

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Горно-Алтайский государственный университет»
(ФГБОУ ВО ГАГУ, ГАГУ, Горно-Алтайский государственный университет)

Языки программирования рабочая программа дисциплины (модуля)

Закреплена за кафедрой **кафедра экономики, туризма и прикладной информатики**

Учебный план 09.03.03_2026_826.plx
09.03.03 Прикладная информатика
Инжиниринг информационных систем

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **4 ЗЕТ**

Часов по учебному плану	144	Виды контроля в семестрах: экзамен 3
в том числе:		
аудиторные занятия	44	
самостоятельная работа	63,3	
часов на контроль	34,75	

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		Итого	
	17			
Неделя				
Вид занятий	уп	рп	уп	рп
Лекции	14	14	14	14
Лабораторные	30	30	30	30
Консультации (для студента)	0,7	0,7	0,7	0,7
Контроль самостоятельной работы при проведении аттестации	0,25	0,25	0,25	0,25
Консультации перед экзаменом	1	1	1	1
Итого ауд.	44	44	44	44
Контактная работа	45,95	45,95	45,95	45,95
Сам. работа	63,3	63,3	63,3	63,3
Часы на контроль	34,75	34,75	34,75	34,75
Итого	144	144	144	144

Программу составил(и):

к.ф.-м.н, доцент, Губкина Елена Владимировна

Рабочая программа дисциплины

Языки программирования

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

утвержденного учёным советом вуза от 29.01.2026 протокол № 2.

Рабочая программа утверждена на заседании кафедры

кафедра экономики, туризма и прикладной информатики

Протокол от 09.04.2026 протокол № 9

Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2021-2022 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2021 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2022-2023 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2022 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2023-2024 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2023 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2024-2025 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2024 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	<i>Цели:</i> Сформировать компетенции обучающегося в области использования компьютера как средства управления информацией; изучить методы программирования для овладения знаниями в области технологии программирования; подготовить обучающихся к осознанному применению, как языков программирования, так и методов программирования.
1.2	<i>Задачи:</i> - знакомство с методами структурного и объектно-ориентированного программирования как наиболее распространенными и эффективными методами разработки программных продуктов; - обучение разработке алгоритмов на основе структурного и объектно-ориентированного подхода; - закрепление навыков алгоритмизации и программирования на основе изучения современных языков программирования; - знакомство с основными структурами данных и типовыми методами обработки этих структур.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.О
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Дискретная математика
2.1.2	Информатика и программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Базовое программирование на Python
2.2.2	Базы данных
2.2.3	Технологическая практика
2.2.4	Интернет-технологии. Интернет-программирование
2.2.5	Проектирование информационных систем
2.2.6	Проектный практикум
2.2.7	Разработка и стандартизация программных средств и информационных технологий
2.2.8	Технологическая (проектно-технологическая) практика
2.2.9	Преддипломная практика

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)**ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения;**

Уровень	ОПК-7: Применяет языки программирования и современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.
Идентификатор	ОПК-7: Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.
Знать:	синтаксис и семантику основных конструкций языков программирования высокого уровня, назначение и действие основных процедур и функций стандартных библиотек и модулей системы программирования.
Уметь:	формализовывать задачу; составлять алгоритмы; собирать программный код; пользоваться средствами отладки; пользоваться документацией и справочной системой; проектировать; документировать; тестировать свою программную разработку.
Владеть:	навыками применения языков программирования высокого уровня для проектирования и разработки информационных систем, для автоматизации бизнес-процессов, а также для решения прикладных задач и ведения БД и информационных хранилищ
владет:	- навыками программирования на языках высокого уровня; навыками отладки и тестирования прототипов имеющих ИС

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем / вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
-------------	--	----------------	-------	-------------	------------	------------	------------

	Раздел 1. Введение в алгоритмизацию и программирование						
1.1	Методологии программирования. /Лек/	3	1	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
1.2	Методологии программирования. /Лаб/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
1.3	Методологии программирования. /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
1.4	Алгоритмические структуры. /Лек/	3	1	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
1.5	Алгоритмические структуры. /Лаб/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
1.6	Алгоритмические структуры. /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
	Раздел 2. Синтаксис и основные конструкции языков программирования.						
2.1	Синтаксис и семантика формального языка. /Лек/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
2.2	Синтаксис и семантика формального языка. /Лаб/	3	4	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
2.3	Синтаксис и семантика формального языка. /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
2.4	Основные конструкции алгоритмических языков /Лек/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
2.5	Основные конструкции алгоритмических языков /Лаб/	3	4	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
2.6	Основные конструкции алгоритмических языков /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
	Раздел 3. Типы данных и операторы языков программирования						
3.1	Простые типы данных. /Лек/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
3.2	Простые типы данных. /Лаб/	3	4	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
3.3	Простые типы данных. /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
3.4	Основные операторы языков программирования. /Лек/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
3.5	Основные операторы языков программирования. /Лаб/	3	4	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
3.6	Основные операторы языков программирования. /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	

	Раздел 4. Структурный подход к программированию						
4.1	Структурированные типы данных языков программирования /Лек/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
4.2	Структурированные типы данных языков программирования /Лаб/	3	4	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
4.3	Структурированные типы данных языков программирования /Ср/	3	8	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
4.4	Алгоритмы поиска и сортировки. /Лек/	3	2	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
4.5	Алгоритмы поиска и сортировки. /Лаб/	3	6	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
4.6	Алгоритмы поиска и сортировки. /Ср/	3	7,3	ИД-1.ОПК-7 ИД-2.ОПК-7	Л1.1 Л1.2Л2.1 Л2.2	0	
	Раздел 5. Консультации						
5.1	Консультация по дисциплине /Конс/	3	0,7	ИД-1.ОПК-7 ИД-2.ОПК-7		0	
	Раздел 6. Промежуточная аттестация (экзамен)						
6.1	Подготовка к экзамену /Экзамен/	3	34,75	ИД-1.ОПК-7 ИД-2.ОПК-7		0	
6.2	Контроль СР /КСРАтт/	3	0,25	ИД-1.ОПК-7 ИД-2.ОПК-7		0	
6.3	Контактная работа /КонсЭк/	3	1	ИД-1.ОПК-7 ИД-2.ОПК-7		0	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Пояснительная записка

1. Назначение фонда оценочных средств. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Языки программирования
2. Фонд оценочных средств включает контрольные материалы для проведения текущего контроля и промежуточной аттестации вопросов к промежуточной аттестации

5.2. Оценочные средства для текущего контроля

Название вопроса*: 1 (ОПК-7)

Формулировка вопроса: Что не относится к базовым алгоритмическим конструкциям в языках программирования

Варианты ответов

- 1) линейные алгоритмы
- 2) циклические алгоритмы
- 3) регрессионные алгоритмы
- 4) алгоритмы выбора

Ключ: 3 регрессионные алгоритмы

Название вопроса*: 2 (ОПК-7)

Формулировка вопроса: Что относится к этапу программирования:

Варианты ответов

- 1) написание кода программы;
- 2) разработка интерфейса;
- 3) работоспособность;
- 4) анализ требований.

Ключ: 1) написание кода программы;

Название вопроса*: 3 (ОПК-7)

Формулировка вопроса: Формулировка вопроса: Сколько раз выполниться тело цикла

S=0

While S<10:

print(S)

S=S+1

Ключ: 9

Название вопроса*: 4 (ОПК-7)

Формулировка вопроса: : На обработку поступает натуральное число. Нужно написать программу, которая выводит на экран его первую (старшую) цифру.

Программист торопился и написал программу неправильно. Введите номер строки, в которой допущена ошибка

```
n = float(input())
```

```
while n >=10 :
```

```
    n //= 10
```

```
print(n)
```

Ключ: 1

Критерии оценки для всех аттестаций, проходящих в форме компьютерного тестирования.

менее 60% - неудовлетворительно

60%-74 %- удовлетворительно

75%-89%- хорошо

90% и более - отлично

Входной контроль

1. Команда input() используется для

Выберите один ответ:

a. для вывода данных на экран

b. для считывания данных с клавиатуры

2. Команда print() используется для

Выберите один ответ:

a. такой функции в этом языке нет

b. считывание данных с клавиатуры

c. вывода данных на экран

3. Конструкция d != 2 означает

Выберите один ответ:

a. значение переменной d не равно 2

b. переменной d присваивается значение 2

c. значение переменной d равно 2

4. Переменная int:

Выберите один ответ:

a. целая

b. символьная

c. логическая

d. вещественная

5.Преимуществами языка Python являются

Выберите один или несколько ответов:

a. динамическая типизация (для несложных программ)

b. наличие большой библиотеки классов

c. простота

d. встраиваемость

e. платформонезависимость

6. Вычислите остаток от деления.

```
2%5
```

7. Вычислите остаток от деления.

```
20%5
```

8. Укажите значение переменной s после выполнения следующего кода:

```
s = 0
k = 30
d = k - 5
k = 2 * d
s = k - 100
```

9. Укажите значение переменной x после выполнения следующего кода:

```
x = 3
y = 4
z = x + y
z = z + 1
x = y
y = 5
x = z + y + 7
```

Первая промежуточная аттестация

1. Какой тип будет у переменной x после выполнения данной программы

```
x = float(input())
```

```
if x > 0:
    print(x)
else:
    print(-x)
```

Выберите один ответ:

- a. int
- b. float
- c. long

2. Введено значение x=2. Чему будет равно значение y? введите ТОЛЬКО значение.

```
x = float(input())
```

```
if x < 0 :
```

```
    print("y=",0)
```

```
elif x > 0 and x <=1:
```

```
    print("y=",x)
```

```
elif x > 1:
```

```
    print("y=",pow(x,3))
```

3. Дана программа. Чему равно значение z?

```
from statistics import mean
```

```
mas = [2, 4, 8,-4]
```

```
z=mean(mas)
```

```
print("z=",z)
```

4. Дана программа на языке Python Чему равно значение переменной z после выполнения программы

```
mas = [2, 4, 8,-4]
```

```
z=sum(mas)
```

```
print("z=",z)
```

5. Дана программа. С клавиатуры ввели значения 2; 3; 4; 0. Чему будет равно значение переменной s после выполнения программы

```
s=1;
```

```
a=int(input("a="))
```

```
while a!=0:
```

```
    s=s*a
```

```
    a=int(input("a="))
```

```
print(s)
```

6. Дан массив `c=array([[1,2],[2,3],[4,5]])`. Чему равен срез `c[:1]`

Выберите один ответ:

a. `array[1,2,4]`

b. `array[2,3]`

c. `array[2,3,5]`

7. Как называется встроенный в языке Python тип данных неупорядоченной коллекции из нуля или более пар ключ-значение?

Выберите один ответ:

a. char

b. dict

c. string

8. Каким способом НЕЛЬЗЯ объявлять переменные в Python

Выберите один ответ:

a. `int a =5`

b. `a=int(5)`

c. `a=5`

9. Какой из типов данных не изменяемые

Выберите один ответ:

a. tuple

b. list

c. string

10. Определите какие из описаний верные

Выберите один или несколько ответов:

a. `mi_list=[True, 786, 'Text', 70,2]`

b. `second_list=(123,"Privet")`

c. `new_list={True, 786, 'Text', 70,2}`

11. Переменная `int`:

Выберите один ответ:

a. символьная

b. логическая

c. вещественная

d. целая

12. Результат вычисления выражения:

`(0 < 5 <= 3) and (0 / 0)`:

Выберите один ответ:

a. синтаксическая ошибка

b. True (или 1)

c. False (или 0)

13. Символ `#` в Python обозначает

Выберите один ответ:

- a. подключение дополнительных модулей
- b. такой символ не используется
- c. комментарии

14. Функция ввода с клавиатуры в языке Python

Выберите один ответ:

- a. input (a)
- b. print(a)
- c. int(a)

15. Сколько раз выполниться тело цикла

```
S=0
```

```
While S<10:
```

```
    print(S)
```

```
S=S+1
```

16. Что будет в результате выполнения следующего действия print (23 % 2)

17. Дан пример кода и сообщение об ошибке. Какой символ пропущен?

Пример кода

```
a = 5
```

```
if a = 5:
```

```
    print("Ура!")
```

Сообщение об ошибке

File "a.py", line 3

```
    if a = 5
```

```
        ^
```

SyntaxError: invalid syntax

Вторая промежуточная аттестация

1. Дан массив

```
a= array([ 0,  1,  8, 27, 64, 125, 216, 343, 512, 729])
```

что будет выведено на экран при выполнении команды a[8]

2. Библиотека matplotlib как добавить сетку

Выберите один ответ:

- a. pylab.plot
- b. pylab.grid
- c. pylab.legend

3. С какого индекса начинается нумерация элементов массива

4. Какой тип будет у переменной x после выполнения данной программы

```
x = float(input())
```

```
if x > 0:
```

```
    print(x)
```

```
else:
```

```
    print(-x)
```

Выберите один ответ:

- a. long
- b. int
- c. float

5. Дана программа. Чему равно значение z?

```
from statistics import mean
```

```
mas = [2, 4, 8, -4]
```

```
z=mean(mas)
```

```
print("z=",z)
```

6. Введите результат работы программы

```
list1 = ["1", "2", "3", "3"]
```

```
list2=[]
```

```
for i in range(0,len(list1)):
```

```
    list2.append(i*2)
```

```
print(list2)
```

7. Список это

Выберите один ответ:

- a. неизменяемая последовательность произвольных элементов
- b. изменяемая последовательность элементов одного типа
- c. неизменяемая последовательность элементов одного типа
- d. изменяемая последовательность произвольных элементов

8. Элементы списка перечисляются в

Выберите один ответ:

- a. {} фигурных скобках
- b. () круглых скобках
- c. [] квадратных скобках
- d. вообще без скобок

5.3. Темы письменных работ (эссе, рефераты, курсовые работы и др.)

1. Определение и проблемы языков программирования.

2. Парадигмы языков программирования.

3. Императивные языки программирования. Функциональные языки программирования

Объектно-ориентированные языки программирования. Логические языки программирования. Скриптовые языки программирования

6. Аппаратная организация компьютеров. Принцип программного управления. Структура виртуальной машины.

Порядок функционирования виртуальной машины.

7. Трансляция и интерпретация. Этапы трансляции.

8. Нотации выражений. Порядок вычислений. Присваивание.

9. Базовые операторы. Операторы перехода. Поток управления. Составные операторы.

10. Условные операторы. Операторы циклов.

1. Качество синтаксиса. Синтаксические элементы.

2. Абстрактные синтаксические деревья. Способы обхода деревьев для различных нотаций арифметических выражений.

Связь АСД с деревьями разбора.

3. Грамматики. КС-грамматика. Форма Бэкуса-Наура.

4. Деревья разбора. Выводы. Синтаксическая неоднозначность.
5. Списки. Расширенная форма Бекуса-Наура. Синтаксические схемы.
6. Синтезируемые атрибуты. Атрибутные грамматики.
7. Операционная и аксиоматическая семантика.
8. Правило упрощения. Правила вывода для последовательности, условного оператора и цикла while.
9. Определение инварианта цикла по индукции. Общая схема определения инварианта цикла.
10. Денотационная семантика.

Критерии оценки

Отметка «отлично» за письменную работу, реферат, сообщение ставится, если изложенный в докладе материал:

- отличается глубиной и содержательностью, соответствует заявленной теме;
- четко структурирован, с выделением основных моментов;
- доклад сделан кратко, четко, с выделением основных данных;
- на вопросы по теме доклада получены полные исчерпывающие ответы.

Отметка «хорошо» ставится, если изложенный в докладе материал:

- характеризуется достаточным содержательным уровнем, но отличается недостаточной структурированностью;
- доклад длинный, не вполне четкий;
- на вопросы по теме доклада получены полные исчерпывающие ответы только после наводящих вопросов, или не на все вопросы.

Отметка «удовлетворительно» ставится, если изложенный в докладе материал:

- недостаточно раскрыт, носит фрагментарный характер, слабо структурирован;
- докладчик слабо ориентируется в излагаемом материале;
- на вопросы по теме доклада не были получены ответы или они не были правильными.

Отметка «неудовлетворительно» ставится, если:

- доклад не сделан;
- докладчик не ориентируется в излагаемом материале;
- на вопросы по выполненной работе не были получены ответы или они не были правильными.

5.4. Оценочные средства для промежуточной аттестации

1. Определение программирования. Краткая история и классификация языков программирования.
2. Основные понятия алгоритмических языков. Постоянная.
3. Основные понятия алгоритмических языков. Переменная.
4. Основные понятия алгоритмических языков. Алфавит языка программирования.
5. Основные понятия алгоритмических языков. Синтаксис.
6. Основные понятия алгоритмических языков. Семантика.
7. Основные понятия алгоритмических языков. Лексема и грамматика языка.
8. Свойства ячеек оперативной памяти
9. Определение алгоритма. Исполнитель алгоритма.
10. Свойства алгоритмов.
11. Способы описания алгоритмов.
12. Стандартные типы данных. Целые типы.
13. Вещественные типы данных.
14. Символьные и логические типы данных.
15. Нестандартные типы данных.
16. Команда присваивания.
17. Стандартные функции и процедуры.
18. Команды ввода данных.
19. Команды вывода данных. Форматированный вывод.
20. Базовые алгоритмические структуры. Структура следования (линейная)
21. Базовые алгоритмические структуры. Полное логическое условие.
22. Базовые алгоритмические структуры. Неполное логическое условие
23. Базовые алгоритмические структуры. Структура выбора Case
24. Базовые алгоритмические структуры. Арифметический цикл For\do
25. Базовые алгоритмические структуры. Цикл While\DO (Оператор цикла с предварительным условием)
26. Массивы. Одномерные массивы.
27. Массивы. Матрицы.
28. Массивы. Сортировка.
29. Квадратная матрица.
30. Задача ввода и вывода элементов одномерного массива.
31. Ввод элементов одномерного массива случайным образом.
32. Ввод элементов двумерного массива.
33. Найти минимальный элемент в массиве
34. Написать программу, определяющую сумму элементов массива A(N).

35. . Дан целочисленный одномерный массив $A(N)$. Определить сумму четных элементов массива.
36. . Составить программу для построения таблицы умножения двух чисел (таблицы Пифагора) и занесения её в двумерный массив p . Вывести Массив на экран в виде таблицы.
37. . Дан целочисленный двумерный массив $A(N,N)$. Определить сумму элементов главной диагонали массива (матрицы).
38. . Вычисление элементов главной диагонали квадратной матрицы;
39. Вычисление элементов побочной диагонали квадратной матрицы;
40. Вычисление элементов, расположенных выше главной диагонали квадратной матрицы.
41. Вычисление элементов, расположенных ниже главной диагонали квадратной матрицы.
42. Вычисление элементов, расположенных выше побочной диагонали квадратной матрицы.
43. Вычисление элементов, расположенных ниже побочной диагонали квадратной матрицы.
44. . Написать программу, которая всюду в заданном тексте `text` будет, заменять каждое встретившееся слово `word1` другим словом такой же длины `word2`.
45. Тип строковый. Встроенные функции для строковых данных.
46. Тип запись.
47. Тип множество.
48. Подпрограммы- функции.
49. Описать функцию для вычисления функции $y=1/x$.
50. Описать функцию для вычисления $tg(x)$ и вычислить значение выражения $tg(x)+ctg(x)+tg^2(x)$.
51. Рекурсивные функции.
52. Пример вычисления факториала.

Критерии оценки

Отлично - если ответ показывает глубокое и систематическое знание всего программного материала и структуры конкретного вопроса, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой. Студент демонстрирует отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей научной области. Знание основной литературы и знакомство с дополнительно рекомендованной литературой. Логически корректное и убедительное изложение ответа.

Хорошо - знание узловых проблем программы и основного содержания лекционного курса; умение пользоваться концептуально-понятийным аппаратом в процессе анализа основных проблем в рамках данной темы; знание важнейших работ из списка рекомендованной литературы. В целом логически корректное, но не всегда точное и аргументированное изложение ответа.

Удовлетворительно – фрагментарные, поверхностные знания важнейших разделов программы и содержания лекционного курса; затруднения с использованием научно-понятийного аппарата и терминологии учебной дисциплины; неполное знакомство с рекомендованной литературой; частичные затруднения с выполнением предусмотренных программой заданий; стремление логически определено и последовательно изложить ответ.

Неудовлетворительно – незнание, либо отрывочное представление о данной проблеме в рамках учебно-программного материала; неумение использовать понятийный аппарат; отсутствие логической связи в ответе.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Дроботун Н. В., Рудков Е. О., Баев Н. А.	Алгоритмизация и программирование. Язык Python: учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020	https://www.iprbookshop.ru/102400.html
Л1.2	Громов Ю. Ю., Иванова О. Г., Кулаков Ю. В., Минин [и др.] Ю. В.	Методы программирования: учебное пособие	Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2012	https://www.iprbookshop.ru/63867.html

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Бульонков М. А., Емельянов П. Г., Скопин И. Н.	Базовые понятия и методы программирования: учебное пособие	Новосибирск: Новосибирский государственный университет, 2023	https://www.iprbookshop.ru/134565.html

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.2	Рик Гаско	Простой Python просто с нуля	Москва: СОЛОН-Пресс, 2023	https://www.iprbookshop.ru/142050.html
6.3.1 Перечень программного обеспечения				
6.3.1.1	MS Office			
6.3.1.2	MS Windows			
6.3.1.3	Яндекс.Браузер			
6.3.1.4	LibreOffice			
6.3.1.5	Kaspersky Endpoint Security для бизнеса СТАНДАРТНЫЙ			
6.3.1.6	NVDA			
6.3.1.7	РЕД ОС			
6.3.1.8	Python			
6.3.2 Перечень информационных справочных систем				

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ		
	метод проектов	
	презентация	
	дискуссия	

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)		
Номер аудитории	Назначение	Основное оснащение
322 А2	Компьютерный класс. Лаборатория информатики и информационно-коммуникативных технологий. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся). Компьютеры, ученическая доска, подключение к сети Интернет
134 А1	Центр стратегических исследований (лаборатория). Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся). Ученическая доска, интерактивная доска, проектор, ноутбук
319 А2	Компьютерный класс. Лаборатория региональной экономики. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся). Компьютеры, интерактивная доска с проектором, подключение к сети интернет

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)
<p>1. Планирование и организация времени, необходимого для изучения дисциплины. Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины: Изучение конспекта лекции в тот же день, после лекции – 10-15 минут. Изучение конспекта лекции за день перед следующей лекцией – 10-15 минут. Изучение теоретического материала по учебнику и конспекту – 1 час в неделю. Подготовка к лабораторному занятию – 30 мин. Всего в неделю – 2 часа 55 минут.</p> <p>2. Описание последовательности действий студента («сценарий изучения дисциплины»).</p> <p>При изучении дисциплины очень полезно самостоятельно изучать материал, который еще не прочитан на лекции. Тогда</p>

лекция будет гораздо понятнее. Однако легче при изучении курса следовать изложению материала на лекции. Для понимания материала и качественного

его усвоения рекомендуется такая последовательность действий:

1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной сегодня (10-15 минут).
2. При подготовке к лекции следующего дня, нужно просмотреть текст предыдущей лекции, подумать о том, какая может быть тема следующей лекции (10-15 минут).
3. В течение недели выбрать время (1 час) для работы с литературой по алгоритмическим методам в библиотеке или изучить дополнительную литературу в электронной форме.
3. Методические рекомендации по подготовке семинарских и практических занятий.

По данному курсу предусмотрены лабораторные занятия в компьютерном классе. При подготовке к лабораторным занятиям следует изучить соответствующий теоретический материал по языку программирования Python.

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучаются и книги по программированию.

Литературу по курсу «Язык программирования Python» рекомендуется изучать на сайтах <http://www.python.org/>, <http://www.enthought.com/products/epdlibraries.php>, которые содержат в свободном доступе все необходимые дистрибутивы, полную информацию и уроки по программированию на языке Python. Полезно использовать несколько учебников по курсу «Программирование на Python». Однако легче освоить курс придерживаясь одного учебника и конспекта. Рекомендуется, кроме «заучивания» материала, добиться состояния понимания изучаемой темы дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений по программированию на данную тему.

Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?. При изучении теоретического материала всегда нужно рисовать схемы или графики. Необходимо изучить лабораторную работу предыдущего занятия и выяснить те вопросы, которые показались непонятными. Полезно вначале попытаться написать программный код самостоятельно, а затем сравнить его с тем, что был рассмотрен на предыдущем занятии. Такой подход позволяет студентам быстрее освоить алгоритмические методы языка и сократить время на его изучение.

4. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучаются и книги по программированию. Литературу по курсу «Язык программирования Python» рекомендуется изучать на сайтах <http://www.python.org/>, <http://www.enthought.com/products/epdlibraries.php>, которые содержат в свободном доступе все необходимые дистрибутивы, полную информацию и уроки по программированию на языке Python. Полезно использовать несколько учебников по курсу «Программирование на Python». Однако легче освоить курс придерживаясь одного учебника и конспекта. Рекомендуется, кроме «заучивания» материала, добиться состояния понимания изучаемой темы дисциплины. С этой целью рекомендуется после изучения очередного параграфа

выполнить несколько простых упражнений по программированию на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?. При изучении теоретического материала всегда нужно рисовать схемы или графики.

5. Советы по подготовке к экзамену. Дополнительно к изучению конспектов лекции необходимо пользоваться учебником по программированию. Кроме «заучивания» материала зачета, очень важно добиться состояния понимания изучаемых тем дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений по программированию на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?. При изучении теоретического материала всегда нужно рисовать схемы или графики. В конце подготовки к зачету полезно самостоятельно написать программу зачета/экзамена.