроль успеваемости осуществляется в процессе выполнения практических и самостоятельных работ в соответствии с нижеприведенным графиком.

График выполнения самостоятельных работ студентами в І семестре:

Виды работ	Недели учебного процесса																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ПР-2.1	ВЗ						33										
ПР-2.2							ВЗ							33			

ВЗ – выдача задания

33 – защита задания

Инновационные формы проведения учебных занятий (для очной формы)

Семестр	Вид учебных занятий	Используемые инновационные формы проведения учебных занятий	Количество академ. часов
I	Лекционные занятия	Разбор конкретных задач и заданий (совместно с преподавателем) для выработки навыков командной работы по всем темам в соответствии с возможностями студентов (индивидуальные задания). Изучение применения классов как основы объектно-ориентированного программирования.	10
I	Практические занятия	Взаимопомощь студентов друг другу в процессах разработки алгоритма, написания кода, поиска ошибок в программах, тестирования. Такая взаимопомощь развивает качества, необходимые студентам в их будущей профессиональной работе, позволяет быстрее продвинуться к цели семинара не только сильным студентам, но и слабым. Выполнение контрольных, курсовой работы и домашних заданий стимулирует поиск и нахождение самостоятельных решений, нацелены на выработку профессиональных умений и навыков.	34
Всего:			44

Вопросы к экзамену по дисциплине «Программированию на языке высокого уровня», семестр I

1. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритмов.
2. Описание алгоритмов при помощи блок-схем.
3. Консольный проект.
4. Понятие переменной. Типы данных.
5. Область действия переменной.
6. Целые типы переменных. Ограничения по значению.
7. Числа с плавающей точкой. Ограничения по точности.
8. Арифметические операции. Порядок выполнения операций.
9. Булевские переменные. Возможные значения. Операции над булевыми переменными.
10. Логические операции.
11. Преобразование типов данных.
12. Ввод/вывод на консоль. Форматирование строк.
13. Операторы ветвления.
14. Оператор switch.
15. Операторы циклов (while, do, for).
16. Циклические алгоритмы. Операторы цикла с предусловием и постусловием.
17. Генератор случайных чисел.
18. Понятие одномерного массива (размер, индекс, доступ к элементам).
19. Понятие многомерного массива (размер, индексы, доступ к элементам).
20. Ступенчатые массивы. Оператор foreach.
21. Класс System.Array. Основные свойства и методы.
22. Класс: основные понятия и состав (поля, методы, конструкторы).

23. Создание объектов класса. Спецификаторы класса.
24. Методы класса. Сигнатура метода.
25. Конструктор класса и его назначение. Пример.
26. Интегрированная среда разработки приложений (IDE) и её основные возможности.
27. Важнейшие задачи при создании программ.
28. Краткая история методов программирования.
29. Основные принципы объектно-ориентированного программирования.
30. Инкапсуляция.
31. Наследование.
32. Полиморфизм.
33. Динамические структуры данных. Основные понятия.
34. Необходимость в динамических структурах данных.
35. Линейные списки (стеки, очереди, деки).
36. Основные понятия теории графов.
37. Определение и основные понятия дерева.

Примеры заданий по теме «Линейные, условные алгоритмы, операторы цикла» для сам. выполнения

Линейные алгоритмы

1. Даны два числа. Найти среднее арифметическое их квадратов и среднее арифметическое их модулей.
2. Даны стороны треугольника А, В, С. Вычислите площадь треугольника по формуле Герона:
3. Найти периметр и площадь прямоугольного треугольника, если даны длины его катетов а и b.
4. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
5. Найти длину окружности и площадь круга заданного радиуса R. В качестве значения π использовать 3.14.
6. Дано целое четырехзначное число. Найти сумму его цифр.
7. Дано целое четырехзначное число. Найти произведение его цифр.
8. Даны три числа А, В, С. Определить, можно ли построить треугольник со сторонами, равными этим числам. Программа должна ответить ДА или НЕТ.
9. Определить в какой четверти координатной плоскости лежит точка с координатами X, Y.
10. Дано натуральное число. Определить какой цифрой оно оканчивается: четной, нечетной или нулем.
11. Определить максимальное из трех чисел..
12. На чемпионате по футболу команде за выигрыш дается 2 очка, за проигрыш – 0, за ничью – 1. Известно количество очков, полученных командой за игру. Определить словесный результат игры (выигрыш, проигрыш или ничья).
13. Написать программу нахождения корней квадратного уравнения.
14. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести название соответствующего времени года ("зима", "весна" и т.д.).
15. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней в этом месяце

для невисокосного года.

Примеры заданий по теме «Массивы» для сам. выполнения

1. В одномерном числовом массиве посчитайте количество элементов, по модулю меньших заданного числа T .
2. В одномерном числовом массиве посчитайте количество и сумму четных элементов.
3. В одномерном числовом массиве посчитайте количество и сумму чисел, кратных 3.
4. Посчитать средний рост студента в группе.
5. Распечатать результаты контрольной работы по информатике, то есть посчитать среднюю отметку, количество пятерок, четверок, троек и т.д.
6. В одномерном числовом массиве посчитайте количество элементов, оканчивающихся на цифру 5.
7. Информация о температуре и осадках в течение N дней задана в виде двух массивов. Определить сколько осадков выпало в виде дождя и в виде снега, если считать, что дождь идет при температуре > 0 градусов по Цельсию.
8. В двумерном массиве $B(5,5)$ поменять местами второй и третий столбец. Результат вывести на экран.
9. Посчитать количество элементов в массиве, которые меньше среднего значения всех элементов массива $A(10,10)$.
10. Поиск наименьшего (наибольшего) элемента в двумерном массиве.
11. Подсчитать количество отрицательных элементов в двумерном массиве $C(6,8)$.
12. Пусть B – двумерный массив целых чисел размерности $N \times N$. Ввести его случайным образом. Поменять местами максимальный из чётных элементов и минимальный из элементов побочной диагонали и вывести на экран.
13. Пусть B – двумерный массив целых чисел размерности $N \times N$. Ввести его случайным образом. Поменять местами главную и побочной диагонали. Массив после каждой перестановки вывести на экран.
14. Пусть B – двумерный массив целых чисел размерности $N \times N$. Ввести его случайным образом. Поменять местами максимальный по модулю элемент и минимальный из элементов главной диагонали. Массив после каждой перестановки вывести на экран.
15. Пусть B – двумерный массив целых чисел размерности $N \times N$. Ввести его случайным образом. Поменять местами строки, содержащие максимальный и минимальный элементы. Массив после каждой перестановки вывести на экран.
16. Пусть B – двумерный массив целых чисел размерности $N \times N$. Ввести его случайным образом. Поменять местами строки с минимальной и максимальной суммой элементов. Массив после каждой перестановки вывести на экран.
17. Пусть B – двумерный массив целых чисел размерности $N \times M$. Ввести его случайным образом. Определить максимальное из чисел, встречающихся в заданной матрице более одного раза. Массив после каждого преобразования вывести на экран.
18. Пусть B – двумерный массив целых чисел размерности $N \times M$. Ввести его случайным образом. Определить максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы. Массив после каждого преобразования вывести на экран.

Примеры заданий по теме «Классы» для сам. выполнения

1. Определить класс «Дом», содержащий следующие поля:
 - количество этажей

- ширина дома
- длина дома
- стоимость постройки 1 кв.м дома

Определить методы класса:

- общая площадь дома
- количество затрачиваемых материалов
- стоимость постройки дома

Создать три экземпляра класса

2. Определить класс «Студент», содержащий следующие поля:

- номер студента
- оценка по экзамену 1
- оценка по экзамену 2
- оценка по курсовой работе

Определить методы класса:

- средняя оценка
- минимальная оценка
- максимальная оценка

Создать три экземпляра класса

I курс, II семестр

Балльно-рейтинговая система (для очной формы)

I курс, II семестр (экзамен, курсовая работа)

Во II семестре студент может получить максимально 70 баллов. Итоговой формой контроля во II семестре является экзамен. На экзамене студент может набрать максимально 30 баллов.

В течение II семестра студент может заработать баллы за следующие виды работ:

№	Вид работы	Сумма баллов
1	Работа на практических занятиях	17
2	Курсовая Работа (ПР-6)	36
3	Аудиторные занятия (посещение)	17
	Итого:	70

Если к моменту окончания семестра студент набирает от 51 до 70 баллов, то он получает допуск к экзамену.

Если студент к моменту окончания семестра набирает от 61 до 70 баллов, то он может получить автоматическую оценку «удовлетворительно». При желании повысить свою оценку, студент имеет право отказаться от автоматической оценки и сдать экзамен.

Если студент не набрал минимального числа баллов (51 балл), то он не получает допуск к экзамену.

Соответствие рейтинговых баллов и академических оценок

Общая сумма баллов за семестр	Итоговая оценка
86-100	Отлично
71-85	Хорошо
51-70	Допуск к экзамену
в том числе:	
61-70	Возможность получения автоматической оценки «удовлетворительно»
51-60	Только допуск к экзамену
0-50 *	Неудовлетворительно (студент не допущен к экзамену)

Текущий контроль успеваемости осуществляется в процессе выполнения практических и самостоятельных работ в соответствии с нижеприведенным графиком.

График выполнения самостоятельных работ студентами во II семестре:

Виды работ	Недели учебного процесса																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ПР-6		ВЗ												33			

ВЗ – выдача задания

33 – защита задания

Инновационные формы проведения учебных занятий (для очной формы)

Семестр	Вид учебных занятий	Используемые инновационные формы проведения учебных занятий	Количество академ. часов
---------	---------------------	---	--------------------------

II	Лекционные занятия	Разбор конкретных задач и заданий (совместно с преподавателем) для выработки навыков командной работы по всем темам в соответствии с возможностями студентов (индивидуальные задания). Изучение применения классов как основы объектно-ориентированного программирования.	10
II	Практические занятия	Взаимопомощь студентов друг другу в процессах разработки алгоритма, написания кода, поиска ошибок в программах, тестирования. Такая взаимопомощь развивает качества, необходимые студентам в их будущей профессиональной работе, позволяет быстрее продвинуться к цели семинара не только сильным студентам, но и слабым. Выполнение контрольных, курсовой работы и домашних заданий стимулирует поиск и нахождение самостоятельных решений, нацелены на выработку профессиональных умений и навыков.	34
Всего:			44

Вопросы к экзамену по дисциплине «Программированию на языке высокого уровня», II семестр

1. Класс: состав и использование (поля, методы, свойства, конструкторы).
2. Свойства классов, их назначение и возможности.
3. Метод как интерфейс класса. Сигнатура метода. Спецификаторы.
4. Передача параметров методу по значению. Пример.
5. Передача параметров методу по ссылке (по адресу). Пример.
6. Наследование классов. Пример.
7. Панель инструментов Toolbox. Понятие элементов управления C#.
8. Событие и обработчики события. Примеры.
9. Класс Form: свойства, методы, события.
10. Свойства и методы элементов управления для отображения и ввода текстовой информации (Label, TextBox). Примеры.
11. Свойства и методы контейнеров и переключателей (GroupBox, CheckBox, RadioButton). Примеры.
12. Свойства и методы списков и комбинированных списков (ListBox, ComboBox). Примеры.
13. Свойства и методы графических элементов управления (PictureBox). Примеры.
14. Основные элементы интерфейса: формы, меню, элементы управления, диалоговые окна.
15. Обработка исключительных ситуаций (операторы try и catch).
16. Модальные и немодальные окна приложения. Методы работы с окнами.
17. Виды и назначение окон приложения. SDI и MDI интерфейсы.
18. Диалоговые окна, их свойства.
19. Символы и строки. Основные понятия.
20. Класс Char. Символы, инициализация символов, методы. Примеры.
21. Массивы символов. Инициализация массивов символов. Пример.
22. Класс String. Операции над строками, свойства и методы. Пример.
23. Методы Split() и Join(). Пример.
24. Класс StringBuilder – построитель строк. Пример.
25. Отличие классов String и StringBuilder.

26. Иерархия классов ввода/вывода.
27. Потоки ввода/вывода. Работа с потоками данных.
28. Пространство имён System.IO и его основные классы.
29. Понятие статических и динамических методов.
30. Понятие файла, способы и параметры доступа к файлам.
31. Два способа описания пути к файлу (управляющие символы и дословные литералы).
Пример.
32. Символьные потоки (классы StreamReader, StreamWriter).
33. Байтовые потоки (класс FileStream).
34. Двоичные потоки (классы BinaryWriter и BinaryReader).
35. Основные понятия об элементе управления DataGridView.
36. Жизненный цикл ПО. Этапы жизненного цикла.
37. Рекурсия. Пример – функция факториал.

Примерная тематика курсовых работ

Темы курсовых работ можно разбить на несколько групп:

- разработка фрагмента информационной системы;
- разработка математического алгоритма с вводом данных и получением результата;
- разработка программ-тестов.
- студент может предложить свою тему по согласованию с преподавателем.

Разработка фрагмента информационно-справочной системы

1. Секретарь руководителя.
2. Библиографическая система.
3. Результаты сессии.
4. Обмен жилья.
5. Обработки итоговой таблицы чемпионата по футболу.
6. Обработки итоговой таблицы шахматного турнира.
7. Туристическое агенство.
8. Учебная часть ВУЗ-а.
9. Студенческий строительный отряд.
10. Назначение стипендии.
11. Администратор гостиницы.
12. Касса автовокзала или аэрофлота.
13. Библиотека.
14. Генеалогическое дерево моей семьи.
15. Биржа труда.
16. Рецепты приготовления блюд.
17. Отдел кадров.
18. Бюро знакомств.
19. Конфигурация компьютера.
20. ВУЗ-ы города Москвы.
21. Для поступающих в университет “Дубна”.
22. Город Протвино.
23. Каталог компьютерных игр.
24. Приемная комиссия ВУЗ-а.
25. Доска объявлений.
26. Крылатые фразы.
27. Художники.
28. Композиторы.
29. И т.д., практически любая деятельность может послужить основой для разработки информационно-справочной системы.

Разработка математического алгоритма

1. Программа обработки экспериментальных данных. Вычисление математического ожидания M и дисперсии D . Определение наибольшего и наименьшего элемента выборки данных и его порядкового номера в выборке. Печать гистограммы.
2. Программа работа с матрицами: транспонирование матриц, сложение, умножение матриц, нахождение определителя, обратной матрицы и т.д. Желательно представить каждую матрицу (исходные и результирующие) в виде таблицы, операции задавать с помощью меню.
3. Изучение фракталов. Используя известные математические алгоритмы [6] разработать программу, показывающую удивительный мир фракталов.
4. Представление линейных структур данных с использованием массивов (списки, очереди, стеки) [6].
5. Представление древовидных структур на базе элемента управления TreeView.

Разработка программ-тестов

Программы-тесты отличаются большим разнообразием, можно написать тесты по дисциплинам:

1. Программирование.
2. Русский язык.
3. Физика (или раздел физики).
4. Линейная алгебра.
5. Математический анализ.
6. Иностранный язык.
7. Психология.
8. Химия (или раздел химии).
9. Астрономия и т.д.
10. Тест, в котором предмет является параметром; вопросы тестов хранятся в файлах.

Фрагмент создания приложения «Тест»

Постановка задачи: Составить программу – тест «Решительны ли Вы?». На экране по очереди появляются 12 вопросов. На каждый из них нужно ввести ответ (*да* или *нет*). В зависимости от вопроса и ответа к общей сумме прибавляется определённое число очков. После окончания опроса по общей сумме очков делается заключение по поводу решительности опрашиваемого.

Для создания приложения «Тест» вам потребуется создать 6 форм:

Оформить все формы, заполнить файлы с вопросами.

Приводится фрагмент программы, в которой данные из одной формы передаются в другую форму.

```
private void button1_Click(object sender, System.EventArgs e)
{
    //Показываем результаты тестирования в зависимости от набранных баллов
    if (button1.Text == "РЕЗУЛЬТАТЫ")
    {
        if (ball <= 9)
        {
            t1 = "От 0 до 9"; //Результат
            //Рекомендации
            t2 = "Вы очень нерешительны. \nСтарайтесь изменить \nсвой
характер";
            t3 = "0_9.jpg"; //Название картинки
        }
        if (ball > 9 & ball < 19)
        {
            t1 = "От 10 до 18";
            t2 = "Вы принимаете \nпрещения осторожно, \nно не пасуете перед
\nсерьёзными проблемами";
        }
    }
}
```

```

        t3 = "10_18.jpg";
    }
    if (ball > 18 & ball < 29)
    {
        t1 = "От 19 до 28";
        t2 = "Вы достаточно решительны. \nПринятые решения
\nпотстаиваете до конца. \nСтарайтесь быть объективным. \nКонсультируйтесь по
вопросам, \nв которых недостаточно \nкомпетентны.";
        t3 = "19_28.jpg";
    }
    if (ball >= 29)
    {
        t1 = "От 29 и выше";
        t2 = "Вы принимаете решения единолично, \nне считаясь с мнением
других. \nЭто подавляет инициативу \nподчинённых, наносит \nущерб психологическому
\nклимату коллектива. \nНужно срочно менять стиль работы.";
        t3 = "от_29.jpg";
    }
    //Передача результатов в Form5
    Form5 f5 = new Form5();
    f5.TBtext = t1;
    f5.TBtext1 = t2;
    f5.Picture = t3;
    f5.Show();
    this.Close();
}

//Считывание из буфера bufq (в котором находятся вопросы)
if (numb < n-1)
{
    textBox1.Text = (numb + 1).ToString();
    textBox2.Text = bufq[numb];
    numb++;
}
else
{
    textBox1.Text = "";
    textBox2.Text = "";
    button1.Text = "РЕЗУЛЬТАТЫ";//Изменение надписи на кнопке, когда все
вопросы закончились
}

И т.д.

```

Примеры заданий по теме «Символы и строки» для сам. выполнения

Задание 1

1. Дано название планеты. Напечатать его на экран столбиком по три буквы.
2. С клавиатуры вводится строка, исключить из неё 2 первых слова, затем последнее слово поставить вперёд. Результат вывести на экран.

Задание 2

1. Составить программу, которая печатает заданное с клавиатуры слово, начиная с последней буквы (в обратном порядке). Вывести полученное слово на экран столбиком.
2. С клавиатуры вводится строка. Исключить из неё слова, состоящие из 4-х букв.

Задание 3

1. С клавиатуры вводится строка. Найти в ней слово максимальной длины и вывести его на экран.
2. Написать программу, определяющую является ли строка символов палиндромом. Программа должна ответить ДА или НЕТ.

Задание 4

1. С клавиатуры вводится строка. Удалить из неё слова, содержащие цифры. Результат вывести на экран.
2. Написать программу, определяющую является буква во введённом слове гласной или согласной.