

Министерство науки и высшего образования  
Российской Федерации

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

Физико-технический факультет  
Кафедра компьютерных технологий



П.А. Машаров

2024 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Укрупненная группа направлений подготовки	44.00.00 Образование и педагогические науки
Программа высшего образования	Программа бакалавриата
Направление подготовки	44.03.05 Педагогическое образование (с двумя профилями подготовки)
Профиль подготовки	Математика и информатика
Квалификация	Бакалавр
Форма обучения	Очная, заочная

Рабочая программа адаптирована для лиц  
с ограниченными возможностями здоровья и инвалидов

Донецк 2024

Рабочая программа дисциплины «Алгоритмизация и программирование» для обучающихся по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки) (Профиль: Математика и информатика), составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), утвержденного приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 125 (с изм. и доп.), Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры 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.

Разработчики:

Старший преподаватель  
компьютерных технологий

кафедры



В.Е. Бодряга

Старший преподаватель  
компьютерных технологий

кафедры



А.Е. Гукай

Рабочая программа одобрена на заседании кафедры компьютерных технологий  
Протокол от 26.03.2024 г. № 12

Заведующий кафедрой



Г.В. Аверин

СОГЛАСОВАНО:

Декан факультета математики и  
информационных технологий  
28.03.2024 г.



И.А. Моисеенко

Учебно-методическая комиссия факультета математики и информационных технологий.  
Протокол от 27.03.2024 г. № 3

Председатель



Л. И. Селякова

Руководитель основной профессиональной  
образовательной программы,  
д-р пед. наук, проф., зав. каф. ВМиМППМ  
26.03.2024 г.



Е.И. Скафа

## 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 и информатике в объеме программы средней школы;

дисциплины программы бакалавриата: Программное обеспечение ЭВМ, Математический анализ, Решение задач школьного курса информатики и ИКТ, Алгебра.

1.2. Дисциплины, курсовые работы и практики, для которых освоение данной дисциплины необходимо как предшествующее:

Научный семинар по вопросам обучения алгоритмизации и программированию в школе, Производственная практика: научно-исследовательская работа (обязательная), Производственная практика: преддипломная практика (обязательная).

## 2. ОПИСАНИЕ ДИСЦИПЛИНЫ

### 2.1. Общая характеристика

Наименование показателя	Значение показателя
Название образовательной программы	44.03.05 Педагогическое образование (с двумя профилями подготовки) (Профиль: Математика и информатика)
Шифр и название в соответствии с учебным планом	Б1.Б.М8 Алгоритмизация и программирование
Часть образовательной программы	Базовая часть
Количество зачетных единиц / всего часов	7 / 245

### 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	17	34	–	57	108	зачет
	2	3	34	34	–	76	144	экзамен
Очная, всего			51	68	–	133	252	
Заочная	1	2	4	6	–	98	108	зачет
	2	3	6	6	–	132	144	экзамен
Заочная, всего			10	12	–	230	252	

## 3. ЦЕЛИ ДИСЦИПЛИНЫ

Формирование представлений студента о фундаментальных понятиях алгоритмизации, парадигмах и стиле программирования, базовых навыков программирования.

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

##### 4.1. Компетенции

Компетенции	Индикаторы	Результаты обучения
ОПК-8. Способен осуществлять педагогическую деятельность на основе специальных научных знаний	ОПК-8.11. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.11.1 <b>Знать:</b> алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ОПК-8.11.2 <b>Уметь:</b> составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули ОПК-8.11.3 <b>Владеть:</b> языком программирования; навыками отладки и тестирования работоспособности программы

#### 5. ПРОГРАММА ДИСЦИПЛИНЫ

Название темы	Краткое содержание темы (вопросы темы)
Раздел 1. Алгоритм. Интегрированная среда разработки	
Основы алгоритмизации. Система и язык программирования в средней и высшей школе КуМир	<ol style="list-style-type: none"> <li>1. Понятие алгоритма и его свойства.</li> <li>2. Методы разработки алгоритмов, виды записи. Блок-схемы.</li> <li>3. Эволюция и классификация языков программирования.</li> <li>4. Алгоритмы и исполнители в системе программирования КуМир.</li> </ol>
Интегрированная среда разработки Python IDLE (Integrated Development and Learning Environment).	<ol style="list-style-type: none"> <li>1. Обзор возможностей интегрированных сред.</li> <li>2. Написание, запуск, отладка и корректировка программы.</li> </ol>
Раздел 2. Основы языка высокого уровня Python	
Основные понятия языка высокого уровня Python	<ol style="list-style-type: none"> <li>1. Программа, порядок ее разработки и исполнения.</li> <li>2. Язык Python: алфавит, синтаксис, семантика.</li> <li>3. Концепция типа данных. Линейные программы</li> </ol>

Структурное программирование	<ol style="list-style-type: none"> <li>1. Базовые конструкции структурного программирования и их реализация в виде управляющих конструкций языка Python.</li> <li>2. Встроенные структуры: словари, кортежи, множества.</li> <li>3. Программирование условий: условный оператор, оператор выбора.</li> <li>4. Программирование циклов.</li> <li>5. Средства организации модульности в языках высокого уровня</li> </ol>
Структуры и типы данных. Парадигмы и технологии программирования	<ol style="list-style-type: none"> <li>1. Абстрактные типы данных: стек, линейный список.</li> <li>2. Реализация динамических структур средствами языка Python.</li> <li>3. Парадигмы и технологии программирования</li> </ol>
Раздел 3. Базовые алгоритмические конструкции	
Введение в алгоритмизацию и программирование на с#	<ol style="list-style-type: none"> <li>1. Введение в алгоритмизацию и программирование.</li> <li>2. Алгоритм, свойства, виды представления.</li> <li>3. Основные парадигмы современного программирования</li> </ol>
Интегрированная среда разработки MS Visual Studio.	<ol style="list-style-type: none"> <li>1. Интегрированная среда разработки MS Visual Studio.</li> <li>2. Редактор кода, отладчик. Знакомство с .NET.</li> </ol>
Линейный вычислительный процесс	<ol style="list-style-type: none"> <li>1. Структура программы на языке С#.</li> <li>2. Простые типы данных С#. Переменные, константы, идентификаторы, операторы, выражения.</li> <li>3. Работа с консолью.</li> </ol>
Разветвляющийся и циклический вычислительный процесс	<ol style="list-style-type: none"> <li>1. Разветвляющийся и циклический вычислительные процессы.</li> <li>2. Конструкции языка С#: if, switch, for, while, break, continue.</li> </ol>
Раздел 4. Пользовательские функции и типы данных	
Массивы	<ol style="list-style-type: none"> <li>1. Одномерные и двумерные массивы. Свойства массивов.</li> <li>2. Ступенчатые и прямоугольные массивы.</li> <li>3. Генерация случайных чисел.</li> </ol>
Пользовательские функции. Рекурсия	<ol style="list-style-type: none"> <li>1. Пользовательские функции.</li> <li>2. Передача параметров: модификаторы ref, out.</li> <li>3. Ключевое слово params. Рекурсивные функции</li> </ol>
Абстрактные синтаксические деревья. РБНФ-выражения	<ol style="list-style-type: none"> <li>1. Синтаксис, семантика, грамматика.</li> <li>2. Деревья разбора.</li> <li>3. Абстрактные синтаксические деревья. РБНФ-выражения.</li> </ol>

Работа со строками	<ol style="list-style-type: none"> <li>1. Работа со строками.</li> <li>2. Тип String, StringBuilder.</li> <li>3. Введение в регулярные выражения. Основные элементы синтаксиса.</li> </ol>
Перечисляемый тип данных. Структуры	<ol style="list-style-type: none"> <li>1. Перечисляемый тип данных enum.</li> <li>2. Структуры struct. Работа с датой и временем (DateTime, TimeSpan).</li> </ol>
Раздел 5. Базовые алгоритмы и структуры данных	
Работа с файлами	<ol style="list-style-type: none"> <li>1. Работа с двоичными и текстовыми файлами.</li> <li>2. Тип FileStream, BinaryReader, BinaryWriter, StreamReader, StreamWriter.</li> </ol>
Простые алгоритмы сортировки	<ol style="list-style-type: none"> <li>1. Простые алгоритмы сортировки: выбором, вставками, пузырьком, шейкерная, Шелла.</li> <li>2. Введение в анализ сложности алгоритмов.</li> </ol>
Простые алгоритмы поиска	<ol style="list-style-type: none"> <li>1. Алгоритмы поиска: линейный поиск перебором, бинарный, интерполяционный поиск.</li> </ol>
Поиск последовательностей	<ol style="list-style-type: none"> <li>1. Поиск последовательностей: перебор, алгоритм Кнута-Мориса-Пратта, алгоритм Бойера-Мура-Хорспула</li> </ol>
Фундаментальные структуры данных и абстрактные типы данных	<ol style="list-style-type: none"> <li>1. Фундаментальные структуры данных (ФСД) и абстрактные типы данных (АТД).</li> <li>2. Односвязные и двусвязные списки.</li> <li>3. Циклические списки.</li> </ol>
Стек, очередь, дек	<ol style="list-style-type: none"> <li>1. АТД стека, очереди, дека</li> </ol>
Деревья	<ol style="list-style-type: none"> <li>1. Деревья. Обход деревьев.</li> <li>2. Деревья бинарного поиска.</li> <li>3. Сбалансированные деревья. В-деревья.</li> </ol>
Раздел 6. Продвинутое алгоритмы и структуры данных	
Хеш-таблицы	<ol style="list-style-type: none"> <li>1. Хеш-таблицы.</li> <li>2. Хеш-функции</li> </ol>
Коллекции в .NET	<ol style="list-style-type: none"> <li>1. Коллекции в .NET: List, LinkedList, Dictionary, Stack, Queue</li> </ol>
Двоичная куча. Очередь с приоритетами	<ol style="list-style-type: none"> <li>1. Двоичная куча.</li> <li>2. АТД очереди с приоритетами. Пирамидальная сортировка</li> </ol>
Продвинутое алгоритмы сортировки	<ol style="list-style-type: none"> <li>1. Стратегия «Разделяй и властвуй».</li> <li>2. Быстрая сортировка.</li> <li>3. Сортировка слиянием. Сортировка подсчетами</li> </ol>
Графы	<ol style="list-style-type: none"> <li>1. Графы. Представление графа в памяти.</li> <li>2. Поиск в глубину и ширину.</li> <li>3. Поиск кратчайших путей в графе. Минимальные остовные деревья</li> </ol>
Динамическое программирование	<ol style="list-style-type: none"> <li>1. Принципы динамического программирования.</li> <li>2. Задача о рюкзаке.</li> </ol>

## 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
Раздел 1. Алгоритм. Интегрированная среда разработки	<b>4</b>	<b>12</b>		<b>20</b>	<b>36</b>
Основы алгоритмизации	2	6		10	18
Интегрированная среда разработки Python IDLE (Integrated Development and Learning Environment)	2	6		10	18
Раздел 2. Основы языка высокого уровня Python	<b>13</b>	<b>22</b>		<b>37</b>	<b>72</b>
Основные понятия языка высокого уровня Python	4	6		11	21
Структурное программирование	5	8		12	25
Структуры и типы данных. Парадигмы и технологии программирования	4	8		14	26
<b>ИТОГО ЗА СЕМЕСТР</b>	<b>17</b>	<b>34</b>		<b>57</b>	<b>108</b>

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

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
Раздел 3. Базовые алгоритмические конструкции	<b>5</b>	<b>5</b>		<b>14</b>	<b>24</b>
Введение в алгоритмизацию и программирование на с#	1	1		5	7
Интегрированная среда разработки MS Visual Studio.	2	2		3	7
Линейный вычислительный процесс	1	1		3	5
Разветвляющийся и циклический вычислительный процесс	1	1		3	5
Раздел 4. Пользовательские функции и типы данных	<b>10</b>	<b>10</b>		<b>15</b>	<b>35</b>
Массивы	2	2		3	7
Пользовательские функции. Рекурсия	2	2		3	7
Абстрактные синтаксические деревья. РБНФ-выражения	2	2		3	7
Работа со строками	2	2		3	7
Перечисляемый тип данных. Структуры	2	2		3	7
Раздел 5. Базовые алгоритмы и структуры данных	<b>13</b>	<b>13</b>		<b>21</b>	<b>47</b>
Работа с файлами	2	2		3	7
Простые алгоритмы сортировки	2	2		3	7
Простые алгоритмы поиска	2	2		3	7
Поиск последовательностей	2	2		3	7
Фундаментальные структуры данных и абстрактные типы данных	2	2		3	7
Стек, очередь, дек	1	1		3	5
Деревья	2	2		3	7

Раздел 6. Продвинутое алгоритмы и структуры данных	<b>6</b>	<b>6</b>		<b>26</b>	<b>38</b>
Хеш-таблицы	1	1		3	5
Коллекции в .NET	1	1		3	5
Двоичная куча. Очередь с приоритетами	1	1		5	7
Продвинутое алгоритмы сортировки	1	1		5	7
Графы	1	1		5	7
Динамическое программирование	1	1		5	7
<b>ИТОГО ЗА СЕМЕСТР</b>	<b>34</b>	<b>34</b>		<b>76</b>	<b>144</b>
<b>ИТОГО ПО КОМПОНЕНТУ ОПОП</b>	<b>51</b>	<b>68</b>		<b>133</b>	<b>252</b>

6.3. Форма обучения – заочная, курс – 1, семестр – 2

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
Раздел 1. Алгоритм. Интегрированная среда разработки	<b>1</b>	<b>2</b>		<b>30</b>	<b>33</b>
Основы алгоритмизации	0,5	1		15	16,5
Интегрированная среда разработки Python IDLE (Integrated Development and Learning Environment)	0,5	1		15	16,5
Раздел 2. Основы языка высокого уровня Python	<b>3</b>	<b>4</b>		<b>68</b>	<b>75</b>
Основные понятия языка высокого уровня Python	1	1		20	22
Структурное программирование	1	2		23	26
Структуры и типы данных. Парадигмы и технологии программирования	1	1		25	27
<b>ИТОГО ЗА СЕМЕСТР</b>	<b>4</b>	<b>6</b>		<b>98</b>	<b>108</b>

6.3. Форма обучения – заочная, курс – 2, семестр – 3

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
Раздел 3. Базовые алгоритмические конструкции	<b>1,6</b>	<b>1,6</b>		<b>28</b>	<b>31,2</b>
Введение в алгоритмизацию и программирование на с#	0,2	0,2		10	10,4
Интегрированная среда разработки MS Visual Studio.	1	1		10	12
Линейный вычислительный процесс	0,2	0,2		4	4,4
Разветвляющийся и циклический вычислительный процесс	0,2	0,2		4	4,4
Раздел 4. Пользовательские функции и типы данных	<b>1,6</b>	<b>1,6</b>		<b>29</b>	<b>32,2</b>
Массивы	0,5	0,5		6	7
Пользовательские функции. Рекурсия	0,5	0,5		6	7
Абстрактные синтаксические деревья. РБНФ-выражения	0,2	0,2		6	6,4
Работа со строками	0,2	0,2		6	6,4
Перечисляемый тип данных. Структуры	0,2	0,2		5	5,4
Раздел 5. Базовые алгоритмы и структуры данных	<b>1,4</b>	<b>1,4</b>		<b>35</b>	<b>37,8</b>



Работа с файлами	0,2	0,2		5	5,4
Простые алгоритмы сортировки	0,2	0,2		5	5,4
Простые алгоритмы поиска	0,2	0,2		5	5,4
Поиск последовательностей	0,2	0,2		5	5,4
Фундаментальные структуры данных и абстрактные типы данных	0,2	0,2		5	5,4
Стек, очередь, дек	0,2	0,2		5	5,4
Деревья	0,2	0,2		5	5,4
Раздел 6. Продвинутое алгоритмы и структуры данных	<b>1,4</b>	<b>1,4</b>		<b>40</b>	<b>42,8</b>
Хеш-таблицы	0,2	0,2		5	5,4
Коллекции в .NET	0,2	0,2		5	5,4
Двоичная куча. Очередь с приоритетами	0,2	0,2		7	7,4
Продвинутое алгоритмы сортировки	0,2	0,2		7	7,4
Графы	0,4	0,4		8	8,8
Динамическое программирование	0,2	0,2		8	8,4
ИТОГО ЗА СЕМЕСТР	<b>6</b>	<b>6</b>		<b>132</b>	<b>144</b>
ИТОГО ПО КОМПОНЕНТУ ОПОП	<b>10</b>	<b>12</b>		<b>230</b>	<b>252</b>

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

### 7.1. Контрольные вопросы

#### Раздел 1

1. Определение алгоритма. Свойства алгоритма. Формы записи алгоритмов. Примеры.
2. Определение алгоритма. Свойства алгоритма. Формы записи алгоритмов. Примеры.
3. Запись алгоритмов блок-схемами. Основные элементы блок-схем.
4. Алгоритмы с ветвлением. Пример алгоритма.
5. Алгоритм цикла с предусловием. Пример алгоритма.
6. Алгоритм цикла с постусловием. Пример алгоритма.
7. Алгоритм цикла с управляющей переменной. Пример алгоритма.
8. Что такое КуМир как язык и система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе.
9. Система КуМир. Алгоритмы и исполнители. Способы написания алгоритмов. Пульс в КуМир.

#### Раздел 2

10. Транслятор, интерпретатор, компилятор.
11. Назовите поколения языков программирования и их характеристики.
12. Дайте определение алфавита и лексики языка программирования. Приведите пример.
13. Дайте определение синтаксиса и семантики программирования. Приведите пример.
14. Из каких частей состоит исходная программа.
15. Основные типы данных языка Python.
16. Язык Python. Ввод-вывод данных. Линейные алгоритмы.
17. Язык Python. Условный оператор.
18. Язык Python. Разветвляющиеся алгоритмы.

19. Язык Python. Типы данных. Целый и вещественный типы данных. Операции с переменными этого типа. Привести пример.
20. Логический тип данных Boolean. Символьный тип данных. Операции с переменными этого типа. Привести пример использования в программном коде.
21. Язык Python. Циклы. Реализация и работа вложенных циклов.
22. Язык Python. Строковый тип данных.
23. Язык Python. Списки. Примеры использования.
24. Язык Python. Функции. Использование при решении математических задач.
25. Роль функций в языке Python. Как они объявляются. Какие следует указывать имена функций.
26. Какую роль в Python играет имя функции. Оператор вызова функций.
27. Оператор return в функциях. Особенность(и) его работы.
28. Способы передачи аргументов функции. Фактические и формальные параметры.
29. Способы объявлений функций с произвольным числом фактических и формальных параметров. Возможные их комбинации.

### **Раздел 3**

30. Основные парадигмы современного программирования. Сравнительный анализ с примерами кода.
31. Интегрированная среда разработки MS Visual Studio Community. Дополнительные инструменты. Ускорение процесса разработки программного обеспечения.
32. Компиляция. Линковка. Сравнение инструментов.

### **Раздел 4**

33. Создание и индексация массивов.
34. Двумерные массивы.
35. Арифметические операции и функции с массивами.
36. Массивы. Оператор foreach. Виды двумерных массивов. Инициализация одномерных и двумерных массивов. Генерация случайных чисел.
37. Пользовательские функции. Ключевые слова params, ref, out. Параметры по умолчанию. Оператор return.
38. Функции. Рекурсивный вызов функции. Использование на практике вычисления факториала целого числа.
39. Рекурсия. Базовый случай, шаг рекурсии. Стек вызовов. Рекурсивные алгоритмы нахождения НОД двух чисел и факториала числа.
40. Работа со строками как с массивом символов. Тип string. Основные операции со строками (поиск, вставка, удаление, замена подстроки).
41. Регулярные выражения. Основные синтаксические элементы регулярных выражений. Квантификаторы.
42. Тип данных - перечисление. Синтаксис. Занимаемый объем памяти. Модификации. Примеры.
43. Структуры в C#. Работа с датой и временем в C#.
44. Структура программы. Переменные и константы. Идентификаторы. Операторы. Выражения.
45. Абстрактные синтаксические деревья. РБНФ-выражения. Примеры.
46. Рекурсия. Примеры кодов рекурсивных алгоритмов. Фракталы.
47. Абстрактные синтаксические деревья. РБНФ-выражения.
48. Регулярные выражения. Примеры задач с решениями на основе строковых методов и регулярных выражений.

### **Раздел 5**

49. Работа с файлами. Разбор популярных форматов файлов (wav, mp3, bmp и др.).
50. Деревья бинарного поиска. Сбалансированные деревья. В-деревья.

51. Фундаментальные типы данных: char — символьные данные, int — целые, float — с плавающей точкой, double — двойной точности, void — без значения.
52. Что такое абстрактный тип данных (АТД)?

### **Раздел 6**

53. Хеш-таблицы. Хеш-функции. Использование в современных фреймворках.
54. Коллекции в .NET с нюансами реализации.
55. Поразрядная сортировка. Экзотические сортировки.
56. Графы. Продвинутое алгоритмы.
57. Динамическое программирование. Примеры задач с решением.
58. Алгоритм. Свойства алгоритма. Виды представления алгоритма.
59. Основные парадигмы и подходы современного программирования.
60. Интегрированная среда разработки MS Visual Studio. Работа с проектами. Отладка программ. Конфигурации Debug и Release.
61. Основные элементы фреймворка .NET: общезыковая среда выполнения (CLR) и базовая библиотека классов (BCL).
62. Процесс создания программы. Компиляция и линковка. Пункт меню Build и Debug в MS Visual Studio.
63. Простые типы данных .NET. Преобразование и приведение простых типов. Типизация и ключевое слово var.
64. Арифметические операции. Постфиксный и префиксный инкремент (декремент). Математические операции (возведение в степень, тригонометрия, логарифмы).
65. Простые и составные логические выражения. Логические операторы. Особенности "укороченных" логических операторов. Таблицы истинности.
66. Ветвления. Операторы if, switch, goto. Тернарный оператор (?).
67. Циклы. Операторы for, while, do while, continue, break.

### **7.2. Темы письменных работ (типы задач)**

Контрольная работа во втором семестре по проверке теоретических знаний – по всем изученным темам, с использованием указанных выше контрольных вопросов. Проводится в форме тестирования по теме «Алгоритмизация и основы языка Python».

### **7.3. Образец содержания экзаменационного билета**

Экзаменационный билет по курсу "Алгоритмизация и программирование на C#".

Семестр 3

Теоретические вопросы:

1. Объясните понятие алгоритма. Какие существуют виды алгоритмов?
2. Опишите основные этапы разработки программного обеспечения.

Практический вопрос:

Задача:

3. Написать программу, которая вычисляет факториал числа.

Требования:

- Программа должна принимать на вход целое число  $n$  ( $n > 0$ ).
- Программа должна вычислять факториал числа  $n$  ( $n! = 1 * 2 * 3 * \dots * n$ ).
- Программа должна выводить результат на экран.

Пример вывода:

Введите число: 5  
Факториал 5: 120

Дополнительно:

Вы можете добавить проверку входных данных (например, проверка на то, что  $n$  является целым числом больше 0).

Вы можете использовать различные методы для вычисления факториала (например, рекурсивный или итеративный).

Вы можете добавить в программу описание используемого алгоритма.

Критерии оценки:

Правильность решения: 5 баллов (за верное решение задачи и корректный вывод результата).

Ясность и структурированность кода: 3 балла (за понятный, читаемый и хорошо структурированный код).

Эффективность кода: 2 балла (за использование эффективных алгоритмов и методов).

Общий балл: 10 баллов.

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

## 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	Организационно-учебная работа в аудитории	5
	Самостоятельная работа	10
	Контрольная работа по теоретическому материалу	25
2	Организационно-учебная работа в аудитории	5
	Самостоятельная работа	25
ИТОГО		70
Зачет		30
Общий итог за семестр		100

### 8.2. Семестр 3

Номера разделов	Виды работ	Максимальное количество баллов
3	Организационно-учебная работа в аудитории	5
	Самостоятельная работа	5
4	Организационно-учебная работа в аудитории	10
	Самостоятельная работа	10
5	Организационно-учебная работа в аудитории	10
	Самостоятельная работа	10
6	Организационно-учебная работа в аудитории	5
	Самостоятельная работа	5
ИТОГО		
Экзамен		40
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 Главном корпусе ДонГУ (г. Донецк, пр. Гурова, 6), в Учебно-практическом вычислительном центре ДонГУ (г. Донецк, пр. Гурова, 6, корпус 12).

Для проведения лекций требуется аудитория, оборудованная меловой или маркерной доской / сенсорным экраном / мультимедийный проектор с экраном и ноутбуком, комплект учебной мебели для студентов, рабочее место преподавателя.

Для проведения практических занятий требуется аудитория, оборудованная меловой или маркерной доской / сенсорным экраном / мультимедийный проектор с экраном и ноутбук, комплект учебной мебели для студентов, рабочее место преподавателя.

Для проведения лабораторных занятий требуется аудитория, оборудованная маркерной доской или сенсорным экраном / мультимедийный проектор с экраном и ноутбук, персональные компьютеры, комплект учебной мебели для студентов, рабочее место преподавателя, выход в Интернет – проводной или с использованием Wi-Fi.

Для самостоятельной работы используются текстовые и электронные ресурсы Научной библиотеки университета и других электронных библиотечных баз данных, учебно-методическое обеспечение, представленное в аудиториях Главного корпуса (ауд. 511, 605, 610).

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. Чернышев, С. А. Основы программирования на Python: учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022.— (Высшее образование). — 286 с.
2. Сысоева М. В., Сысоев И. В. Программирование для «нормальных» с нуля на языке Python: Учебник. В двух частях. Часть 1 / Ответственный редактор: В. Л. Черный : — М.: Базальт СПО; МАКС Пресс, 2018. — 176 с. [+4 с. вкл]: ил. — (Библиотека ALT)
3. Nakov S. Fundamentals of Computer Programming with C# [Текст] / S. Nakov & Co. — BASD, Sofia, 2013. — 1121 p.

### 11.2. Дополнительная литература

4. Буйначев, С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91, с.
5. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с. — (Высшее образование). — ISBN 978-5-534- 14638-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/478098> (дата обращения: 13.07.2021).

6. Стивенс Р. Алгоритмы. Теория и практическое применение [Текст] / Р. Стивенс. – М.: Издательство «Э», 2016. – 544 с.
7. Дасгупта С. Алгоритмы [Текст] / С. Дасгупта, Х. Пападимитриу, У. Вазирани. – М.: МЦНМО, 2014. – 320 с.
8. Златопольский Д. М. Сборник задач по программированию [Текст] / Д. М. Златопольский. – СПб.: БХВ-Петербург, 2007. – 240 с.
9. Нейгел К. C#5.0 и платформа .NET 4.5 для профессионалов [Текст] / К. Нейгел, Б. Ивсен., Дж. Глини. – М.: Издательский дом «Вильямс», 2014. – 1440 с.
10. Кормен Т. Алгоритмы: Построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М.: МЦНМО, 1999. – 960 с.

## 12. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. **Национальная электронная библиотека (НЭБ):** федеральная государственная информационная система / Министерство Культуры РФ; Российская государственная библиотека. – Москва, 2019- . – URL: <https://rusneb.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный, подписка. Необходима установка программного обеспечения. – Текст: электронный.
2. **eLIBRARY.RU:** научная электронная библиотека: сайт. – Москва, 2000- . – URL: <https://elibrary.ru> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
3. Научная электронная библиотека **«КиберЛенинка»:** сайт / Ассоциация «Открытая наука». – Москва, 2014- . – URL: <https://cyberleninka.ru/>. – Режим доступа: свободный. – Текст: электронный.
4. Электронно-библиотечная система **«Лань»:** [сайт]. – URL: <https://e.lanbook.com> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
5. **ЭБС Юрайт:** электронная библиотечная система: сайт. – Москва, 2013. – URL: <https://biblio-online.ru> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
6. **Электронно-библиотечная система ДонГУ:** сайт / ФГБОУ ВО «ДонГУ». – Донецк, 2016- . – URL: <http://library.donnu.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный. – Текст: электронный.
7. **Электронный каталог** Научной библиотеки ДонГУ: раздел сайта / НБ ДонГУ. – Текст: электронный // ЭБС ДонГУ: сайт. – URL: <http://library.donnu.ru/catalog/> (дата обращения: 01.09.2023). – Режим доступа: поиск свободный, электронные документы – для пользователей ДонГУ.
8. **Электронный архив ДонГУ:** раздел сайта / НБ ДонГУ. – Текст: электронный // ЭБС ДонГУ: сайт. – URL: <http://repo.donnu.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный.

## 13. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. 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).
1. 5. Python IDLE (freeware), установщик с официального сайта <https://python.org>.
2. Sublime Text (freeware)
3. Visual Studio Code (freeware)

4. Atom (freeware, текстовый редактор)
5. Антивирус Касперского, Adobe Acrobat Reader, xPDF (лицензии GPL, Apache, BSD для свободного программного обеспечения).
6. IDE Visual Studio Community любой версии (рекомендованы версии от 2015 г.)
7. IDE MonoDevelop (для пользователей ОС Linux).
8. Текстовый редактор Notepad++ с дополнением HexEditor.