

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»  
Факультет компьютерных наук и информационных технологий



УТВЕРЖДАЮ:

Декан факультета С. В. Миронов

2023 г.

Рабочая программа дисциплины  
«Информационные технологии и программирование»  
Направление подготовки бакалавриата  
03.03.01 «Прикладные математика и физика»  
Профиль подготовки бакалавриата  
«Нелинейные процессы в микроволновых системах»  
Квалификация (степень) выпускника  
*Бакалавр*

Форма обучения  
*очная*

Саратов, 2023

Статус	ФИО	Подпись	Дата
Преподаватели-разработчики	Сысоев Илья Вячеславович	<i>И.С.</i>	31.08.23
	Корнилов Максим Вячеславович	<i>М.К.</i>	31.08.23
Председатель НМК	Кондратова Юлия Николаевна	<i>Ю.К.</i>	31.08.23
Заведующий кафедрой	Тынанко Игорь Евстафьевич	<i>И.Т.</i>	31.08.23
Специалист Учебного управления			

# 1 Цели освоения дисциплины

Целью освоения дисциплины «Информационные технологии и программирование» является формирование у студентов общепрофессиональных и специальных знаний в области прикладного программирования, в том числе в области написания, тестирования, расширения и изменения ранее написанных прикладных программ.

Задачами освоения дисциплины являются:

- формирование *знаний* об информационных процессах, методах поиска, сбора, хранения, обработки, предоставления, распространения информации и способах осуществления таких процессов и методов (информационных технологиях);
- формирование *умений* самостоятельно писать, модифицировать и тестировать прикладные компьютерные программы;
- формирование *владений* языками программирования, средами разработки программ, алгоритмическими приёмами и подходами к тестирования и выявлению ошибок в программах.

## 2 Место дисциплины в структуре ООП бакалавриата

Дисциплина «Информационные технологии и программирование» относится к обязательной части блока 1 «Дисциплины (модули)» и изучается студентами в течение 1-го и 2-го учебных семестров (1-ый курс). Дисциплина опирается на базовые знания математики и информатики, доступные учащимся в результате освоения школьной программы. Дисциплина готовит студентов к освоению дальнейших общих и специальных дисциплин.

## 3 Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора (индикаторов) достижения компетенции	Результаты обучения
ОПК-6 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ИД-1ОПК-6. Использует современные языки программирования для разработки алгоритмов и компьютерных программ, пригодных для практического применения	Знать базовые алгоритмы обработки данных. Уметь выбирать соответствующие задаче методы обработки экспериментальных данных. Владеть навыками считывания, записи, графического отображения экспериментальных данных.
	ИД-2ОПК-6. Применяет современные программные среды разработки информационных систем и технологий, методы отладки и тестирования, читает коды программных продуктов, написанные на освоенных языках программирования, и вносит требуемые изменения	Знать сферы применения ЭВМ в исследованиях. Уметь вычленять типы работ, поддающиеся алгоритмизации и программированию. Владеть методами автоматизации исследований.
	ИД-3ОПК-6. Готов самостоятельно осваивать новые для себя языки программирования, среды разработки информационных систем и технологий	Знать современные среды разработки информационных систем и технологий. Уметь анализировать, и модифицировать код программ. Владеть навыками отладки и тестирования программных продуктов.

## 4 Структура и содержание дисциплины

Общая трудоёмкость дисциплины составляет 6 зачётных единиц (216 часов).

#### 4.1 Распределение часов по темам

№ п/п	Наименование раздела, подраздела, темы лекции	Семестр	Неделя семестра	Бюджет учебного времени					Формы текущего контроля успеваемости и промежуточной аттестации (по семестрам)
				Всего	В том числе				
					Лек- ции	Лабо- ра- тор- ные	Лаб. практ. под- готов- ка	СРС	
1	2	3	4	5	6	7	8	9	10
Очная полная программа									
1	История развития информатики и вычислительной техники. Классификация языков программирования. Типы данных	1	1-2	12	4	4		4	индивидуальные отчёты по заданиям
2	Условия и логические выражения, составные типы данных. Математический модуль, генерация случайных чисел	1	3-4	12	4	4		4	индивидуальные отчёты по заданиям
3	Цикл с условием, суммирование с накоплением, расчёт среднего, флаги, счётчики событий	1	5-6	12	4	4		4	индивидуальные отчёты по заданиям
4	Цикл обхода последовательности, цикл со счётчиком, досрочное прерывание выполнения. Вложенные циклы	1	7-8	12	4	4		4	индивидуальные отчёты по заданиям
5	Массивы: одномерные и двумерные, создание и операции с массивами как переменными и как с наборами переменных, чтение и запись массивов в файл	1	9-10	12	4	4	4	4	индивидуальные отчёты по заданиям
6	Построение графиков на компьютере: визуальный и командный интерфейсы к построению графиков, программы для построения, библиотеки и модули, визуализация и сохранение, векторные и растровые форматы графических файлов	1	11-12	12	4	4	4	4	индивидуальные отчёты по заданиям

7	Файлы: текстовый доступ на чтение и запись, двоичный доступ; операционная система, дерево каталогов, и работа с ними	1	13-14	12	4	4	4	4	индивидуальные отчёты по заданиям
8	Библиотеки алгоритмов: линейная алгебра, Фурье-анализ, генерация случайных чисел	1	15-17	24	6	6	6	12	индивидуальные отчёты по заданиям контрольная работа
Промежуточная аттестация									экзамен
Итого за 1 учебный семестр				144	34	34	18	40	36
9	Подпрограммы: функции и процедуры, возврат значения, передача по указателю и по значению, именованные аргументы и значения по умолчанию, функции с переменным числом аргументов. Принципы функционального программирования: рекурсия, функции высших порядков, безымянные функции, конвейер. Модули	2	1-4	18	8	8	4	2	индивидуальные отчёты по заданиям
10	Графический интерфейс пользователя: базовые виджеты, модальные диалоги, составные виджеты, работа с графическими объектами на форме	2	5-8	18	8	8	4	2	индивидуальные отчёты по заданиям
11	Объектно-ориентированное программирование, написание программ с наследованием	2	9-10	10	4	4	2	2	индивидуальные отчёты по заданиям
12	Базы данных: общая организация, виды баз данных, язык SQL и интеграция его с языками программирования, типы данных в базах	2	11-16	26	12	12	6	2	индивидуальные отчёты по заданиям контрольная работа
Промежуточная аттестация									зачёт
Итого за 2 учебный семестр				72	32	32	16	8	
Итого по курсу				216	66	66	34	48	36

## 4.2 Содержание дисциплины

1. История развития информатики и вычислительной техники: архитектуры ЭВМ, элементная база ЭВМ, понятие информатики и кибернетики. История и классификация языков программирования.

Типы данных: числовые, логические, строковые. Арифметические операции с числовыми типами, операции со строками. Операции сравнения и логические операции.

2. Условия и логические выражения. Таблицы истинности. Условный оператор, оператор с несколькими условиями. Сложные условия. Составные типы данных: списки, кортежи, словари (ассоциированные массивы). Стандартная библиотека, математический модуль, модуль генерации случайных чисел.
3. Понятие циклов. Цикл с условием. Суммирование с накоплением, расчёт среднего, произведение с накоплением, расчёт среднего геометрического. Переменные флаги, переменные счётчики событий.
4. Цикл обхода последовательности, цикл со счётчиком, досрочное прерывание выполнения цикла. Вложенные циклы, операции с двумя счётчиками.
5. Массивы: одномерные (векторы) и двумерные. Различие между массивами и списками: последовательный и прямой доступ. Форма и размер массива. Создание и операции с массивами как переменными — векторные операции, сечения массивов, операции со столбцами и строками матриц как с объектами, перемножение матриц и векторов. Работа с массивами как с контейнерами: доступ к отдельным элементам, обход массивов поэлементно по значению и по индексу. Запись в файл и чтение массивов из файла.
6. Построение графиков на компьютере: визуальный и командный интерфейсы к построению графиков. Отдельные программы для работы с графической информацией, в том числе для построения графиков. Библиотеки и модули для построения графиков: универсальные библиотеки с интерфейсом к различным языкам и специализированные библиотеки. Визуализация зависимостей. Типы графиков: линейные, точечные, столбцовые, круговые. Трёхмерные графики и линии уровней. Форматы векторной и растровой графики, особенности и методы работы. Сохранение графиков в файлы и чтение графических файлов из программы.
7. Работа с содержимым файлов. Текстовый доступ на чтение и запись, чтение всего файла целиком и построчно, разбор (парсинг) текстовых выражений. Двоичный доступ к файлам на чтение и запись, преимущества и недостатки двоичного доступа. Специализированные форматы файлов. Понятие об операционной системе, понятие о дереве каталогов, абсолютный и относительный пути. Программное создание и удаление файлов и каталогов, чтение содержимого каталогов.
8. Понятие о численных методах и библиотеках алгоритмов. Обзор наиболее популярных библиотек. Стандартные библиотеки линейной алгебры, Фурье-анализа, генерации случайных чисел.
9. Понятие о подпрограммах: функции и процедуры. Возврат значения из функции, возврат нескольких значений; передача данных в функцию по указателю (ссылке) и по значению, передача данных из подпрограммы по указателю (ссылке). Именованные аргументы и значения по умолчанию, функции с переменным числом аргументов. Принципы функционального программирования: рекурсия, функции высших порядков, безымянные функции, конвейер функций. Модули, стандартные, внешние и собственные модули, различие модуля и главной программы.
10. Графический интерфейс пользователя. Базовые виджеты: кнопка, поле ввода, метка, листбокс, радиокнопки, флажки. Глобальное меню и контекстное меню. Модальные диалоги: диалоги открытия файлов на чтение и запись, диалог выбора цвета. Составные виджеты: комбобокс, линейка прокрутки. Работа с графическими объектами на форме: холст и объекты на холсте.
11. Объектно-ориентированное программирование. Понятие о наследовании, инкапсуляции и полиморфизме. Написание программ с наследованием.
12. Базы данных: общая организация, виды баз данных. Реляционные базы данных, таблицы как основной элемент реляционных баз данных: создание, чтение, модификация таблиц. Типы данных в базах. Связывание таблиц. Язык SQL и интеграция его с языками программирования.

#### 4.3 Перечень работ компьютерного практикума, в том числе проектных заданий (примерный)

1. Арифметические выражения и стандартные функции.

Цель работы: научиться использовать арифметические операторы и стандартные функции.

Задание: используя арифметические операторы, а также стандартные функции округления, отсечения целой части и модуля, составьте арифметические выражения и напишите программу для вычисления и вывода ряда величин (необходимая информация запрашивается у пользователя).

2. Операции с несколькими числами.  
Цель работы: научиться писать программы и обмениваться информацией с компьютером.  
Задание: реализовать вычисление по заданной формуле функции нескольких переменных.
3. Условия и сравнения.  
Цель работы: научиться использовать условный оператор и операторы сравнения.  
Задание: реализовать вычисления с ветвлением в зависимости от значения введённого аргумента, конкретное задание выдаётся преподавателем.
4. Сложные условия.  
Цель работы: научиться составлять сложные условия.  
Задание: составить сложные условия с использованием операторов сравнения и логических операторов.
5. Повторяющиеся операции.  
Цель работы: научиться реализовывать повторяющиеся операции с помощью цикла.  
Задание: реализуйте вывод многократно повторяющихся символов на экран по заданному преподавателем правилу.
6. Накопление произведения и суммы.  
Цель работы: овладеть алгоритмическими приёмами накопления произведения и суммы.  
Задание: написать программу для суммирования или мультиплицирования с накоплением заданной последовательности, конкретный вид последовательности определяется преподавателем.
7. Рекуррентные соотношения.  
Цель работы: освоить рекуррентные соотношения.  
Задание: написать программу, реализующую вычисления по заданному преподавателем рекуррентному соотношению.
8. Табулирование функций.  
Цель работы: научиться табулировать функции.  
Задание: написать программу, которая выводит значения аргумента и функции при:  
  - (а) шаге по аргументу, его начальном значении и числе значений;
  - (b) шаге по аргументу, его начальном и конечном значении;
  - (с) начальном и конечном значении аргумента и числе значений.
9. Вычисление номера шага.  
Цель работы: научиться вычислять номер шага, на котором происходит некоторое событие.  
Задание: написать программу, которая при помощи конструкции цикла вычисляет номер шага, при котором впервые выполняется заданное преподавателем условие.
10. Вычисления с заданной точностью.  
Цель работы: научиться вычислять значения по рекуррентным формулам с заданной точностью.  
Задание: написать программу, реализующую вычисление по заданной рекуррентной формуле с оставкой не по числу шагов, а по достижении заданной точности.
11. Чтение из файла.  
Цель работы: научиться читать из файла числовые данные, записанные в столбец или несколько столбцов.  
Задание: написать программу, читающую из данного преподавателем текстового файла данные, записанные в один или несколько (заранее известное количество) столбцов.
12. Запись в текстовый файл.  
Цель работы: научиться записывать числовые данные в текстовый файл в один или несколько столбцов.  
Задание: написать программу, реализующую запись значений, генерируемых в цикле по оговоренному правилу, в файл.
13. Вычисление минимума и максимума, суммы, среднего значения по данным из файла.  
Цель работы: научиться рассчитывать минимум, максимум сумму или среднее значения данных, записанных в файл.  
Задание: написать программу, реализующую чтение числовых данных из файла, записанных в один столбец, и расчёт по ним суммы, среднего, минимального или максимального значения.

14. Чтение массива из файла.  
Цель работы: научиться читать из файла числовые данные в массив.  
Задание: написать программу, читающую из данного преподавателем текстового файла данные в массив.
15. Запись массива в текстовый файл.  
Цель работы: научиться записывать числовые данные из массива в текстовый файл.  
Задание: написать программу, реализующую запись значений одного или нескольких массивов в файл.
16. Вычисление минимума и максимума, суммы, среднего значения массива числовых данных.  
Цель работы: научиться рассчитывать минимум, максимум сумму или среднее значения для массива.  
Задание: написать программу, реализующую расчёт суммы, среднего, минимального или максимального значения для массива.
17. Записи.  
Цель работы: освоить работу с записями (структурами).  
Задание: написать программу, производящую вычисления с различными полями двух или нескольких записей, задание детализируется преподавателем.
18. Строки.  
Цель работы: освоить базовые операции со строками.  
Задание: написать программу, реализующую некоторые базовые операции со строками, в том числе со строками, являющимися полями записей.
19. Указатели.  
Цель работы: научиться работать с указателями.  
Задание: решить 1–6, используя вместо части переменных указатели.
20. Связанные списки.  
Цель работы: научиться создавать и обходить связанные списки.  
Задание: решить задания 14–16, используя вместо массивов связанные списки.
21. Вложенные циклы.  
Цель работы: научиться использовать вложенные циклы для сложных повторяющихся операций.  
Задание: написать программу, совмещающую вычисления с заданной точностью и табулирование функции, для чего поместите цикл с условием внутрь цикла со счётчиком.
22. Чтение и запись двумерных массивов.  
Цель работы: научиться читать и писать двумерные массивы из файла/в файл.  
Задание: написать программу, считывающую данные из файла, записанного в несколько столбцов, в двумерный массив и выводящую данные из двумерного массива в файл.
23. Процедуры.  
Цель работы: научиться писать простые процедуры.  
Задание: написать и протестировать подпрограмму, выполняющую заданные несложные действия из заданий 1–6, используя только локальные переменные и передачу аргумента по значению.
24. Процедуры со сложными аргументами.  
Цель работы: научиться передавать сложные аргументы: массивы и записи в процедуры.  
Задание: написать и протестировать подпрограмму, выполняющую одно из заданий 14–17 и использованием только локальных переменных и передающую значения в основную программу по ссылке.
25. Функции.  
Цель работы: научиться писать функции со сложными аргументами.  
Задание: написать и протестировать функцию расчёта одного из следующих свойств массива:
  - (a) минимума,
  - (b) максимума,
  - (c) суммы элементов,
  - (d) среднего,
  - (e) медианного значения,

(f) евклидовой длины.

26. Модули.

Цель работы: научиться писать и подключать модули.

Задание: реализуйте подпрограмму из заданий 23–26 в отдельном модуле, обеспечьте корректный импорт этого модуля.

27. Двоичные файлы.

Цель работы: научиться читать и писать двоичные файлы.

Задание: переработайте задания 14–15 с использованием двоичного типизированного файла вместо текстового.

## 5 Образовательные технологии, применяемые при освоении дисциплины

При реализации различных видов учебной работы по данной дисциплине с целью создания условий для самоактуализации и самореализации обучающихся, предоставления возможностей для конструирования собственного знания, используются следующие современные образовательные технологии:

- информационно-коммуникационные технологии;
- конкурсное и соревновательное обучение;
- работа в малых группах;
- творческие задания.

Аудиторные занятия проводятся в форме лекций и компьютерного практикума. Самостоятельная работа предполагает прежде всего самостоятельное выполнение студентом заданий компьютерного практикума, а также разбор лекционного материала и материала лабораторных практикумов.

Курс построен таким образом, что лекционные занятия и практикум тесно связаны друг с другом, лекционный материал подготавливает студента к выполнению соответствующего практического задания. Поскольку лекции носят практическую направленность, на них, часто в интерактивной форме, объясняются в том числе различные технические аспекты практических заданий.

Лекционные занятия проводятся частично при помощи мультимедийного проектора. Перед занятиями или непосредственно после них студентам может выдаваться конспективный материал.

Практикум является главной формой занятий, т. к. развивает у студентов непосредственные навыки и служит закреплению полученного на лекциях и лабораторных практикумах материала. Задания практикума выдаются с небольшим опережением, чтобы дать возможность студенту самому планировать свою работу. В аудитории студенты должны отлаживать критически важные фрагменты программ, обсуждать с преподавателем структуру и ключевые особенности реализации алгоритма, сдавать сделанные задания. Несмотря на большой объём аудиторных занятий, выполнение всех заданий в аудитории невозможно, поэтому наиболее простая и трудоёмкая часть заданий, не требующая постоянного контроля и содействия преподавателя, должна выполняться студентами в рамках самостоятельной работы.

В рамках лабораторных занятий осуществляется практическая подготовка, предусматривающая участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью. Во время практической подготовки студенты решают практические задачи, связанные с применением информационных технологий и средств программирования для разработки приложений, относящихся к профессиональной деятельности.

Задания практикума предполагают наличие у студентов базовых навыков программирования и логического мышления, поскольку все задания выполняются с помощью численных методов на компьютере. Тем не менее, практикум не ставит целью расширение знаний студента в области программирования. В качестве инструментария может применяться любой знакомый студенту язык программирования, может использоваться любая операционная система. Задания практикума не предполагают наличие в языке программирования мощных встроенных библиотек численных алгоритмов и построены таким образом, чтобы следующие задания опирались на предыдущие. В качестве средства реализации может быть выбран любой современный распространённый язык, например, Pascal, Fortran, Python, C или иной. Выбор средств разработки: языка и среды является свободным для студента при соблюдении требования о технической и юридической возможности и правомочности использования выбранных средств.



## **Условия обучения инвалидов и лиц с ограниченными возможностями здоровья**

Для обучения инвалидов и лиц с ограниченными возможностями здоровья создаются следующие дополнительные условия:

- предоставление инвалидам по зрению или слабовидящим возможностей использовать крупноформатные наглядные материалы;
- организация коллективных занятий в студенческих группах с целью оказания помощи в получении информации инвалидам и лицам с ограниченными возможностями по здоровью;
- проведение индивидуальных коррекционных консультаций для инвалидов и лиц с ограниченными возможностями здоровья;
- использование индивидуальных графиков обучения;
- использование дистанционных образовательных технологий.

## **6 Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**

### **6.1 Организация самостоятельной работы**

Конспекты лекций и задания для практикума выдаются студентам свободно после каждой лекции. При необходимости, они также могут получить программное обеспечение, лицензия которого позволяет его свободное распространение, в частности интерпретатор языка Python и модули к нему, среду разработки Lazarus для программирования на языке FreePascal, среду разработки Code::Blocks и набор компиляторов gcc и другие средства, которые могут быть доступны в компьютерном классе или иметься на руках у преподавателя.

Основным критерием освоения студентами курса выступает успешное выполнение ими работ практикума. Все задания практикума строятся таким образом, чтобы как на завершающем, так и, по возможности, на промежуточных этапах можно было проконтролировать правильность выполнения. Для контроля правильности выполнения используются специальные тестовые примеры, для которых результат известен из теоретических соображений. Данные для этих примеров могут даваться преподавателем либо, в ряде случаев, генерироваться самими студентами. В ряде случаев действенным методом контроля правильности полученных результатов является использование уже отлаженных ранее программ, например, рассчитав сумму чисел, записанных в текстовый файл, студент может использовать этот результат для проверки корректности работы с двоичным.

### **6.2 Методические рекомендации студентам**

- Систематически посещать лекции, а в случае пропуска занятий по уважительной причине в кратчайшие сроки ознакомиться с пропущенным материалом. Это позволит без лишнего напряжения и полно воспринять представленный материал, а также при выполнении практических заданий не потерять связь с лекционным курсом.
- Иметь при себе тетрадь с лекциями по курсу и на каждом занятии практикума.
- Стараться выполнить наиболее сложные задания в присутствии преподавателя, оставив для самостоятельной разработки более трудоёмкие, но алгоритмически несложные части работы.
- При возникновении сложно преодолимых трудностей не стесняться консультироваться у преподавателя и обращаться к рекомендованной литературе.

### **6.3 Методические рекомендации преподавателю**

- Ознакомить студентов с программой курса, списком экзаменационных и контрольных вопросов, списком рекомендованной литературы и планом работы на семестр на первой лекции.
- Раздать студентам материалы для самостоятельной подготовки, в том числе и задания для вычислительной практики.

- Консультировать студентов по мере прослушивания курса и выполнения ими заданий самостоятельной подготовки.
- Во время занятий стараться охватить наиболее сложные и неочевидные моменты.

## 6.4 Рекомендации по использованию информационных технологий

Кроме традиционного использования учебников, монографий и периодических научных изданий для более глубокого усвоения материала дисциплины целесообразно пользоваться ресурсами интернета и электронными учебными материалами, распространяемыми на компакт-дисках и пр.

Также приветствуется использование сетевых баз данных, библиотек методов и эталонных сигналов и прочих ресурсов, содержащих полезную информацию, способную расширить и углубить знания студентов в области обработки экспериментальных рядов.

## 6.5 Вопросы для проведения промежуточной аттестации по итогам освоения дисциплины

### 6.5.1 На 1-ый учебный семестр

- Основные понятия информационных процессов
  - Информация, сообщение, код, кодирование.
  - Информатика, информационная технология.
  - Поиск, передача, хранение информации.
  - Общая схема системы связи.
  - Свойства информации.
- Технические средства реализации информационных процессов
  - Архитектура и структура ЭВМ.
  - Классификация ЭВМ.
  - Структура и принципы функционирования микро-ЭВМ.
  - Память ЭВМ.
  - Устройства ввода-вывода.
  - Телеобработка.
- Программные средства реализации информационных процессов
  - Программа, программное обеспечение ЭВМ, алгоритм, программирование.
  - Системное программное обеспечение.
  - Операционная система ЭВМ, операционная оболочка.
  - Файл, файловая система, каталог.
  - Сервисное программное обеспечение.
  - Прикладное программное обеспечение.
- Линейные программы
  - Числовые типы данных. Оператор присваивания. Арифметические операторы. Скобки в арифметических выражениях.
  - Логический тип данных. Операторы сравнения. Логические операторы.
  - Математический модуль `math`. Базовые математические функции с 1 аргументом.
  - Понятие о составных типах данных. Списки. Методы списков. Операции со списками.
  - Понятие о составных типах данных. Кортежи. Операции с кортежами.
  - Строковый тип данных. Операции над строками и их методы. Стандартные функции `chr` и `ord`.
  - Универсальные стандартные функции `len`, `sum`, `type`, `print`.
- Условия и циклы

- (a) Понятие о составном операторе. Условный оператор. Конструкции `else` и `elif`.
- (b) Цикл с условием `while`.
- (c) Оператор цикла обхода последовательности `for`. Функция `range`.
- (d) Вложенные условия и циклы. Оператор досрочного выхода из цикла `break`.

6. Массивы. Модуль `numpy`.

- (a) Создание и индексация массивов.
- (b) Арифметические операции и функции с массивами.
- (c) Двумерные массивы, форма массивов.

7. Графики. Модуль `matplotlib`.

- (a) Простые графики.
- (b) Заголовок, подписи, сетка, легенда.
- (c) Несколько графиков на одном полотне.
- (d) Гистограммы, диаграммы-столбцы.
- (e) Круговые и контурные диаграммы.
- (f) Трёхмерные графики.
- (g) Учёт ошибок.

8. Файлы.

- (a) Чтение из текстовых файлов целиком и циклом построчно.
- (b) Запись из текстовые файлы отдельных чисел и массивов/списков.
- (c) Чтение байтов из файла, разбор файла: преобразование байтов числа разной разрядности.
- (d) Запись в файл чисел, в том числе массивов в бинарной форме.

9. Библиотеки, встроенные в `numpy`.

- (a) Элементы линейной алгебры.
- (b) Быстрое преобразование Фурье.
- (c) Генерация случайных чисел.

## 6.5.2 На 2-ой учебный семестр

1. Функции

- (a) Синтаксис объявления функции. Создание функции с известным числом аргументов.
- (b) Правила возврата и присвоения результата выполнения функции, оператор `return`. Функции без `return` и с несколькими `return`. Возврат нескольких значений из функции.
- (c) Значения параметров функции по умолчанию. Обращение к параметрам функции по имени при её вызове.
- (d) Функции с переменным числом аргументов.
- (e) Функция в функции. Область видимости параметров, декларации `nonlocal` и `global`. Рекурсивные функции.

2. Графический интерфейс пользователя.

- (a) Форма и кнопки, расположение виджетов на форме.
- (b) Текстовые поля ввода, метки, списки.
- (c) Радиокнопки, флажки.
- (d) Стандартные диалоги.
- (e) Полотно, рисование, редактирование.

3. Объектно-ориентированное программирование.

- (a) Использование методов стандартных классов.
  - (b) Создание собственных классов с нуля. Поля и методы, конструктор.
  - (c) Наследование, обращение к конструктору базового класса. Объекты как экземпляры классов.
4. Базы данных.
- (a) Понятие базы данных. Классификация баз данных.
  - (b) Понятие об SQL как о языке запросов. Принципы SQL. Адресация SQL запросов средствами стандартной библиотеки языка программирования.
  - (c) Создание и ликвидация базы данных, таблиц средствами языка программирования и SQL.
  - (d) Пополнение таблиц, модификация таблиц, связывание таблиц.
  - (e) Запросы к базе данных. Сложные запросы.

## 6.6 Контрольные вопросы

1. Информатика, информационная технология.
2. Общая схема системы связи.
3. Архитектура и структура ЭВМ.
4. Классификация ЭВМ.
5. Структура микро-ЭВМ.
6. Системное программное обеспечение.
7. Операционная система ЭВМ.
8. Прикладное программное обеспечение.
9. Понятие алгоритма.
10. Переменные и их типы, правила составления идентификаторов.
11. Оператор присваивания.
12. Арифметические операторы.
13. Стандартные функции.
14. Соответствие типов, допустимые и недопустимые присваивания.
15. Структура программы.
16. Ввод/вывод в текстовом режиме.
17. Условный оператор. Операторы сравнения.
18. Логический тип данных и логические выражения. Логические операторы. Вычисление значения логических выражений.
19. Цикл со счётчиком, основные правила.
20. Операторы досрочного завершения цикла и итерации, цикл с уменьшающимся счётчиком.
21. Приём накопления суммы и произведения.
22. Вычисления с помощью рекуррентных соотношений.
23. Запись в текстовый файл чисел.
24. Чтение чисел из текстового файла.
25. Многомерные рекуррентные отношения.
26. Создание массива, выделение памяти.
27. Стандартные функции с массивами.

28. Чтение и запись из файла в массив.
29. Использование записей. Производные типы.
30. Символы, кодировки, вычисление символа по коду и кода по символу.
31. Строки, стандартные функции со строками. Явное преобразование типов.
32. Стандартные модули, библиотечные функции для работы с числовыми типами.
33. Вложенные циклы.
34. Двумерные массивы.
35. Подпрограммы: процедуры и функции.
36. Передача данных в подпрограмму по значению и по ссылке.
37. Возврат составных данных из функции.
38. Локальные переменные, области видимости.
39. Перенос данных в модули.
40. Типизированные указатели на данные скалярных типов.
41. Типизированные указатели на данные составных типов.
42. Особенности работы с двоичными файлами.

## 7 Данные для учета успеваемости студентов в БАРС

Таблица 1.1 Таблица максимальных баллов по видам учебной деятельности

	1	2	3	4	5	6	7	8
Семестр	Лекции	Лабораторные занятия	Практические занятия	СРС	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
1	0	40	0	20	0	10	30	100
2	0	40	0	30	0	10	20	100

### 7.1 Программа оценивания учебной деятельности студента

#### Семестр 1

*Лекции* — посещаемость не оценивается.

*Лабораторные занятия* — выполнение работ компьютерного практикума, предусмотренных рабочей программой: от 0 до 40 баллов.

*Практические занятия* — учебным планом по данной дисциплине не предусмотрены.

*Самостоятельная работа* в семестре (СРС) — выполнение заданий компьютерного практикума самостоятельно: от 0 до 20 баллов.

*Автоматизированное тестирование* — не предусмотрено рабочей программой.

*Другие виды учебной деятельности* — выполнение индивидуальных заданий по инициативе студента: от 0 до 10 баллов.

*Промежуточная аттестация (экзамен)* — в зависимости от ответа студента на экзамене: от 0 до 30 баллов.

Подготовка студента к прохождению промежуточной аттестации осуществляется в ходе лекционных и лабораторных занятий, а также во внеаудиторные часы в рамках самостоятельной работы студента. Во время самостоятельной подготовки студент пользуется конспектами лекций, основной и дополнительной литературой по дисциплине. Промежуточная аттестация проводится в виде письменного экзамена. Во время проведения экзамена студент должен дать развернутый ответ на вопросы экзаменационного билета. Преподаватель вправе задавать дополнительные вопросы по всем разделам изучаемой дисциплины. Во время ответа студент должен продемонстрировать знания по всему материалу изучаемой дисциплины. Студент должен уметь разделять факты и их интерпретацию, владеть методами аргументирования своих

утверждений, а также методами построения и анализа математических моделей технологических процессов и производственных систем. Полнота ответа определяется показателями оценивания результатов обучения (раздел 2 «Фонда оценочных средств»).

При проведении промежуточной аттестации в виде экзамена:

- ответ на «отлично» оценивается от 24 до 30 баллов;
- ответ на «хорошо» оценивается от 16 до 23 баллов;
- ответ на «удовлетворительно» оценивается от 8 до 15 баллов;
- ответ на «неудовлетворительно» оценивается от 0 до 7 баллов.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 1 семестр по дисциплине «Информационные технологии и программирование» составляет 100 баллов.

Таблица 2.2 Таблица пересчёта полученной студентом суммы баллов по дисциплине «Информационные технологии и программирование» в оценку (экзамен)

количество баллов	оценка
85–100	«отлично»
61–84	«хорошо»
40–60	«удовлетворительно»
0–39	«неудовлетворительно»

Текущие индивидуально набранные студентами баллы доводятся до их сведения 2 раза за семестр: в конце 6 и 12 недель обучения. Оценка студентам, успешно прошедшим обучение по дисциплине «Информационные технологии и программирование», может быть проставлена без сдачи экзамена на основании рейтинговой оценки по решению преподавателя.

## Семестр 2

*Лекции* — посещаемость не оценивается.

*Лабораторные занятия* — выполнение работ компьютерного практикума, предусмотренных рабочей программой: от 0 до 40 баллов.

*Практические занятия* — учебным планом по данной дисциплине не предусмотрены.

*Самостоятельная работа* в семестре (СРС) — выполнение заданий компьютерного практикума самостоятельно: от 0 до 30 баллов.

*Автоматизированное тестирование* — не предусмотрено рабочей программой.

*Другие виды учебной деятельности* — выполнение индивидуальных заданий по инициативе студента: от 0 до 10 баллов.

*Промежуточная аттестация (зачёт)* — в зависимости от ответа студента на зачёте: от 0 до 20 баллов.

При проведении промежуточной аттестации в виде зачёта:

- ответ на «зачтено» оценивается от 12 до 20 баллов;
- ответ на «не зачтено» оценивается от 0 до 11 баллов.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 2 семестр по дисциплине «Информационные технологии и программирование» составляет 100 баллов.

Таблица 2.1 Таблица пересчёта полученной студентом суммы баллов по дисциплине «Информационные технологии и программирование» в оценку (зачёт)

количество баллов	оценка
40–100	«зачтено»
0–39	«не зачтено»

Текущие индивидуально набранные студентами баллы доводятся до их сведения 2 раза за семестр: в конце 6 и 12 недель обучения. Оценка студентам, успешно прошедшим обучение по дисциплине «Информационные технологии и программирование», может быть проставлена без сдачи зачёта на основании рейтинговой оценки по решению преподавателя.

## 8 Учебно-методическое и информационное обеспечение дисциплины

### а) Литература

1. Сузи Р.А. Язык программирования Python [Электронный ресурс].— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2020.— 350 с.— Режим доступа: электронный <https://www.iprbookshop.ru/97589.html> — ЭБС «IPRbooks».
2. Маккинли У. Python и анализ данных [Электронный ресурс]. — Саратов : Профобразование, 2019. — 482 с. — Режим доступа: электронный — URL: <https://www.iprbookshop.ru/88752.html> — ЭБС «IPR SMART».
3. Сысоева М.В. Программирование для «нормальных» с нуля на языке Python: учебник: в 2-х частях / М.В. Сысоева, И.В. Сысоев - Москва: Базальт СПО; МАКС Пресс, 2018 - . - Текст: непосредственный. Ч.1. - Москва: Базальт СПО; МАКС Пресс, 2018. - 176 с.: ил. - (Библиотека ALT). - ISBN 978-5-317-05784-8: ~Б. ц.
4. Сысоева М.В. Программирование для «нормальных» с нуля на языке Python: учебник: в 2-х частях / М.В. Сысоева, И.В. Сысоев - Москва: Базальт СПО; МАКС Пресс, 2018 - . - Текст: непосредственный. Ч.2. - Москва: Базальт СПО; МАКС Пресс, 2023 - 184 с.: ил. - (Библиотека ALT). - ISBN 978-5-317-06945-2. - ISBN 978-5-317-06947-6 (Часть 2): ~Б. ц.

### б) Программное обеспечение и Интернет-ресурсы

1. Операционная система семейства Windows (лицензия № 61137891 от 09.11.2012), либо Linux (свободная лицензия GNU/GPL).
2. Интернет браузер, например, Firefox, Google Chrome/Chromium, Midori, Opera или иные совместимые.
3. Средства разработки и тестирования:
  - (а) интерпретатор Python 3.5 или более поздней версии;
  - (б) среды разработки с открытым исходным кодом IDLE(X), Geany, Spyder3;
  - (в) библиотеки алгоритмов numpy, matplotlib, scipy.
4. Зональная научная библиотека им. В.А. Артисевич Саратовского государственного университета им. Н.Г. Чернышевского. – Режим доступа: <https://www.sgu.ru/structure/znbsgu>

## 9 Материально-техническое обеспечение дисциплины

Лекционные занятия по дисциплине «Информационные технологии и программирование» проводятся в аудиториях при помощи мультимедийного проектора, а также доски. Перед лекциями студентам может выдаваться печатный вариант краткого конспекта лекций.

Лабораторные занятия со студентами проходят в компьютерном классе, где установлено необходимое программное обеспечение и имеется достаточное количество персональных компьютеров. Также в классе имеется маркерная доска для объяснения студентам рабочих общих моментов практикума. Для проведения лабораторных занятий, включая практическую подготовку, необходим компьютерный класс факультета с установленным соответствующим программным обеспечением и доступом в Internet.

Самостоятельная работа студента с теоретическим материалом возможна с использованием электронных ресурсов дома, в научной библиотеке, и других местах, постоянный доступ в Интернет при этом не обязателен. Также студент может дома решать задачи практикума, используя домашний персональный компьютер, поскольку все задания практикума могут быть выполнены с использованием открытого свободного программного обеспечения.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению 03.03.01 «Прикладные математика и физика» и профилю подготовки «Нелинейные процессы в микроволновых системах».

Авторы — профессор кафедры системного анализа и автоматического управления, д.ф.-м.н., доцент Сысоев И. В.; доцент кафедры системного анализа и автоматического управления, к.ф.-м.н., Корнилов М. В.

Программа одобрена на заседании кафедры системного анализа и автоматического управления от 31 августа 2023 г., протокол № 1.