

Министерство науки и высшего образования  
Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Донецкий государственный университет»

Факультет физико-технический  
Кафедра радиоп физики и инфокоммуникационных технологий



УТВЕРЖДАЮ

проректор

П.А. Машаров

«29» марта 2024 г.

МП

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»**

Укрупненная группа направлений подготовки	10.00.00 Информационная безопасность
Программа высшего образования	Программа бакалавриат
Направление подготовки	10.03.01 Информационная безопасность
Профиль подготовки	Безопасность автоматизированных систем
Квалификация	Бакалавр
Форма обучения	очная

Рабочая программа адаптирована для лиц  
с ограниченными возможностями здоровья и инвалидов

Донецк 2024

Рабочая программа дисциплины «Технологии и методы программирования» для обучающихся по направлению подготовки 10.03.01 Информационная безопасность (Профиль: Безопасность автоматизированных систем), составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 10.03.01 Информационная безопасность, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 17 ноября 2020 г. № 1427 (с изм. и доп.). Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 06 апреля 2021 г. № 245 (с изм. и доп.), в соответствии с учебным планом, утвержденным Ученым советом ФГБОУ ВО «ДонГУ» для набора 2024 года.

Разработчик:

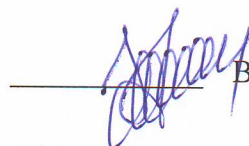
Ст. преподаватель  
кафедры радиопизики  
и инфокоммуникационных технологий



Е.Н. Кожекина

Рабочая программа утверждена на заседании кафедры радиопизики и инфокоммуникационных технологий  
Протокол от 26.03.2024 г. № 16

Заведующий кафедрой



В.В. Данилов

СОГЛАСОВАНО:

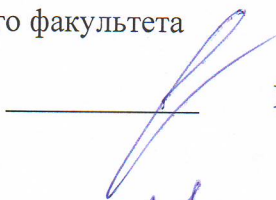
И.о. декана физико-технического факультета  
28.03.2024 г.



С.А. Фоменко

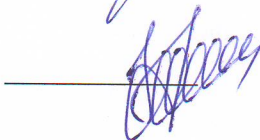
Учебно-методическая комиссия физико-технического факультета  
Протокол от 27.03.2024 г. № 2

Председатель



В. Н. Котенко

Руководитель основной профессиональной  
образовательной программы  
д-р тех. наук, проф.  
26.03.2024 г.



В.В. Данилов

## 1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Требования к предварительной подготовке обучающихся, предшествующие и сопутствующие дисциплины, на которых основывается изучение данной:  
базовая подготовка по математике в объёме программы средней школы;  
дисциплины программы бакалавриата: Математический анализ, Информатика.

## 2. ОПИСАНИЕ ДИСЦИПЛИНЫ

### 2.1. Общая характеристика

Наименование показателя	Значение показателя
Название образовательной программы	10.03.01 Информационная безопасность (Программа бакалавриата: 10.03.01 Информационная безопасность (Профиль: Безопасность автоматизированных систем))
Шифр и название в соответствии с учебным планом	Б1.Б.М3.2 Технологии и методы программирования
Часть образовательной программы	Базовая часть
Количество зачетных единиц / всего часов	8 / 288

### 2.2. Распределение часов по формам и периодам обучения

Форма обучения	курс	семестр	Общее количество часов					Форма контроля
			лекционных	лабораторных	практических	самостоятельной работы + контроль	всего	
Очная	1	2	30	30	-	84	144	зачет
Очная	2	3	34	34	-	76	144	экзамен
Очная, всего			64	64	-	160	288	

## 3. ЦЕЛИ ДИСЦИПЛИНЫ

Приобретение умений разработки и тестирования программ на высокоуровневом языке Python 3. Развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений. Обучение особенностям применения различных технологий программирования, принципам построения и анализа алгоритмов задач в предметной области. Обучение оценке сложности поставленной задачи и выбору наиболее оптимального алгоритма и соответствующего языка программирования для её реализации. Подготовка к активному использованию методов и средств, а также основ программирования при решении задач выбранной специальности.

## 4. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ КОМПОНЕНТА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ, ИХ ИНДИКАТОРЫ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

### 4.1. Компетенции

Компетенции	Индикаторы	Результаты обучения
ОПК-7. Способен использовать языки программирования	ОПК-7.1. Владеет навыками и технологиями	ОПК-7.1.1. Знает теоретические основы алгоритмизации и программирования, построения сетей коммуникаций, знает

и технологии разработки программных средств для решения задач профессиональной деятельности;	разработки приложений для решения задач информационной безопасности	несколько языков программирования на уровне решения стандартных задач. ОПК-7.1.2. Знает теоретические основы алгоритмизации и программирования, построения сетей коммуникаций, основные алгоритмы обработки данных, знает несколько языков программирования на уровне создания программного обеспечения прикладного уровня ОПК-7.1.3. Знает теоретические основы алгоритмизации и программирования, построения сетей коммуникаций, приемы разработки новых алгоритмов обработки данных, знает несколько языков программирования на уровне создания программного обеспечения системного и прикладного уровня
--	---	---

## 5. ПРОГРАММА ДИСЦИПЛИНЫ

Название темы	Краткое содержание темы (вопросы темы)
<b>Семестр:1</b>	
<b>Раздел 1. Основы теории множеств</b>	
Определения, примеры, операции над множествами	Множества и операции над ними. Простейшие теоретико-множественные тождества
<b>Раздел 2. Логика высказываний</b>	
Понятие формулы. Таблицы истинности. Эквивалентность формул. Основные тождества логики высказываний и теории множеств.	Таблицы истинности. Эквивалентность формул. Основные пропозициональные связки в логике высказываний: -отрицание («неверно, что»), & - конъюнкция («и»), $\vee$ - нестрогая дизъюнкция («или»), $\vee$ - строгая дизъюнкция («либо... либо»), $\rightarrow$ - импликация («если... то»), $\leftrightarrow$ - эквиваленция («если и только если»)
Нормальные формы. Приведение формулы к СДНФ и СКНФ	Рассматриваются свойства булевых функций, методы их минимизации и приведения к нормальным формам, описание всех замкнутых классов булевых функций, а также методы определения полноты систем таких функций. Кроме того, рассматриваются вопросы практического применения булевой алгебры при построении математических моделей в различных областях
<b>Раздел 3. Логика предикатов</b>	
Примеры. Термы и формулы логики предикатов. Истинность формулы на модели. Тождественно истинные и выполнимые формулы.	Логические теории разных типов. Логика предикатов первого порядка, логика предикатов высших порядков.
Семантическая эквивалентность формул. Основные тождества логики предикатов.	Объекты, их свойства, отношения между объектами и функции. Утверждения о свойствах объектов и отношениях между ними. Предикаты. Синтаксис логики предикатов. Семантика логики

	предикатов: системы, состояния и значения формул на состояния
<b>Раздел 4. Отношения и функции</b>	
Предпорядок, отношения эквивалентности и частичного порядка. Понятие решетки.	Эквивалентность и разбиение, фактор-множество. Максимальные и минимальные, наибольшие и наименьшие элементы, точная верхняя и нижняя грани.
<b>Раздел 5. Булевы алгебры</b>	
Основные свойства булевой алгебры. Примеры. Атомные и безатомные элементы булевых алгебр. Конечные булевы алгебры, теорема Стоуна для конечных булевых алгебр.	Множество-степень, понятие и основные свойства булевой алгебры. Примеры. Атомные и безатомные элементы булевых алгебр. Конечные булевы алгебры, теорема Стоуна для конечных булевых алгебр.
<b>Раздел 6. Мощность множества</b>	
Понятие равномощности множеств. Теорема Кантора-Бернштейна. Теорема Кантора. Бесконечность класса бесконечных мощностей. Парадоксы теории множеств	Понятие равномощности множеств. Теорема Кантора-Бернштейна. Теорема Кантора. Бесконечность класса бесконечных мощностей. Парадоксы теории множеств
<b>Раздел 7. Счетные и континуальные множества</b>	
Счетные и континуальные множества	Счётные множества. Счётность множества слов в счётном алфавите. Континуум. Несчётность множества вещественных чисел. Равномощность множества вещественных чисел и множества всех подмножеств множества натуральных чисел. Континуум-гипотеза и обобщённая континуум-гипотеза. Ординальные и
<b>Раздел 8. Машины Тьюринга</b>	
Машины Тьюринга. Операторы над машинами Тьюринга. Функции, вычислимые на машинах Тьюринг	Понятие алгоритма. Вычислимые функции. Машины Тьюринга. Операторы над машинами Тьюринга. Функции, вычислимые на машинах Тьюринг
<b>Раздел 9. Частично рекурсивные функции</b>	
Частично рекурсивные функции. Канторовская нумерация.	Операторы суперпозиции, примитивной рекурсии и минимизации. Примитивно-рекурсивные, общерекурсивные и частично рекурсивные функции. Канторовская нумерация.
<b>Раздел 10. Секвенциальное исчисление высказываний.</b>	
Секвенциальное исчисление высказываний: Аксиомы и правила вывода.	Секвенциальное исчисление высказываний: Аксиомы и правила вывода. Допустимые правила вывода. Теорема о корректности. Теорема о подстановке. Теорема о замене. Теорема о существовании КНФ. Теорема о полноте исчисления секвенций
<b>Раздел 11. Исчисление высказываний гильбертовского типа</b>	
Аксиомы и правила вывода. Теорема об эквивалентности исчисления секвенций. Теорема о дедукции	Аксиомы и правила вывода. Теорема об эквивалентности исчисления секвенций и исчисления высказываний гильбертовского типа. Теорема о дедукции
<b>Семестр:2</b>	

<b>Раздел 1. Базовые конструкции языка Python.</b>	
Компилируемые и интерпретируемые языки программирования. Интерпретатор Python. Стандарт кода PEP8. Переменные, оператор присваивания, функции type и id. Числовые типы, арифметические операции. Особенности округления при целочисленном делении в языках C++ и Python	Переменные, оператор присваивания, функции type и id. Числовые типы, арифметические операции. Функции print() и input(). Преобразование строк в числа int() и float(). Динамическая типизация. Изменяемые и неизменяемые типы данных в Python. Целочисленная арифметика. Особенности округления при целочисленном делении в языках C++ и Python. Логический тип bool. Операторы сравнения и операторы and, or, not.
<b>Раздел 2. Условные операторы, циклы, генераторы списков</b>	
Условный оператор, логические операции, тернарный условный оператор, вложенные и каскадные условия	Условный оператор if. Конструкция if-else. Вложенные условия и множественный выбор. Конструкция if-elif-else. Тернарный условный оператор. Вложенное тернарное условие. Оператор цикла while. Операторы циклов break, continue и else.
Операторы циклов, генераторы списков (List comprehensions). вложенные генераторы списков. Обработка исключений. Инструкции assert в Python	Оператор цикла for. Функция range(). Примеры работы оператора цикла for. Функция enumerate(). Итератор и итерируемые объекты. Функции iter() и next(). Вложенные циклы. Примеры задач с вложенными циклами. Треугольник Паскаля как пример работы вложенных циклов. Генераторы списков (List comprehensions). Вложенные генераторы списков
<b>Раздел 3. Последовательности в Python.</b>	
Строки. Методы строк.	Строки. Методы строк. Срезы (slices). Форматирование строк. Метод format, f-строка. Оператор '%' при выводе значений.
Списки, кортежи, двумерные массивы	Списки. Хранение в памяти. Создание списка. Срезы (slice). Простые операции. Методы списков. Обращение к элементам кортежа. Получение подкортежей. Кортеж как параметр и результат функций. Перебор кортежей. Проверка наличия значения.
Словари, множества (set и frozenset)	Словари, методы словарей. Преобразование списков и кортежей в словарь. Получение, изменение, удаление элементов. Копирование и объединение словарей. Перебор словаря. Комплексные словари. Операции над множествами, диаграммы Эйлера-Венна. Введение в множества в Python. Основы работы с множествами. Методы множеств. Генераторы множеств и frozenset.
<b>Раздел 4. Функции в python: синтаксис, логика и применение. Элементы функционального программирования</b>	
Функции, параметры и аргументы. Операторы return, yield в функциях.	Функции: определение def и их вызов. Оператор return в функциях. Функциональное

Области видимости переменных. Ключевые слова global и nonlocal. Замыкания в Python.	программирование. Позиционные и именованные аргументы. Фактические и формальные параметры. Функции с произвольным числом параметров *args и **kwargs. Операторы * и ** для упаковки и распаковки коллекций. Рекурсивные функции. Области видимости переменных. Ключевые слова global и nonlocal. Замыкания в Python.
Анонимные (lambda) функции. Функции map, filter, reduce, zip. Введение в декораторы функций. Декораторы с параметрами	Анонимные (lambda) функции. Функции map, filter, reduce, zip. Введение в декораторы функций. Декораторы с параметрами. Сохранение свойств декорируемых функций
<b>Раздел 5. Рекурсивные алгоритмы</b>	
Рекурсивные алгоритмы. Сочетания. Размещения. Оптимизация рекурсии. Кривая Коха	Рекурсивные алгоритмы. Факториал. Числа Фибоначчи. Ханойская башня. Задача о 8 ферзях. Оптимизация задачи о восьми ферзях. Сочетания. Размещения. Оптимизация рекурсии. Кривая Коха
<b>Раздел 6. Модули и пакеты в Python</b>	
Стандартные модули. Импортирование модулей. Собственные модули. Специализированные модули.	Основные стандартные модули и пакеты в Python. Импортирование модулей. Модули itertools и functools. Создание собственных модулей и их импортирование. Специализированные модули и приложения
<b>Раздел 7. Работа с файлами, файловой системой и процессами в Python</b>	
Основные возможности Python по работе с файлами и файловой системой. Модули os, shutil, pathlib, glob. Работа с датой и временем	Основные возможности Python по работе с файлами и файловой системой. Модули os, shutil, pathlib, glob. Работа с датой и временем
Аргументы командной строки, модуль argparse. Сериализация pickle, json, yaml. Работа с внешними процессами. Модуль subprocess	Аргументы командной строки, модуль argparse. Сериализация pickle, json, yaml. Работа с внешними процессами. Модуль subprocess
<b>Раздел 8. Сложность алгоритмов</b>	
Линейный поиск. Сложность алгоритмов. Нотации O, $\Omega$ , $\Theta$ . Бинарный поиск	Линейный поиск. Сложность алгоритмов. Нотации O, $\Omega$ , $\Theta$ . Бинарный поиск.
Квадратичные сортировки.	Сортировка выбором. Сортировка методом пузырька. Сортировка вставками. Сортировка подсчетом. Поразрядная сортировка (Radix Sort). Синхронная сортировка массивов
Быстрые сортировки. Устойчивость сортировок.	Быстрая сортировка (Тони Хоара). Сортировка слиянием.
<b>Раздел 9. Перебор. Жадные алгоритмы. Алгоритмы динамического программирования</b>	
Жадные алгоритмы	Характеристики жадного алгоритма. Примеры жадных алгоритмов (размен монет, задача о рюкзаке). Применение жадных алгоритмов. Преимущества использования жадных алгоритмов. Недостатки/ограничения использования жадных алгоритмов

Задачи одномерного динамического программирования	Динамическое программирование. Мемоизация. Числа Фибоначчи, кузнечик, модификации задачи о кузнечике. Задача о размене монет. Восстановление ответа
Задачи двумерного динамического программирования	Самая длинная общая подпоследовательность. Задача о рюкзаке. Расстояние Левенштейна (редакционное расстояние). Самый большой подмассив. Размен монет. Самая длинная возрастающая подпоследовательность. Умножение цепочки матриц. Задача коммивояжера. 0-1 Целочисленное программирование. Расстояние Левенштейна с разрешенными операциями. Самая длинная палиндромная подстрока. Задача о подмассиве максимального произведения (Maximum Product Subarray). Самый большой прямоугольник на гистограмме. Бросание яиц. Подсчет битов. Идеальные квадраты. Раздел равной суммы подмножества (Partition Equal Subset Sum). Самая длинная общая подстрока. Уникальные пути. Расстояние Левенштейна с помощью разрешенных операций. Проблема суммы подмножества. Самая длинная палиндромная последовательность. Самый большой прямоугольник на гистограмме. Уникальные пути Проверка равенства строк
<b>Раздел 10. Строковые алгоритмы</b>	
Простейший поиск подстроки. Суффиксы, префиксы, грани. Префикс-функция. Алгоритм Кнута - Морируса – Пратта. Алгоритм Бойера – Мура. Алгоритм Бойера - Мура на практике	Простейший поиск подстроки. Суффиксы, префиксы, грани. Префикс-функция. Алгоритм Кнута - Морируса – Пратта. Алгоритм Бойера – Мура. Алгоритм Бойера - Мура на практике
Суффиксные деревья. Поиск по суффиксному дереву. Построение суффиксного дерева. Скорость алгоритмов поиска текста	Суффиксные деревья. Поиск по суффиксному дереву. Построение суффиксного дерева. Скорость алгоритмов поиска текста
<b>Раздел 11. Структуры данных</b>	
Структуры данных: линейные, иерархические и табличные	Линейные: массив; список; связанный список; стек; очередь; хэш-таблица. Иерархические: двоичные деревья; n-арные деревья; иерархический список. Сетевые: неориентированный граф; ориентированный граф. Табличные: таблица реляционной базы данных; двумерный массив. Куча/Пирамида (Heap). Индуктивные функции. Однопроходные алгоритмы. Асимптотика алгоритмов
<b>Раздел 12. Алгоритмы на графах</b>	
Алгоритмы поиска вершин графа, сортировки, алгоритмы поиска кратчайшего пути	Обход графа в глубину (DFS). Выделение и подсчёт компонент связности. Проверка графа на двудольность. Выделение компонент сильной

	связности орграфа алгоритмом Косарайю. Топологическая сортировка через алгоритм Тарьяна. Обход графа в ширину. Алгоритм Дейкстры с очередью. Алгоритмы Флойда-Уоршелла. Двоичные деревья поиска. Асимптотика основных операций. Балансировка деревьев. Малый левый и правый повороты. Большой левый и правый повороты. . Сортировка кучей. Хранение графа в памяти
Алгоритм для нахождения минимального остовного дерева для связного взвешенного графа, Асимптотика основных операций. Балансировка деревьев. Хранение графа в памяти	Алгоритм для нахождения минимального остовного дерева для связного взвешенного графа, Асимптотика основных операций. Балансировка деревьев. Хранение графа в памяти
<b>Раздел 13. Алгоритмы сжатия</b>	
Алгоритмы сжатия	Введение в сжатие. Кодирование длин серий. Код Хаффмана. Сжатие Лемпеля - Зива – Велча. Сжатие с потерями

## 6. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 6.1. Форма обучения – очная, курс – 1, семестр – 2

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
<b>Раздел 1. Основы теории множеств.</b>	2	2	-	7	11
Множества и операции над ними. Простейшие теоретико-множественные тождества	2	2	-	7	11
<b>Раздел 3. Логика высказываний</b>	4	4	-	7	15
Понятие формулы. Таблицы истинности. Эквивалентность формул. Основные тождества логики высказываний и теории множеств.	2	2	-	3	7
Нормальные формы. Приведение формулы к СДНФ и СКНФ	2	2	-	4	8
<b>Раздел 4. Логика предикатов</b>	4	4	-	7	15
Понятие алгебраической системы, алгебры, модели. Примеры. Термы и формулы логики предикатов. Истинность формулы на модели. Тождественно истинные и выполнимые формулы.	2	2	-	3	7
Семантическая эквивалентность формул. Основные тождества логики предикатов	2	2	-	4	8
<b>Раздел 5. Отношения и функции</b>	2	2	-	7	11
Предпорядок, отношения эквивалентности и частичного порядка. Эквивалентность и разбиение, фактор-множество. Максимальные и минимальные, наибольшие и наименьшие	2	2	-	7	11

элементы, точная верхняя и нижняя грани. Понятие решетки					
<b>Раздел 6. Булевы алгебры</b>	2	4	-	7	13
Множество-степень, понятие и основные свойства булевой алгебры Примеры. Атомные и безатомные элементы булевых алгебр. Конечные булевы алгебры, теорема Стоуна для конечных булевых алгебр	2	4	-	7	13
<b>Раздел 7. Мощность множества</b>	2	2	-	7	11
Понятие равномощности множеств. Теорема Кантора-Бернштейна. Теорема Кантора. Бесконечность класса бесконечных мощностей. Парадоксы теории множеств	2	2	-	7	11
<b>Раздел 8. Счетные и континуальные множества</b>	2	2	-	7	11
Счётные множества. Счётность множества слов в счётном алфавите. Континуум. Несчётность множества вещественных чисел. Равномощность множества вещественных чисел и множества всех подмножеств множества натуральных чисел. Континуум-гипотеза и обобщённая континуум-гипотеза. Ординальные и кардинальные числа	2	2	-	7	11
<b>Раздел 9. Машины Тьюринга</b>	4	4	-	7	15
Понятие алгоритма. Вычислимые функции. Машины Тьюринга. Операторы над машинами Тьюринга. Функции, вычислимые на машинах Тьюринг	4	4	-	7	15
<b>Раздел 10. Частично рекурсивные функции</b>	2	2	-	7	11
Операторы суперпозиции, примитивной рекурсии и минимизации. Примитивно-рекурсивные, общерекурсивные и частично рекурсивные функции. Канторовская нумерация	2	2	-	7	11
<b>Раздел 11. Секвенциальное исчисление высказываний.</b>	2	2	-	7	11
Секвенциальное исчисление высказываний: Аксиомы и правила вывода. Допустимые правила вывода. Теорема о корректности. Теорема о подстановке. Теорема о замене. Теорема о существовании КНФ. Теорема о полноте исчисления секвенций	2	2	-	7	11
<b>Раздел 12. Исчисление высказываний гильбертовского типа</b>	2	2	-	7	11
Аксиомы и правила вывода. Теорема об эквивалентности исчисления секвенций и	2	2	-	7	11

исчисления высказываний гильбертовского типа. Теорема о дедукции					
<b>ИТОГО ЗА СЕМЕСТР</b>	<b>30</b>	<b>30</b>	<b>-</b>	<b>84</b>	<b>144</b>

6.2. Форма обучения – очная, курс – 2, семестр – 3

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС	Всего
<b>Раздел 1. Базовые конструкции языка Python</b>	1	1	-	4	6
Компилируемые и интерпретируемые языки программирования. Интерпретатор Python. Стандарт кода PEP8. Переменные, оператор присваивания, функции type и id. Числовые типы, арифметические операции. Особенности округления при целочисленном делении в языках C++ и Python	1	1	-	4	6
<b>Раздел 2. Условные операторы, циклы, генераторы списков. Обработка исключений. Инструкции assert в Python</b>	3	3	-	6	12
Условный оператор, логические операции, тернарный условный оператор, вложенные и каскадные условия	1	1	-	3	5
Операторы циклов, генераторы списков (List comprehensions). вложенные генераторы списков. Обработка исключений. Инструкции assert в Python	2	2	-	3	7
<b>Раздел 3. Последовательности в Python.</b>	3	3	-	6	12
Строки. Методы и функции. Срезы (slices). Форматирование строк. Метод format, f-строка. Оператор '%' при выводе значений	1	1	-	0	2
Списки, кортежи, двумерные массивы	1	1	-	3	5
Словари, множества (set и frozenset)	1	1	-	3	5
<b>Раздел 4. Функции в python: синтаксис, логика и применение. Элементы функционального программирования</b>	4	4	-	6	14
Функции, параметры и аргументы. Операторы return, yield в функциях. Области видимости переменных. Ключевые слова global и nonlocal. Замыкания в Python.	2	2	-	3	7
Анонимные (lambda) функции. Функции map, filter, reduce, zip. Введение в декораторы функций. Декораторы с параметрами	2	2	-	3	7
<b>Раздел 5. Рекурсивные алгоритмы</b>	1	1	-	6	8
Рекурсивные алгоритмы, Сочетания. Размещения. Оптимизация рекурсии. Кривая Коха	1	1	-	6	8

<b>Раздел 6. Модули и пакеты в Python</b>	1	1	-	6	8
Основные стандартные модули и пакеты в Python. Импортирование модулей. Создание собственных модулей и их импортирование. Специализированные модули и приложения	1	1	-	6	8
<b>Раздел 7. Работа с файлами, файловой системой и процессами в Python</b>	4	4	-	6	14
Основные возможности Python по работе с файлами и файловой системой. Модули os, shutil, pathlib, glob. Работа с датой и временем	2	2	-	3	7
Аргументы командной строки, модуль argparse. Сериализация pickle, json, yaml. Работа с внешними процессами. Модуль subprocess	2	2	-	3	7
<b>Раздел 8. Алгоритмы поиска и сортировки</b>	3	3	-	6	12
Линейный поиск. Сложность алгоритмов. Нотации O, Ω, Θ. Бинарный поиск	1	1	-	2	4
Квадратичные сортировки: Сортировка выбором. Сортировка методом пузырька. Сортировка вставками. Сортировка подсчетом. Поразрядная сортировка (Radix Sort). Синхронная сортировка массивов	1	1	-	2	4
Быстрые сортировки. Быстрая сортировка Сортировка слиянием. Устойчивость сортировок.	1	1	-	2	4
<b>Раздел 9. Перебор. Жадные алгоритмы. Алгоритмы динамического программирования</b>	4	4	-	6	14
Жадные алгоритмы	1	1	-	2	4
Задачи одномерного динамического программирования	1	1	-	2	4
Задачи двумерного динамического программирования	2	2	-	2	6
<b>Раздел 10. Строковые алгоритмы</b>	3	4	-	6	13
Простейший поиск подстроки. Суффиксы, префиксы, грани. Префикс-функция. Алгоритм Кнута - Морриса – Пратта. Алгоритм Бойера – Мура. Алгоритм Бойера - Мура на практике	2	2	-	3	7
Суффиксные деревья. Поиск по суффиксному дереву. Построение суффиксного дерева. Скорость алгоритмов поиска текста	1	2	-	3	6
<b>Раздел 11. Структуры данных</b>	1	1	-	6	8
Структуры данных: линейные, иерархические и табличные	1	1	-	6	8
<b>Раздел 12. Алгоритмы на графах</b>	5	4	-	6	15

Алгоритмы поиска вершин графа, сортировки, алгоритмы поиска кратчайшего пути	2	1	-	3	6
Алгоритм для нахождения минимального остовного дерева для связного взвешенного графа, Асимптотика основных операций. Балансировка деревьев. Хранение графа в памяти	3	3	-	3	9
<b>Раздел 13. Алгоритмы сжатия</b>	1	1	-	6	8
Введение в сжатие. Кодирование длин серий. Код Хаффмана. Сжатие Лемпеля - Зива – Велча. Сжатие с потерями	1	1	-	6	8
<b>ИТОГО ЗА СЕМЕСТР</b>	34	34	-	76	144
<b>ИТОГО ПО КОМПОНЕНТУ ОПОП</b>	64	64	-	160	288

## 7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### Семестр 2

#### 7.1. Контрольные вопросы

##### Раздел 1.

Множества и операции над ними. Простейшие теоретико-множественные тождества

##### Раздел 2.

Язык логики высказываний. Понятие формулы. Таблицы истинности. Эквивалентность формул. Связь теоретико-множественных тождеств и тождеств логики высказываний. Основные тождества логики высказываний и теории множеств. Нормальные формы. Приведение формулы к СДНФ и СКНФ.

##### Раздел 3.

Понятие алгебраической системы, алгебры, модели. Примеры. Термы и формулы логики предикатов. Истинность формулы на модели. Тождественно истинные и выполнимые формулы. Семантическая эквивалентность формул. Основные тождества логики предикатов.

##### Раздел 4.

Предпорядок, отношения эквивалентности и частичного порядка. Эквивалентность и разбиение, фактор-множество. Максимальные и минимальные, наибольшие и наименьшие элементы, точная верхняя и нижняя грани. Понятие решетки.

##### Раздел 5.

Множество-степень, понятие и основные свойства булевой алгебры. Примеры. Атомные и безатомные элементы булевых алгебр. Конечные булевы алгебры, теорема Стоуна для конечных булевых алгебр.

##### Раздел 6.

Понятие равномощности множеств. Теорема Кантора-Бернштейна. Теорема Кантора. Бесконечность класса бесконечных мощностей. Парадоксы теории множеств

##### Раздел 7.

Счётные множества. Счётность множества слов в счётном алфавите. Континуум. Несчётность множества вещественных чисел. Равномощность множества вещественных чисел и множества всех подмножеств множества натуральных чисел. Континуум-гипотеза и обобщённая континуум-гипотеза. Ординальные и кардинальные числа

##### Раздел 8.

Понятие алгоритма. Вычислимые функции. Машины Тьюринга. Операторы над машинами Тьюринга. Функции, вычислимые на машинах Тьюринг

##### Раздел 9.

Операторы суперпозиции, примитивной рекурсии и минимизации. Примитивно-рекурсивные, общерекурсивные и частично-рекурсивные функции. Канторовская нумерация

#### Раздел 10.

Секвенциальное исчисление высказываний: Аксиомы и правила вывода. Допустимые правила вывода. Теорема о корректности. Теорема о подстановке. Теорема о замене. Теорема о существовании КНФ. Теорема о полноте исчисления секвенций.

#### Раздел 11.

Аксиомы и правила вывода. Теорема об эквивалентности исчисления секвенций и исчисления высказываний гильбертовского типа. Теорема о дедукции

### Семестр 2

#### Раздел 1.

Компилируемые и интерпретируемые языки программирования. Интерпретатор Python. Стандарт кода PEP8. Переменные, оператор присваивания, функции type и id. Числовые типы, арифметические операции. Функции print() и input(). Преобразование строк в числа int() и float(). Динамическая типизация. Изменяемые и неизменяемые типы данных в Python. Целочисленная арифметика. Особенности округления при целочисленном делении в языках C++ и Python. Логический тип bool. Операторы сравнения и операторы and, or, not. Математические функции и работа с модулем math. Обработка исключений. Инструкции assert в Python

#### Раздел 2.

Условный оператор if. Конструкция if-else. Вложенные условия и множественный выбор. Конструкция if-elif-else. Тернарный условный оператор. Вложенное тернарное условие. Оператор цикла while. Операторы циклов break, continue и else. Оператор цикла for. Функция range(). Примеры работы оператора цикла for. Функция enumerate(). Итератор и итерируемые объекты. Функции iter() и next(). Вложенные циклы. Примеры задач с вложенными циклами. Треугольник Паскаля как пример работы вложенных циклов. Генераторы списков (List comprehensions). Вложенные генераторы списков.

#### Раздел 3.

Строки. Методы и функции. Срезы (slices). Форматирование строк. Метод format, f-строка. Оператор '%' при выводе значений. Списки, кортежи словари, множества (set и frozenset). Двумерные массивы.

#### Раздел 4.

Функции: определение def и их вызов. Оператор return в функциях. Функциональное программирование. Именованные аргументы. Фактические и формальные параметры. Функции с произвольным числом параметров \*args и \*\*kwargs. Операторы \* и \*\* для упаковки и распаковки коллекций. Рекурсивные функции. Анонимные (lambda) функции. Функции map, filter, reduce, zip.. Области видимости переменных. Ключевые слова global и nonlocal. Замыкания в Python. Введение в декораторы функций. Декораторы с параметрами. Сохранение свойств декорируемых функций

#### Раздел 5.

Факториал. Числа Фибоначчи. Ханойская башня. Задача о 8 ферзях. Оптимизация задачи о восьми ферзях. Сочетания. Размещения. Оптимизация рекурсии. Кривая Коха

#### Раздел 6.

Основные стандартные модули и пакеты в Python. Импорт модулей. Модули itertools и functools. Создание собственных модулей и их импорт. Специализированные модули и приложения

#### Раздел 7.

Основные возможности Python по работе с файлами и файловой системой. Модули os, shutil, pathlib, glob. Работа с датой и временем. Аргументы командной строки, модуль

argparse. Сериализация pickle, json, yaml. Работа с внешними процессами. Модуль subprocess

#### **Раздел 8.**

Сложность алгоритмов. Натация O-большое. Квадратичные сортировки. Сортировка вставками. Сортировка выбором. Сортировка методом пузырька. Сортировка подсчётом. Поразрядная сортировка. Алгоритм Евклида. Быстрое возведение в степень. Ханойские башни. Быстрые сортировки: Тони Хоара и слиянием. Бинарный поиск.

#### **Раздел 9.**

Поиск определённого числа в потоке чисел. Поиск числа в потоке данных. Поиск максимального и минимального числа в потоке. Поиск максимального числа в потоке. Поиск второго по значению экстремума. Бинарный поиск в массиве на Python (два фиктивных элемента). Квадратичные сортировки: Сортировка выбором. Сортировка методом пузырька. Квадратичные сортировки в Python. Сортировка вставками. Быстрые сортировки. Быстрая сортировка Хоара. Быстрая сортировка Хоара: Python. Быстрая сортировка Хоара: Pascal. Сортировка слиянием. Сортировка подсчётом. Синхронная сортировка массивов. Устойчивость сортировок. Поразрядная сортировка. Блинная сортировка

#### **Раздел 10.**

Числа Фибоначчи. Самая длинная общая подпоследовательность. Задача о рюкзаке. Расстояние Левенштейна (редакционное расстояние). Самый большой подмассив. Размен монет. Самая длинная возрастающая подпоследовательность. Умножение цепочки матриц. Задача коммивояжера. 0-1 Целочисленное программирование. Расстояние Левенштейна с разрешенными операциями. Самая длинная палиндромная подстрока. Задача о подмассиве максимального произведения (Maximum Product Subarray). Самый большой прямоугольник на гистограмме. Бросание яиц. Подсчет битов. Идеальные квадраты. Раздел равной суммы подмножества (Partition Equal Subset Sum). Самая длинная общая подстрока. Уникальные пути. Расстояние Левенштейна с помощью разрешенных операций. Проблема суммы подмножества. Самая длинная палиндромная последовательность. Самый большой прямоугольник на гистограмме. Уникальные пути Проверка равенства строк.

#### **Раздел 11.**

Простейший поиск подстроки. Суффиксы, префиксы, грани. Префикс-функция. Алгоритм Кнута - Морриса - Пратта. Алгоритм Бойера - Мура. Алгоритм Бойера - Мура на практике. Суффиксные деревья. Поиск по суффиксному дереву. Построение суффиксного дерева. Скорость алгоритмов поиска текста.

#### **Раздел 12.**

Линейные: массив; список; связанный список; стек; очередь; хэш-таблица. Иерархические: двоичные деревья; n-арные деревья; иерархический список. Сетевые: неориентированный граф; ориентированный граф. Табличные: таблица реляционной базы данных; двумерный массив. Куча/Пирамида (Heap). Индуктивные функции. Однопроходные алгоритмы. Асимптотика алгоритмов

#### **Раздел 13.**

Обход графа в глубину (DFS). Выделение и подсчёт компонент связности. Проверка графа на двудольность. Выделение компонент сильной связности орграфа алгоритмом Косарайо. Топологическая сортировка через алгоритм Тарьяна. Обход графа в ширину. Алгоритм Дейкстры с очередью. Алгоритмы Флойда-Уоршелла. Двоичные деревья поиска. Асимптотика основных операций. Балансировка деревьев. Малый левый и правый повороты. Большой левый и правый повороты. . Сортировка кучей. Хранение графа в памяти.

#### **Раздел 14.**

Введение в сжатие. Кодирование длин серий. Код Хаффмана. Сжатие Лемпеля - Зива - Велча. Сжатие с потерями

## 7.2. Темы письменных работ (типы задач)

Контрольные работы по темам:

### Семестр 1

Контрольная работа №1

1. Составить таблицы истинности для функции вручную и с помощью языка программирования

$$F = A \vee \bar{B} \vee (\bar{A} \vee C)$$

2.

a) Упростить выражение:  $X \wedge Y \wedge Z \vee \bar{X} \wedge Y \wedge Z$

b) Упростить выражение:  $(X \vee Y) \wedge (X \vee Y \vee Z) \wedge (X \vee Y \vee \bar{Z})$

c) Упростить. Построить таблицу истинности для получения логического выражения:

$$X = (B \rightarrow A) \cdot \overline{(\bar{B} \cdot \bar{C})}$$

3. Найти значение выражения:

$$(A \rightarrow B) \vee (B \rightarrow A)$$

Контрольная работа №2

Записать формулу функции  $f$  в виде СДНФ и минимизировать ее графическим методом, методом Квайна, Квайна–Мак-Класки и методом минимизирующих карт Карно (диаграмм Вейча). Сравнить полученные результаты.

$$f = (1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0)$$

Контрольная работа №3

Получить полином Жегалкина:

$$a) f(x_1, x_2, x_3) = (\bar{x}_1 \wedge x_2 \wedge \bar{x}_3) \vee (x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$

$$b) f(x_1, x_2, x_3) = \bar{x}_2 \vee ((x_1 \wedge \bar{x}_3) | (\bar{x}_2 | \bar{x}_3))$$

### Семестр 2

- Условные операторы, циклы, генераторы списков

- Последовательности в Python

Контрольная работа №1

1. Вам дается последовательность целых чисел и вам нужно ее обработать и вывести на экран сумму первой пятерки чисел из этой последовательности, затем сумму второй пятерки, и т. д. Но последовательность не дается вам сразу целиком. С течением времени к вам поступают её последовательные части. Например, сначала первые три элемента, потом следующие шесть, потом следующие два и т. д.
2. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу.
3. По данной строке определите, является ли она палиндромом (то есть, читается одинаково как слева-направо, так и справа-налево).

Контрольная работа по проверке теоретических знаний – по всем темам, с использованием указанных выше контрольных вопросов.

- Функции в python: синтаксис, логика и применение. Элементы функционального программирования

#### Контрольная работа №2

1. Есть список  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ . Выведите все элементы, которые меньше 5.
2. Даны списки:  
 $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ ;  
 $b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$ .  
 Нужно вернуть список, который состоит из элементов, общих для этих двух списков.
3. С помощью анонимной функции извлеките из списка числа, делимые на 15  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ .
4. . Перемножить все числа в списке  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ .
5. Вычисление наибольшего элемента в списке  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ .
6. Вычисление суммы всех элементов списка  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ .  
 Напишите программу, которая принимает два списка и выводит все элементы первого, которых нет во втором  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ ;  
 $b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$ .
7. Посчитайте факториал с помощью lambda-выражения и reduce  
 $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$ ;  
 $b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$ .

#### Контрольная работа №3

1. Напишите программу, которая считывает из файла строку, соответствующую тексту, сжатому с помощью кодирования повторов, и производит обратную операцию, получая исходный текст.
2. Напишите программу, которая считывает текст из файла (в файле может быть больше одной строки) и выводит самое частое слово в этом тексте и через пробел то, сколько раз оно встретилось. Если таких слов несколько, вывести лексикографически первое.
3. Имеется файл с данными по успеваемости абитуриентов. Он представляет из себя набор строк, где в каждой строке записана следующая информация: Фамилия; Оценка по математике; Оценка по физике; Оценка по русскому языку. Поля внутри строки разделены точкой с запятой, оценки - целые числа. Напишите программу, которая считывает исходный файл с подобной структурой и для каждого абитуриента записывает его среднюю оценку по трём предметам на отдельной строке, соответствующей этому абитуриенту, в файл с ответом. Также вычислите средние баллы по математике, физике и русскому языку по всем абитуриентам и добавьте полученные значения, разделённые пробелом, последней строкой в файл с ответом.

#### Контрольная работа №4

1. Сгенерировать все правильные скобочные последовательности со скобками одного вида (что такое правильная скобочная последовательность), где количество скобок равно  $k$ .

2. Дан упорядоченный по возрастанию массив с числами от 0 до  $n-1$ , требуется сгенерировать все его подмножества.
3. Дан упорядоченный по возрастанию массив с числами от 0 до  $n-1$ , требуется сгенерировать все его перестановки.

7.3. Образец содержания экзаменационного билета (при наличии экзамена по дисциплине)

В случае ведения учебного процесса с использованием электронного обучения и дистанционных образовательных технологий, содержание билета может отличаться от приведенного.

### ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

**Донецкий национальный университет**  
**Физико-технический факультет**  
**Кафедра радиофизики и ИКТ**

Образовательно-квалификационный уровень: бакалавр  
 Направление подготовки: 10.03.01  
 Специальность: информационная безопасность Семестр: 3  
 Научная дисциплина: Технологии и методы программирования

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №7

1. Условные конструкции if / elif / else
2. Пользовательские функции. Области видимости.
3. Квадратичные сортировки (сортировка вставками, выбором, методом пузырька).
4. Даны целые числа  $a_1, a_2, \dots, a_n$  (в этой последовательности могут быть повторяющиеся числа). Вывести на печать все числа, которые входят в последовательность по одному разу.

Утверждено на заседании кафедры  
 Протокол \_\_\_\_ от \_\_\_\_\_

Заведующий кафедры радиофизики и ИКТ \_\_\_\_\_ Данилов В.В.  
 (подпись) (фамилия и инициалы)

Экзаменатор \_\_\_\_\_ Кожекина Е. Н.  
 (подпись) (фамилия и инициалы)

### 8. РАСПРЕДЕЛЕНИЕ БАЛЛОВ, КОТОРЫЕ ПОЛУЧАЮТ ОБУЧАЮЩИЕСЯ

Общая оценка знаний, обучающихся по дисциплине проводится по 100-балльной шкале исходя из максимума, приведенного в таблице ниже. Организационно-учебная работа в аудитории оценивается на основе таких критериев как посещаемость занятий, своевременное и качественное выполнение домашних заданий, активность во время проведения лекционных и практических занятий (участие в обсуждении текущего и пройденного материала, решение задач и т.п.).

#### 8.1. Семестр 2

Номера разделов	Виды работ	Максимальное количество баллов
-----------------	------------	--------------------------------

1-3	Организационно-учебная работа в аудитории	2
	Самостоятельная работа	3
	Контрольные работы	5
	Лабораторные работы	5
4-7	Организационно-учебная работа в аудитории	2
	Самостоятельная работа	3
	Контрольные работы	5
	Лабораторные работы	5
8-11	Организационно-учебная работа в аудитории	2
	Самостоятельная работа	3
	Контрольные работы	5
	Лабораторные работы	5
ИТОГО		45
ЗАЧЕТ		55
Общий итог за семестр		100

## 8.1. Семестр 3

1-3	Организационно-учебная работа в аудитории	1
	Самостоятельная работа	2
	Контрольные работы	4
	Лабораторные работы	5
4-7	Организационно-учебная работа в аудитории	1
	Самостоятельная работа	2
	Контрольные работы	4
	Лабораторные работы	5
8-10	Организационно-учебная работа в аудитории	1
	Самостоятельная работа	2
	Контрольные работы	4
	Лабораторные работы	5
11-13	Организационно-учебная работа в аудитории	1
	Самостоятельная работа	2
	Контрольные работы	5
	Лабораторные работы	6
ИТОГО		50
Экзамен		50
Общий итог за семестр		100

## Соответствие баллов оценке

Количество баллов из 100	ECTS	Оценка по пятибалльной шкале	
		Экзамен, дифференцированный зачет	Зачет
90-100	A	отлично	зачтено
80-89	B	хорошо	зачтено
75-79	C		зачтено
70-74	D	удовлетворительно	зачтено
60-69	E		зачтено
35-59	FX	неудовлетворительно	не зачтено
0-34	F		не зачтено

## 9. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- 1) для слепых и слабовидящих:
  - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
  - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
  - письменные задания оформляются увеличенным шрифтом.
- 2) для глухих и слабослышащих:
  - лекции оформляются в виде электронного документа;
  - письменные задания выполняются на компьютере в письменной форме;
  - экзамен проводится в письменной форме на компьютере; возможно проведение в форме тестирования.
- 3) для лиц с нарушениями опорно-двигательного аппарата:
  - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
  - письменные задания выполняются на компьютере;
  - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- 1) для слепых и слабовидящих:
  - в печатной форме увеличенным шрифтом;
  - в форме электронного документа;
- 2) для глухих и слабослышащих:
  - в печатной форме;
  - в форме электронного документа.
- 3) для обучающихся с нарушениями опорно-двигательного аппарата:
  - в печатной форме;
  - в форме электронного документа.

## 10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Учебные занятия проводятся в корпусе №4 ДонГУ (г. Донецк, пр. Театральный, 13). Для проведения лекционных и практических занятий требуется аудитория, оборудованная меловой или маркерной доской, мультимедийный проектор и экран, ноутбук, комплект учебной мебели для студентов, рабочее место преподавателя, выход в Интернет – проводной или с использованием Wi-Fi.

Для проведения лабораторных занятий требуется оборудованная персональными компьютерами аудитория.

Для самостоятельной работы используются текстовые и электронные ресурсы Научной библиотеки университета и других электронных библиотечных баз данных, учебно-методическое обеспечение, представленное в учебно-методическом кабинете Главного корпуса (ауд.405).

Обучающиеся имеют возможность использовать учебные материалы по дисциплине, размещенные на платформе Moodle Центра дистанционного образования ФГБОУ ВО «ДонГУ». При изучении дисциплины применяются электронное обучение и дистанционные образовательные технологии.

С использованием ресурсов платформы дистанционного образования осуществляется текущий контроль знаний, обучающихся на основе тестирования и проверки результатов самостоятельной работы.

## 11. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

### 11.1. Основная литература

1. Светозарова, Г. И. Современные методы программирования в примерах и задачах / Г. И. Светозарова, А. В. Козловский, Е. В. Сигитов ; Под ред. С. В. Емельяновой. - М. : Наука : Физматлит, 1995. - 432 с.
2. Заварыкин, В. М. Техника вычислений и алгоритмизация: Учеб. пособие для пед. ин-тов по физ. мат. спец. / Заварыкин, В. М., Лапчик, М. П., Житомирский, В. Г. - М.: Просвещение, 1987. - 160 с.
3. Кнут, Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 3 : Сортировка и поиск / Пер. с англ. Н. И. Вьюковой и др. ; Под ред. Ю. М. Баяковского. - М. : Мир, 1978. - 844 с.

## 12. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Интерактивный учебник языка Python. [Электронный ресурс]. – URL: [http://pythontutor.ru/lessons/inout\\_and\\_arithmetic\\_operations/](http://pythontutor.ru/lessons/inout_and_arithmetic_operations/)
2. Самоучитель Python. [Электронный ресурс]. – URL: <https://pythonworld.ru/samouchitel-python>
3. Программирование на Python. [Электронный ресурс]. – URL: [https://www.ibm.com/developerworks/ru/library/l-python\\_part\\_1/#ibm-pcon](https://www.ibm.com/developerworks/ru/library/l-python_part_1/#ibm-pcon)
4. "Поколение Python": курс для начинающих [Электронный ресурс]. – URL: <https://stepik.org/course/58852/syllabus>
5. "Поколение Python": курс для продвинутых [Электронный ресурс]. – URL: <https://stepik.org/course/68343/syllabus>
6. Алгоритмы и структуры данных [Электронный ресурс]. – URL: <https://stepik.org/course/180830/syllabus>
7. Алгоритмы: теория и практика. Методы [Электронный ресурс]. – URL: <https://stepik.org/course/217/syllabus>

## 13. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Python 3.13.0
2. Windows 7 PRO (корпоративная лицензия ДонГУ № 46484614)
2. Microsoft Office (корпоративная лицензия ДонГУ № 46472919)
3. Microsoft Visual Studio (лицензия программы Dream Spark для высших учебных заведений)
4. Антивирус Касперского, Adobe Acrobat Reader, xPDF (лицензии GPL, Apache, BSD для свободного программного обеспечения).