

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

УТВЕРЖДАЮ
Проректор по учебной работе,
качеству образования – первый
проректор
_____ Хагуров Т.А.

29 мая 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.37 ПРОГРАММИРОВАНИЕ

Направление подготовки:	44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность (профиль):	Математика, Информатика
Форма обучения:	очная
Квалификация:	бакалавр

Краснодар 2020

Рабочая программа дисциплины составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 44.03.05 Педагогическое образование

Программу составил:

Алексеев Е.Р., доцент кафедры информационных образовательных технологий, кандидат технических наук, доцент

Рабочая программа дисциплины утверждена на заседании кафедры информационных образовательных технологий протокол № 11 от 14.04.2020 г.
Заведующий кафедрой (разработчика) Грушевский С.П.

Рабочая программа обсуждена на заседании кафедры информационных образовательных технологий протокол № 11 от 14.04.2020 г.
Заведующий кафедрой (выпускающей) Грушевский С.П.

Утверждена на заседании учебно-методической комиссии факультета математики и компьютерных наук протокол № 2 от 30.04.2019 г.
Председатель УМК факультета Шмалько С.П.

Рецензенты:

Луценко Е.В., доктор экономических наук, профессор кафедры компьютерных технологий и систем КубГАУ

Кособуцкая Е.В., канд. физ.-мат. наук, доцент кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики КубГУ

1. Цели и задачи изучения дисциплины

1.1 Цель дисциплины

Формирование системы понятий, знаний и умений в области современного программирования, включающего в себя методы проектирования, анализа и создания программных продуктов, основанные на использовании процедурной и объектно-ориентированной методологий.

1.2 Задачи дисциплины:

формирование знаний, умений и навыков анализа и проектирования математических и информационных моделей реальных объектов или структур;

овладение умениями и навыками программирования типовых задач обработки информации (вычисления, сортировка, поиск и т.п.) в одной-двух современных средах программирования;

овладение умениями и навыками использования библиотек объектов (классов) для решения практических задач.

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Программирование» для бакалавриата по направлению «Педагогическое образование» относится к обязательной части Блока 1 «Дисциплины (модули)» учебного плана.

Дисциплина базируется на знаниях, полученных по стандарту общего среднего образования. Изучение дисциплины «Программирование» является базой для дальнейшего освоения студентами дисциплин «Теоретические основы информатики», «Информационные системы», «Математическая логика и теория алгоритмов», «Численные методы», курсов по выбору профессионального цикла, прохождения педагогической практики.

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций ОПК-8, ПКО-1, ПКО-6).

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ОПК-8	Способен осуществлять педагогическую деятельность на основе специальных научных знаний	основные конструкции языка программирования; о парадигмах программирования (императивной, функциональной, логической); о технологиях программирования (структурной, модульной, объектно-ориентированной);	понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; анализировать алгоритмы с использованием таблиц	терминологией по дисциплине; навыками алгоритмического мышления, формального описания алгоритмов
2.	ПКО-1	Способен осваивать и использовать	возможности профессионального самопознания и	осознанно осваивать и использовать	способами совершенствования

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
		базовые научно-теоретические знания и практические умения по предмету в профессиональной деятельности	саморазвития с применением компьютерных технологий; основные структуры данных и типовые методы обработки этих структур; основные операторы и инструкции языка программирования	языки и методы программирования в учебных программах базовых и элективных курсов в различных образовательных учреждениях	профессиональных знаний и умений путем использования возможностей компьютерных технологий
	ПКО-6	Способен поддерживать самостоятельность, инициативность обучающихся, способствовать развитию их творческих способностей в рамках учебно-исследовательской деятельности	содержание работы учителя информатики в средней школе; суть учебного проектирования и исследовательской деятельности по информатике	использовать в образовательном процессе разнообразные программные и интернет-ресурсы; выполнять руководство учебно-проектной деятельностью учащихся	навыками структурного программирования, алгоритмизации, работы в различных средах программирования (составление, отладка и тестирование программ; разработка и использование интерфейсных объектов)

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины за 1 курс составляет 5 зачетных единиц, их распределение по видам работ представлено в таблице.

Вид учебной работы	Всего часов	Семестры (часы)	
		1	2
Контактная работа, в том числе:	106,6	54,3	52,3
Занятия лекционного типа	34	18	16
Лабораторные работы	68	34	34
Контроль самостоятельной работы (КСР)	4	2	2
Иная контактная работа:	0,6	0,3	0,3
Самостоятельная работа, в том числе:	47	27	20
Курсовая работа	–	–	–

Проработка учебного (теоретического) материала	16	9	7
Выполнение индивидуальных заданий	16	9	7
Подготовка к текущему контролю	15	9	6
Контроль:	62,4	26,7	35,7
Подготовка к экзамену	62,4	26,7	35,7
Общая трудоемко сть	час.	216	108
	в том числе контактная работа	106,6	53,3
	зач. ед.	6	3

Общая трудоемкость дисциплины за 2 курс составляет 7 зачетных единиц, их распределение по видам работ представлено в таблице.

Вид учебной работы	Всего часов	Семестры (часы)	
		3	4
Контактная работа, в том числе:	128,5	56,2	72,3
Занятия лекционного типа	52	18	34
Лабораторные работы	68	34	34
Контроль самостоятельной работы (КСР)	8	4	4
Иная контактная работа (ИКР)	0,5	0,2	0,3
Самостоятельная работа, в том числе	70,1	25,1	45
Курсовая работа	–	–	–
Проработка учебного (теоретического) материала	23	8	15
Выполнение индивидуальных заданий	23	8	15
Подготовка к текущему контролю	24,1	9,1	15
Контроль:	53,4	26,7	26,7
Подготовка к экзамену	53,4	26,7	26,7
Общая трудоемко сть	час.	252	108
	в том числе контактная работа	128,5	72,3
	зач. ед.	7	4

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины
Разделы дисциплины, изучаемые в **первом** семестре

№ раздела	Наименование разделов	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛЗ	
1	2	3	4	5	6
1	Основы программирования	79	18	34	27
	Итого:		18	34	27

Примечание: Л – лекции, ЛЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента.

Разделы дисциплины, изучаемые во **втором** семестре

№ раздела	Наименование разделов	Количество часов		
		Всего	Аудиторная работа	Внеаудиторная работа

			Л	ЛЗ	СРС
1	2	3	4	5	6
2	Процедурное программирование. Особенности разработки программ с использованием процедур и функций.	23	5	11	7
3	Файлы, строки и записи	23	5	11	7
4	Графические возможности языка	24	6	12	6
	Итого:		16	34	20

Разделы дисциплины, изучаемые в **третьем** семестре

№ раздела	Наименование разделов	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛЗ	
1	2	3	4	5	6
5	Особенности разработки программ на С(С++). Переменные, операции и операторы языка	21	4	8	9
6	Указатели, массивы и функции в С(С++)	40,1	10	20	10,1
7	Файлы в С(С++)	16	4	6	6
	Итого:		18	34	25,1

Разделы дисциплины, изучаемые в **четвертом** семестре

№ раздела	Наименование разделов	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛЗ	
1	2	3	4	5	6
8	Объектно-ориентированное программирование	105,8	34	34	37,8
	Итого:		34	34	37,8

2.3 Содержание разделов дисциплины

2.3.1 Занятия лекционного типа

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1.	Основы программирования	Алфавит и словарь языка Паскаль. Структура программы. Данные. Константы и переменные. Раздел описания переменных. Типы данных.	Проверка домашнего задания, опрос

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
		<p>Выражения. Операции. Арифметические операции. Математические функции. Операторы. Оператор присваивания. Операторы ввода-вывода.</p> <p>Программирование линейных вычислительных процессов.</p> <p>Операции отношения. Операторы разветвления. Составной оператор. Программирование разветвляющихся процессов.</p> <p>Операторы цикла. Функции в Паскале. Программирование циклических процессов. Массивы. Программирование задач обработки массивов.</p>	
2.	Процедурное программирование. Особенности разработки и программ с использованием процедур и функций.	<p>Процедуры и функции. Формальные и фактические параметры. Параметры-значения и параметры-переменные. Локальные и глобальные переменные. Рекурсивные процедуры и функции. Опережающее описание. Модули. Программирование задач обработки матриц с использованием процедур, функций, модулей.</p>	Проверка домашнего задания, опрос, контрольная работа
3.	Файлы, строки и записи	<p>Файлы. Текстовые и двоичные файлы. Ввод-вывод с использованием файлов.</p> <p>Программирование задач обработки массив и матриц с использованием файлов.</p> <p>Символьный тип данных. Строковый тип данных. Записи. Процедуры и функции для обработки строк. Алгоритмы обработки строк. Записи обработка табличных данных.</p> <p>Программирование задач обработки таблиц.</p>	Проверка домашнего задания, опрос, контрольная работа
4.	Графические возможности языка	<p>Графические возможности языка. Графические примитивы. Алгоритм построения графика функции на экране дисплея. Построение динамических изображений. Построение графиков функций.</p>	Проверка домашнего задания, опрос, контрольная работа
5.	Особенности	Структура программы на языке C(C++). Функция	Проверка

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
	ти разработок и программ на C(C++). Переменные, операции и операторы языка	main(). Основные типы данных. Операции, операторы, функции C++. Ввод-вывод в C(C++). Операторы цикла в C++ Программирование разветвляющихся и циклических процессов. Использование простейших функции при программировании на C++.	домашнего задания, опрос, контрольная работа
6.	Указатели, массивы и функции в C(C++)	Массивы и указатели в C(C++). Особенности использования массивов в C++. Программирование задач обработки массивов с использованием функций в C++.	Проверка домашнего задания, опрос, контрольная работа
7.	Файлы в C(C++)	Файлы. Текстовые и двоичные файлы. Ввод-вывод с использованием файлов. Программирование задач обработки данных в C(C++) с использованием файлов.	
8.	Объектно-ориентированное программирование	Язык программирования Python. Интегрированная среды разработки на языке Python . Окно кода программы. Структура программы. Строки, списки, кортежи, словари. Модули языка. Линейные, разветвляющиеся, циклические алгоритмы. Обработка одномерных и двумерных массивов с использование модуля Numpy. Отладка программ. Функции. Структуры. Использование модуля matplotlib для построения графиков. Использование модуля numru при работе с файлами. Решение практических задач обработки данных с использованием numru, matplotlib. Объектно-ориентированная парадигма программирования. Объекты. Инкапсуляция, полиморфизм, наследование. Объектно-ориентированное проектирование.	Проверка домашнего задания

2.3.2 Занятия семинарского типа

Занятия семинарского типа - не предусмотрены

2.3.3 Лабораторные занятия

№ раздела	Наименование	Наименование лабораторных работ	Форма текущего контроля
-----------	--------------	---------------------------------	-------------------------

	раздела		
1	2	3	4
1.	Основы программирования	<ol style="list-style-type: none"> 1. Простейшие программы. 2. Линейные алгоритмы. Вычисление выражений. Стандартные функции. 3. Линейные алгоритмы. Программирование линейных алгоритмов. 4. Программирование разветвляющихся алгоритмов. 5. Операторы цикла. Простейшие функции. Программирование циклических алгоритмов. 6. Программирование задач обработки массивов. 	Защита лабораторных работ
2.	Процедурное программирование. Особенности разработки программ с использованием процедур и функций.	<ol style="list-style-type: none"> 1. Программирование задач обработки матриц с использованием подпрограмм. 2. Разработка личных модулей. 	Защита лабораторных работ
3.	Файлы, строки и записи	<ol style="list-style-type: none"> 1. Программирование задач обработки данных с использованием файлов. 2. Программирование задач обработки строк. 3. Обработка табличных данных с использованием текстовых и двоичных файлов. 	Защита лабораторных работ
4.	Графические возможности языка	<ol style="list-style-type: none"> 1. Программа построения графика непрерывной и разрывной функции. 2. Построение динамических изображений. 	Защита лабораторных работ
5.	Особенности разработки программ на C(C++). Переменные, операции и операторы языка	<ol style="list-style-type: none"> 1. Программирование разветвляющихся алгоритмов. 2. Программирование циклических алгоритмов. 	Защита лабораторных работ
6.	Указатели, массивы и функции в C(C++)	Программирование задач обработки массивов и матриц с использованием функций на языке C(C++).	Защита лабораторных работ
7.	Файлы в C(C++)	Программирование задач обработки данных в C(C++) с использованием файлов.	Защита лабораторных работ

№ раздела	Наименование раздела	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
			работ
8.	Объектно-ориентированное программирование	1. Программирование разветвляющихся процессов. 2. Задачи обработки массивов. 3. Построение графиков функций. 4. Использование файлов при программировании реальных задач обработки данных.	Защита лабораторных работ

2.3.4. Примерная тематика курсовых работ

Курсовые работы не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1.	Проработка лекционного материала	Основная литература, дополнительная литература, периодические издания, ресурсы сети Интернет. Методические указания по выполнению самостоятельной работы, утвержденные кафедрой информационных образовательных технологий, протокол №1 от 31 августа 2017 г.
2.	Чтение и анализ учебной и научной литературы	
3.	Подготовка к лабораторным занятиям	
4.	Подготовка к экзамену	

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

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

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,

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Семестр	Вид занятия	Используемые интерактивные	Количество
---------	-------------	----------------------------	------------

		образовательные технологии	часов
1	Лекции	Интерактивная лекция с мультимедийным оборудованием. Обсуждение сложных и дискуссионных вопросов и проблем.	80
	Лабораторные работы	Компьютерные занятия в режимах взаимодействия «преподаватель – студент», «студент - преподаватель», «студент – студент».	81
<i>Итого:</i>			161

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

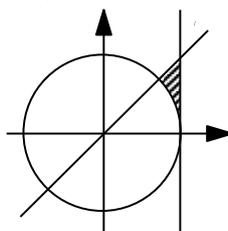
4.1 Фонд оценочных средств для проведения текущей аттестации Контрольная работа «Ветвления»

Вариант I

1. Составить логическое выражение, значение которого равно True, если высказывание истинно, и False, — если ложно.

В двузначном натуральном числе n первая цифра вдвое меньше второй.

2. Записать выражение, истинное тогда и только тогда, когда точка (x, y) принадлежит только заштрихованной области:



3. Изобразить на плоскости (X, Y) область, в которой и только в которой истинно указанное выражение

$$(X \leq 0) \text{ and } (\text{Sqr}(X) + \text{Sqr}(Y) \leq 4) \text{ or } (X \geq 0) \text{ and } (Y \leq -X + 2) \text{ and } (Y \geq X - 2)$$

Опрос «Условный оператор»

1. Введено четырехзначное число. Найти количество единиц в записи числа.
2. Введено трехзначное число. Если в записи этого числа содержится цифра 7, заменить ее на 9.
3. Введено четырехзначное число. Если оно оканчивается нулем, прибавить к нему первую цифру числа.
4. Введено трехзначное число. Если в записи числа нет цифры 5, то заменить последнюю цифру на 5.
5. Введено трехзначное число. Если сумма последних двух цифр нечетна, то поменять их местами.
6. Введено четырехзначное число. Найти количество цифр, кратных 5.
7. Вводятся целые X и Y . Если X делится на Y с остатком, то вывести величину остатка, иначе вывести текст « X кратно Y ».
8. Из чисел A, B, C выбрать максимальное.
9. Введено четырехзначное число. Проверить, равно ли произведение двух первых цифр произведению двух последних.
10. Даны три действительных числа. Найти сумму тех из них, значения которых неотрицательны.

11. Вводятся X и Y . Вывести сначала большее, а потом меньшее из них.
12. Вводятся двузначные X и Y . Вывести то число, у которого сумма цифр больше.

Контрольная работа «Циклы»

ВАРИАНТ 1

1. Вводится последовательность N целых чисел. Найти среднее арифметическое всех отрицательных нечетных чисел.
2. Дано натуральное число n . Найти сумму $S = 1 - 3 + 5 - \dots$ (n слагаемых).
3. Выполнить задание 1 для последовательности, оканчивающейся нулем.

Контрольная работа «Одномерные массивы»

ВАРИАНТ 1

1. Вводится массив 20 целых чисел. Найти среднее геометрическое всех отрицательных трехзначных чисел.
2. Дан целочисленный массив из n целых чисел. Упорядочить его по убыванию суммы цифр числа.
3. Дан целочисленный массив из 100 элементов. Удалить из массива все простые числа, в которых не четных цифр.

4.2 Фонд оценочных средств для проведения промежуточной аттестации

Вопросы к экзамену, 1 семестр

1. Принципы фон Неймана. Классификация языков программирования.
2. Алгоритм, его свойства, способы записи. Базовые алгоритмические конструкции, виды алгоритмов.
3. Понятие лексики, синтаксиса, семантики языка программирования.
4. Алфавит и словарь языка Паскаль. Структура программы.
5. Данные. Константы и переменные. Раздел описания переменных. Типы данных. Целые типы. Вещественные типы.
6. Выражения. Операции. Арифметические операции. Математические функции.
7. Операторы. Оператор присваивания. Операторы ввода-вывода.
8. Логический (булевский) тип. Операции отношения. Логические операции, порядок вычисления.
9. Условный оператор If. Полная и краткая форма оператора. Пример. Составной оператор.
10. Функции `div` и `mod`. Вычисление суммы цифр произвольного целого числа.
11. Оператор выбора Case. Полная и краткая форма оператора. Пример.
12. Оператор цикла For. Программа проверки, является ли число простым. Замена For на While и возможность замены While на For.
13. Оператор цикла Repeat. Программа вычисления суммы ряда.
14. Операторы BREAK, CONTINUE используемые внутри циклов. Оператор EXIT.
15. Одномерные массивы. Типовые алгоритмы для одномерных массивов. Пример.
16. Вложенные циклы. Пример использования.
17. Этапы решения задач на ЭВМ. Ручная прокрутка программ. Отладка программ в Паскале.
18. Сортировка массивов.

Примерные задания к экзамену, 1 семестр.

1. Разложить заданное натуральное число на простые множители.
2. Нахождение минимума и максимума двух, трех, четырех данных чисел без использования массивов и циклов.
3. Нахождение всех корней заданного квадратного уравнения.

4. Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива).
5. Использование цикла для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту и т.д.).
6. Заполнение элементов одномерного и двумерного массивов по заданным правилам.
7. Операции с элементами массива. Линейный поиск элемента. Вставка и удаление элементов в массиве. Перестановка элементов данного массива в обратном порядке. Суммирование элементов массива.
8. Проверка соответствия элементов массива некоторому условию.
9. Нахождение второго по величине (второго максимального или второго минимального) значения в данном массиве за однократный просмотр массива.
10. Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.
11. Операции с элементами массива, отобранных по некоторому условию (например, нахождение минимального четного элемента в массиве, нахождение количества и суммы всех четных элементов в массиве).
12. Сортировка массива.
13. Слияние двух упорядоченных массивов в один без использования сортировки.
14. Обработка отдельных символов данной строки. Подсчет частоты появления символа в строке.
15. Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.

Вопросы к экзамену, 2 семестр

1. Процедуры в Паскале. Параметры-значения и параметры-переменные.
2. Функции в Паскале.
3. Механизм передачи данных в подпрограмму.
4. Рекурсивные функции.
5. Основные алгоритмы обработки массивов: сумма, произведение, максимум и минимум, сортировка, удаление элемента из массива, вставка элемента в упорядоченный массив.
6. Решение задач на обработку массивов с использованием подпрограмм.
7. Матрицы в Паскале.
8. Основные алгоритмы обработки матриц: ввод-вывод матрицы, сумма и произведение элементов матрицы, поиск максимума и минимума, обмен элементов местами, построчная (постолбцовая) обработка матриц.
9. Текстовые файлы.
10. Алгоритм решения системы линейных алгебраических уравнений методом Гаусса.
11. Реализация алгоритма решения системы линейных алгебраических уравнений методом Гаусса.
12. Алгоритм нахождения определителя.
13. Реализация алгоритма нахождения определителя.
14. Алгоритм нахождения обратной матрицы.
15. Реализация алгоритма нахождения обратной матрицы.

Примерные задания к экзамену, 2 семестр.

1. Задан массив целых чисел $X(n)$. Все *простые числа* переписать в массив Y . Из массива Y удалить 5 наибольших элементов массива. Вывести на экран содержимое массива Y в *двоичной системе*.
2. Заданы массивы целых чисел $X(n)$ и $Y(k)$. Все *совершенные числа* из этих массивов переписать в массив Z . В массиве Z найти четыре наименьших элемента массива. Удалить из массива Z все нулевые элементы. Результаты вывести на экран в *восьмеричной системе*.

3. Заданы массивы целых чисел $X(n)$ и $Y(k)$. Два наибольших элемента из массива X и пять последних *простых чисел* из массива Y переписать в массив Z . Проверить содержит ли массив Z числа, в которых есть цифра «7».
4. Заданы массивы целых чисел $X(n)$ и $Y(k)$. Три наименьших *простых числа* из массива Y и числа из массива X , в которых есть цифры «1» и «9» переписать в массив Z . Из массива Z удалить все нечетные числа.
5. Задан массив целых чисел $X(n)$. Шесть наибольших чисел этого массива переписать в массив Z . Удалить из массива Z все четные числа. Вывести на экран элементы массива Z в *восьмеричной системе счисления*.
6. Заданы массивы целых чисел $X(n)$ и $Y(k)$. Числа из массива X , в которых нет «нулей» и *составные числа* из массива Y , переписать в массив Z . Найти в массиве Z пять наибольших нечетных чисел. Выполнить сортировку массивов X , Y и Z в порядке возрастания их элементов.
7. Заданы массивы целых положительных чисел. $X(n)$ - в двоичной системе счисления, а $Y(k)$ — в восьмеричной. Все числа из массивов X и Y переписать в массив десятичных чисел Z . В массиве Z найти пять наибольших *простых чисел*. Удалить из массива Z все *составные числа*.
8. Задан массив целых положительных чисел $X(n)$. Все *простые числа* длиной не более пяти цифр переписать в массив Y . Удалить из массива два наибольших и три наименьших числа.
9. Задан массив целых положительных чисел в пятеричной системе $X(n)$. Из массива X сформировать массив десятичных чисел Z . Найти сумму трех наименьших и четырех наибольших чисел массива Z .
10. Задана матрица целых чисел $A(n \times n)$. Максимальный элемент каждого столбца заменить *суммой цифр* максимального элемента матрицы. Сформировать массив $B(n)$, в который записать количество четных элементов в каждой строке заданной матрицы.
11. Задана матрица целых чисел $B(n \times m)$. Максимальный элемент каждого столбца заменить *суммой цифр* модуля минимального элемента матрицы. Сформировать массив $A(n)$, в который записать количество нечетных элементов в каждой строке заданной матрицы.
12. Задана матрица целых чисел $A(n \times n)$. Сформировать массив $B(n)$ из максимальных элементов столбцов заданной матрицы. Вывести номера строк, в которых *числа-палиндромы* находятся на диагоналях матрицы.
13. Задана матрица вещественных чисел $P(n \times m)$. Сформировать массив $R(k)$ из номеров столбцов матрицы, в которых есть хотя бы один ноль. Найти строку с максимальной суммой элементов и поменять ее с первой строкой.
14. Задана матрица вещественных чисел $C(k \times m)$. Сформировать вектор $D(k)$ из средних арифметических положительных значений строк матрицы, и вектор $G(n)$ из номеров столбцов, которые представляют собой *знакопередающийся ряд*.
15. В каждом столбце матрицы вещественных чисел $P(k \times m)$ заменить минимальный элемент суммой положительных элементов этого же столбца. Сформировать вектор $D(n)$ из номеров строк, представляющих собой *знакопередающийся ряд*.
16. В матрице целых чисел $A(n \times m)$ обнулить строки, в которых более двух *простых чисел*. Сформировать вектор $D(m)$ из минимальных значений столбцов матрицы.
17. В матрице вещественных чисел $P(n \times m)$ найти и вывести номера столбцов, упорядоченных по убыванию элементов. Сформировать вектор $R(n)$ из максимальных значений строк матрицы.

Вопросы к экзамену, 3 семестр

1. Типы данных.
2. Структура программы.
3. Операции и операторы C(C++).

4. Статическая и динамическая память.
5. Адреса и указатели. Структура указателя. Объявление указателей. Типизированные и нетипизированные указатели.
6. Способы задания значения указателя.
7. Выделение и освобождение динамической памяти.
8. Массивы и указатели в C++.
9. Функции в C++.
10. Текстовые файлы.
11. Двоичные файлы.
12. Несвязанные динамические переменные.
13. Связанные динамические структуры данных.
14. Однонаправленные списки. Структура узла списка. Типовые процедуры обработки списка.
15. Очереди. Типовые процедуры обработки очереди.
16. Стеки. Типовые процедуры обработки стека.
17. Алгоритм проверки правильности скобочного выражения с использованием стека.
18. Алгоритм вычисления выражения в постфиксной записи с использованием стека.
19. Двоичные деревья, структура узла дерева, правила обхода деревьев.
20. Реализация алгоритма обхода двоичного дерева.
21. Алгоритм сортировки с использованием двоичного дерева.
22. Графы. Кратчайшие пути. Алгоритм Дейкстры.
23. Трансляция программ. Интерпретаторы и компиляторы.
24. Упрощенная модель компилятора. Лексический анализ.
25. Упрощенная модель компилятора. Синтаксический анализ.
26. Упрощенная модель компилятора. Генерация кода.
27. Упрощенная модель компилятора. Оптимизация кода.
28. Упрощенная модель компилятора. Ассемблирование.

Вопросы к экзамену, 4 семестр

1. Состав языка PYTHON. Алфавит, идентификаторы, ключевые слова, знаки операций, константы, комментарии.
2. Основные типы данных PYTHON. Структура программы.
3. Переменные, операции и выражения в языке PYTHON.
4. Модули в программе на языке PYTHON, три варианта использования команды подключения модуля import.
5. Ввод и вывод данных в языке PYTHON. Функции input, int, map, split. print (с аргументами sep, end, format).
6. Операторы ветвления в языке PYTHON. Каскадное ветвление (elif). Примеры.
7. Операторы цикла for в языке PYTHON. Пример.
8. Операторы цикла while в языке PYTHON. Пример: определить количество цифр в десятичной записи целого положительного числа.
9. Процедуры и функции в языке PYTHON. Примеры.
10. Одномерные массивы в языке PYTHON (списки, тип данных list). Способы ввода и вывода массива. Стандартные алгоритмы обработки массивов.
11. Алгоритмы обработки массивов с использованием встроенных функций Python. Алгоритм поиска заданного элемента: требуется найти в массиве элемент, равный значению переменной X, или сообщить, что его там нет.
12. Алгоритмы обработки массивов с использованием встроенных функций Python. Алгоритм поиска максимального элемента: требуется найти значение и номер максимального элемента.
13. Алгоритмы обработки массивов с использованием встроенных функций Python. Реверс массива.

14. Алгоритмы обработки массивов с использованием встроенных функций Python. Циклический сдвиг элементов массива на один элемент влево.
15. Алгоритмы обработки массивов с использованием встроенных функций Python. Отобразить все четные элементы массива А в новый массив В.
16. Алгоритмы обработки массивов с использованием встроенных функций Python. Сортировка массива, способы задания ключей сортировки, использование лямбда-функции.
17. Алгоритмы обработки матриц с использованием встроенных функций Python. Ввод, вывод, суммирование матрицы.
18. Дисциплина программирования, структурный подход к программированию.
19. Возникновение объектно-ориентированной парадигмы программирования. Концептуальное отличие процедурного программирования от объектно-ориентированного программирования. Преимущества и недостатки решения задачи с помощью ООП.
20. Этапы объектно-ориентированного анализа.
21. Понятия класса и объекта в ООП, различие между ними. Свойства объекта (поля), методы, события. Пример.
22. Основные принципы объектно-ориентированного программирования. Инкапсуляция. Пример.
23. Основные принципы объектно-ориентированного программирования. Полиморфизм. Процедуры и функции как реализация методов. Пример.
24. Основные принципы объектно-ориентированного программирования. Наследование и иерархия объектов. Примеры.
25. Объектно-ориентированное программирование. Объявление переменной для работы с объектом некоторого класса. Конструктор и деструктор. Создание объекта.
26. Абстрактные классы. Пример.
27. Применение библиотек и иерархий объектов при программировании.
28. Интерфейс объекта. Скрытие внутреннего устройства объекта. Открытые (public) и закрытые (protected).
29. Доступ к полям объекта с помощью методов. Пример.
30. Интегрированная среда программирования Wing IDE 101.
31. Среда, управляемая событиями. Событие и сообщение. Передача сообщений между объектами. Пример.
32. Математические объекты: рациональные числа. Описание их свойств и методов в PYTHON.
33. Математические объекты: комплексные числа. Описание их свойств и методов в PYTHON.
34. Математические объекты: вектора. Описание их свойств и методов в PYTHON.
35. Математические объекты: матрицы. Описание их свойств и методов в PYTHON.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических – при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление

информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

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

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.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1 Основная литература:

1. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 161 с. — (Бакалавр. Прикладной курс). — ISBN 978-5-534-10971-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/437489>.
2. Окулов, С.М. Программирование в алгоритмах / С.М. Окулов. — 6-е изд. — Москва : Лаборатория знаний, 2017. — 386 с. — ISBN 978-5-00101-449-2. — Текст : электронный // Электронно-библиотечная система «Лань»: [сайт]. — URL: <https://e.lanbook.com/book/94140>
3. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для прикладного бакалавриата / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2019. — 235 с. — (Бакалавр. Прикладной курс). — ISBN 978-5-534-02816-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/433611> (дата обращения: 11.09.2019).

5.2 Дополнительная литература:

4. Долинер Л. И. [Основы программирования в среде PascalABC.NET: учебное пособие](#). Екатеринбург. Изд-во Урал. ун-та. 2014. – 129 с. [Электронный ресурс, ЭБС «Университетская библиотека ONLINE»], URL: http://biblioclub.ru/index.php?page=book_red&id=275988.
5. Буйначев С. К., Боклаг Н. Ю. Основы программирования на языке Python: учебное пособие. Издательство Уральского университета, 2014. 92 с. [Электронный ресурс, ЭБС «Университетская библиотека ONLINE»], URL: http://biblioclub.ru/index.php?page=book_red&id=275962
6. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для бакалавриата и специалитета / М. : Издательство Юрайт, 2018. – 335 с. [Электронный ресурс]. URL: <https://biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4/programirovanie-na-yazyke-s-prakticheskiy-kurs>.
7. Алексеев Е.Р, Чеснокова О., Кучер Т. Программирование на Free Pascal и Lazarus / - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 552 с.: ил. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_red&id=429189&sr=1.
8. Окулов С.М. Программирование в алгоритмах [Электронный ресурс]. - М.: Лаборатория знаний, 2017. - 386 с. URL: <https://e.lanbook.com/book/94140#authors> .

9. Северенс Ч. Введение в программирование на Python [Электронный ресурс] / Ч. Северенс. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 231 с. URL: <https://biblioclub.ru/index.php?page=book&id=429184>.
10. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата. [Электронный ресурс] М.: Издательство Юрайт, 2018. — 155 с. URL: <https://biblio-online.ru/book/programmirovanie-obektno-orientirovannyu-podhod-414203>.

5.3 Периодические издания

1. Журнал «Информатика и образование».
2. Журнал «Информатика в школе».

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Современное программирование на языке Паскаль. URL <http://pascalabc.net/>.
2. Веб-среда разработки ProgrammingABC.NET WDE (www.pascalabc.net/WDE).
3. www.freepascal.org.
4. <http://www.pascal-central.com/>.
5. К. Поляков. Язык Python. URL: // <http://kpolyakov.spb.ru/school/probook/python.htm>.
6. Поляков К.Ю. Газета «Информатика». Язык Python глазами учителя. – М.: Первое сентября, 2014 – № 9 – С. 4–16 // URL: <http://kpolyakov.spb.ru/download/inf-2014-09a.pdf>.
7. К. Поляков. Язык Python: избранные алгоритмы. Часть 1 // Информатика, № 9, 2014, с. 18-26. // URL: <http://kpolyakov.spb.ru/download/inf-2014-09b.pdf/>
8. К. Поляков. Объектно-ориентированное программирование // URL: <http://kpolyakov.spb.ru/school/c.htm>.
9. Кириенко Д.П. Программирование на Python [Электронный ресурс] URL: <http://server.179.ru/wiki/?page=DenisKirienko/Python>.
10. Кириенко Д.П. Программирование на Python [Электронный ресурс] URL: <http://informatics.mccme.ru/course/view.php?id=156>
11. Сайт дистанционной подготовки по информатике <http://informatics.mccme.ru/moodle/>.
12. Язык программирования Python. Учебное пособие <http://www.intuit.ru/studies/courses/49/49/info>.
13. Столяров А.В. Введение в язык C++. <http://www.stolyarov.info/books/pdf/cppintro3.pdf>.

7. Методические указания для обучающихся по освоению дисциплины

На лекциях преподаватель рассматривает вопросы программы курса, составленной в соответствии с государственным образовательным стандартом. Из-за недостаточного количества аудиторных часов некоторые темы не удастся осветить в полном объеме, поэтому преподаватель некоторые вопросы выносит на самостоятельную работу студентов, рекомендуя ту или иную литературу.

Для освоения материала и систематизации знаний по дисциплине необходимо постоянно разбирать материалы лекций по конспектам и учебным пособиям и в случае необходимости обращаться к преподавателю за консультацией. Полный список источников по дисциплине приведен в п. 5, 6.

После изучения некоторых разделов курса «Программирование» на лабораторных занятиях проводятся контрольные аудиторные работы. Для успешного их написания, готовиться к контрольным работам нужно по материалам лекций, презентаций и рекомендованных источников.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю) (при необходимости)

8.1 Перечень информационных технологий

Мультимедийные лекции; демонстрационные примеры программ; использование компьютера при выдаче заданий и проверке их решений, выполнении лабораторных работ; компьютерное тестирование по итогам изучения разделов дисциплины.

8.2 Перечень необходимого программного обеспечения

1. Операционная система MS Windows или Linux.
2. Интегрированное офисное приложение MS Office (LibreOffice).
3. Программное обеспечение для организации управляемого коллективного и безопасного доступа в Интернет.
4. Среда программирования PascalABC.Net (FreePascal, Lazarus).
5. Среда программирования на языке Python.

8.3 Перечень информационных справочных систем:

Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru/>)

Электронная библиотечная система "Университетская библиотека онлайн" (<https://biblioclub.ru/>)

[Электронная библиотечная система издательства "Лань" https://e.lanbook.com](https://e.lanbook.com)

[Электронная библиотечная система "Юрайт" http://www.biblio-online.ru/](http://www.biblio-online.ru/)

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

№	Вид работ	Материально-техническое обеспечение дисциплины (модуля) и оснащенность
1.	Лекционные занятия	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением (ПО): ауд. 303Н, 308Н, 505А, 507А
2.	Лабораторные занятия	Компьютерный класс с необходимым программным обеспечением, локальной сетью и выходом в Интернет для проведения лабораторных работ: ауд. 301Н, 309Н, 316Н, 320Н
3.	Групповые (индивидуальные) консультации	Ауд. 301Н, 302Н, 303Н, 307Н, 308Н, 308На, 310Н, 312Н, 314Н, 316Н, 318Н, 320Н, 505А, 507А
4.	Текущий контроль, промежуточная аттестация	Ауд. 301Н, 302Н, 303Н, 307Н, 308Н, 308На, 309Н, 310Н, 312Н, 314Н, 318Н, 320Н, 505А, 507А
5.	Самостоятельная работа	Ауд. 304Н, 305Н, 307Н, 308На, 309Н, 310Н, 312Н, 314Н, 316Н, 318Н