

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ИНГУШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра «Информационные системы и технологии»**

**СОГЛАСОВАНО**

**УТВЕРЖДАЮ**

Руководитель образовательной программы

И.о. декана физико-математического  
факультета

\_\_\_\_\_/М.Х. Мальсагов

\_\_\_\_\_/Б.С.Кульбужев

«20» мая 2024г.

«23» мая 2024г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Б1.О.12 «Технологии программирования»**

**Направление подготовки**

**09.03.02 Информационные системы и технологии**

**Направленность (профиль подготовки)**

**Информационные системы и технологии**

**Квалификация выпускника**

**Бакалавр**

**Форма обучения**

**Очная, заочная, очно-заочная**

Магас, 2024г

# 1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

В процессе освоения образовательной программы компетенции формируются по следующим этапам:

- 1) начальный этап дает общее представление о виде деятельности, основных закономерностях функционирования объектов профессиональной деятельности, методов и алгоритмов решения практических задач;
- 2) основной этап позволяет решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам;
- 3) завершающий этап предполагает готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении.

Формируемые дисциплиной знания и умения готовят выпускника данной образовательной программы к выполнению следующих обобщенных трудовых функций (трудовых функций):

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	Код	Наименование	Уровень квалификации	Наименование	Код	Уровень (подуровень) квалификации
06.001 Программист	D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6	6
				Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6	6
				Проектирование программного обеспечения	D/03.6	6

При освоении дисциплины (модуля) компетенции, закрепленные за ней, реализуются по темам (разделам) дисциплины (модуля), в определенной степени (полностью или в оговоренной части) и на определенном этапе, что приведено в Таблице 1.

**Таблица 1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы**

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

<b>Код профессиональной компетенции</b>	<b>Наименование профессиональной компетенции</b>	<b>Код, наименование индикатора достижения профессиональной компетенции</b>	
<b>ОПК-3</b>	<b>ОПК-3. Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</b>	<p>ОПК-3.1. Знать: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. ОПК-3.2. Уметь: решать стандартные задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p> <p>ОПК-3.3. Иметь навыки: подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</p>	
<b>ПК-3</b>	<b>ПК-3.Способен оценивать качество программного обеспечения, в том числе проведение тестирования и исследование результатов</b>	<p>ПК-3.1. Знать: методы автоматической и автоматизированной проверки работоспособности программного обеспечения; основные виды диагностических данных и способы их представления; языки, утилиты и среды программирования, и средства пакетного выполнения процедур; типовые метрики программного обеспечения; основные методы измерения и оценки характеристик</p>	<p>06.004 Специалист по тестированию в области информационных технологий</p>

		<p>программного обеспечения; ПК-3.2. Уметь: писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования; использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования;</p> <p>ПК-3.3. Иметь навыки: разработки процедуры проверки работоспособности программного обеспечения; разработки процедуры сбора диагностических данных; разработки процедуры измерения требуемых характеристик программного обеспечения; оценки и согласование сроков выполнения поставленных целей.</p>	
ПК-9	<p><b>ПК-9. Способен выполнять работы по взаимодействию с заказчиком и другими заинтересованным и сторонами проекта, по организации заключения договоров, мониторингу и управлению исполнением договоров</b></p>	<p>ПК-9.1. Знать: стандарты и методики процессного подхода к ИТ; юридические основы договорной работы; принципы документооборота;</p> <p>ПК-9.2. Уметь: организовать процесс управления договорами об уровне предоставления сервисов ИТ; оценивать и оптимизировать процесс управления договорами об уровне предоставления сервисов ИТ;</p>	<p>06.015 Специалист по информационным системам 06.016 Руководитель проектов в области информационных технологий</p>

		ПК-9.3. Иметь навыки: формирования целей, приоритетов и ограничений процесса управления договорами об уровне предоставления сервисов ИТ и изменение их по мере изменения внешних условий и внутренних потребностей; организации персонала и выделение ресурсов для управления договорами об уровне предоставления сервисов ИТ; контроля выполнения договоров об уровне предоставления сервисов ИТ; анализа управления договорами об уровне предоставления сервисов ИТ, результатов их выполнения и выполнение управленческих действий по результатам анализа.	
--	--	--	--

## **2. Критерии оценивания образовательных результатов обучающегося во время промежуточной аттестации**

Промежуточная аттестация включает в себя теоретические задания, позволяющие оценить уровень усвоения обучающимися знаний, и лабораторные задания, выявляющие степень сформированности умений и владений .

Усвоенные знания и освоенные умения проверяются при помощи электронного тестирования, умения и владения проверяются в ходе выполнения лабораторных работ.

Объем и качество освоения обучающимися дисциплины, уровень сформированности дисциплинарных компетенций оцениваются по результатам текущих и промежуточной аттестаций количественной оценкой,

выраженной в баллах, максимальная сумма баллов по дисциплине равна 100 баллам.

Сумма баллов, набранных студентом по дисциплине, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика уровня освоения дисциплины
от 91 до 100	«отлично»	Студент демонстрирует сформированность дисциплинарной компетенции на итоговом уровне, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«хорошо»	Студент демонстрирует сформированность дисциплинарной компетенции на среднем уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«удовлетворительно»	Студент демонстрирует сформированность дисциплинарной компетенции на базовом уровне: в

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика уровня освоения дисциплины
		ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по дисциплинарной компетенции, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«неудовлетворительно»	Студент демонстрирует сформированность дисциплинарной компетенции на уровне ниже базового, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«неудовлетворительно»	Дисциплинарная компетенция не сформирована. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

### 3. Типовые материалы, необходимые для оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формирование оценки текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины осуществляется с использованием балльно-рейтинговой оценки работы студента. Вид учебной работы, за которую ставятся баллы

Вид учебной работы, за которую ставятся баллы	Баллы
Участие в online занятиях, прослушивание видео лекций	0-5
Практическая работа №1	0-10
Практическая работа №2	0-10

Практическая работа №3	0-10
Практическая работа №4	0-10
Практическая работа №5	0-10
Практическая работа №6	0-10
Тест №1	0-10
Тест №2	0-10
Итоговый тест	0-15
Всего	100

<b>БОНУСЫ</b> (баллы, которые могут быть добавлены до 100):	<b>Баллы</b>
- за активность	0 - 10
- за участие в ОЛИМПИАДЕ (в зависимости от занятого места)	0 - 50
- за участие в НИРС (в зависимости от работы)	0 - 50
- за оформление заявок на полезные модели (рацпредложения)	0 - 50

### **Балльная шкала оценки**

<b>Оценка(экзамен)</b>	<b>Баллы</b>
отлично	91-100
хорошо	81-90
удовлетворительно	61-80
неудовлетворительно	менее 61

### **3.1. Типовой вариант задания на контрольную работу**

#### **Задание 1. Программирование линейных алгоритмов (ввод и вывод данных)**

##### **Ход выполнения работы:**

Организовать ввод и вывод данных следующих типов: символов, строки символов, целых чисел, вещественных чисел. Согласно варианту (рисунок 1.1.)



Задание 1, № варианта		1	2	3	4
целые числа	количество чисел	2	3	4	2
	ширина поля	5	6	4	6
	к какому краю прижимать	п	л	п	п
вещественные числа	количество чисел	4	2	2	4
	ширина поля	6	7	6	6
	количество знаков в дробной части	2	3	3	2
	к какому краю прижимать	л	п	п	л
символы	количество символов	2	4	5	2
	ширина поля	6	6	7	5
	к какому краю прижимать	п	л	п	л
строка	ширина поля	8	4	8	2
	к какому краю прижимать	п	л	п	п

Рисунок 1.1 – Задание 1 для варианта 2

Код для выполнения задания 1:

```

#Задача 1, вариант 2
#Ввод и вывод целых чисел
print("Ввод и вывод целых чисел")
a = int(input("Введите первое число "))
print("|%-6d|" % a)
a = int(input("Введите второе число "))
print("|%-6d|" % a)
a = int(input("Введите Третье число "))
print("|%-6d|" % a)
#Вещественные числа
print("Ввод и вывод вещественных чисел")
a = float(input("Введите первое число "))
print("|%7.3f|" % a)
print("|%7.3e|" % a)
b = float(input("Введите второе число "))
print("|%7.3f|" % b)
print("|%7.3e|" % b)
#Ввод и вывод символов
print("Ввод и вывод символов")
a = (input("Введите первый символ "))
print("|%-6c|" % a)
b = (input("Введите второй символ "))
print("|%-6c|" % b)
c = (input("Введите третий символ "))
print("|%-6c|" % c)
d = (input("Введите четвертый символ "))
print("|%-6c|" % d)
#Ввод и вывод строки
print("Ввод и вывод строк")
a = (input("Введите первая строка "))
print("Вы ввели строку:%-4s" % a)

```

## Задание 2. Программирование линейных алгоритмов (использование математических функций)

### Ход выполнения работы:

Написать программу нахождения значения функции  $y(x)$ , его целой части и дробной части. Согласно варианту (рисунок 1.3)

2

$$\sin(\pi + \sqrt[3]{\sin^2 x})$$

Рисунок 1.3 – Задание 2 для варианта 2:

Код для выполнения задания 2:

```

#Задача 2. Вариант 2
from math import *
x = float(input('Введите x'))
y = sin(pi+sin(x)**(2/3))
print('y=%7.3f' % y)
print('y=', y)
print('целая часть =', int(y))
print('дробная часть =', y-int(y))

```

### Задание 3. Программирование циклических алгоритмов (Нахождение суммы с использованием цикла for)

#### Ход выполнения работы:

Найти сумму (согласно своего варианта) используя цикл for. (рисунок 1.1.)

2	$\sum_{n=1}^{12} \frac{n}{3n^3 + 5}$
---	--------------------------------------

Рисунок 1.1 – Задание 1 для варианта 2

Код для выполнения задания 1:

##Задание 1, Лаб 2, Вариант 2

```
from math import*
```

```
s=0
```

```
for n in range (1, 13):
```

```
a=n/(3*pow(n,3)+5)
```

```
s+=a
```

```
print ('Сумма равна %10.3f%s )
```

### Задание 4. Программирование циклических алгоритмов (Нахождение суммы с использованием цикла while)

#### Ход выполнения работы:

Найти сумму членов ряда больших заданного  $\varepsilon$  используя цикл. Согласно варианту (рисунок 1.4)

2	$\sum_{n=1}^{\infty} \frac{n}{3n^3 + 5}$	0,02
---	--	------

#### Рисунок 1.4 – Задание 2 для варианта 2

Код для выполнения задания 2:

```
## Задание 2, Лаб.2, вариант 2
```

```
from math import*
```

```
s=0
```

```
n=1
```

```
a= n /(3*n**3+5)
```

```
while a>0.02:
```

```
s+=a
```

```
n+=1
```

```
a= n /(3*n**3+5)
```

```
print ('Сумма равна %10.3f%s')
```

#### Задание 5. Программирование разветвляющихся алгоритмов

**Ход выполнения работы:**

Найти сумму функцию  $y(u, t)$  согласно варианту (рисунок 1.1.)

<b>2</b>	$y(u, t) = \begin{cases} u + t, & u > 1 \\ u - t, & 0 \leq u \leq 1 \\ t - u, & u < 0 \end{cases}$
----------	--

#### Рисунок 1.1 – Задание 1 для варианта 2

Код для выполнения задания 1:

```
##Задание 1, Лаб 3, Вариант 2
```

```
u = float(input("Введите значение u: "))
```

```
t = float(input("Введите значение t: "))
```

```
def y(u, t):
```

```
    if u > 1:
```

```
        return u + t
```

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

```
        return u - t
```

```
    else:
```

```
        return t-u
```

```
print(y(u, t))
```

## **Задание 6. Программирование с использованием вспомогательных функций**

### **Ход выполнения работы:**

Написать программу вычисления величины  $z$ , которая вычисляется по формуле согласно своему варианту (рисунок 1.1.)

2	$y(a, x) + y(a - 1, x + 1) + y(a^2 + x, 1)$
---	---

Рисунок 1.1 – Величина  $z$  для варианта 2

Вычисление функции  $y(u, t)$ , через которую описывается величина  $z$  (рисунок 1.2)

$$2 \quad y(u,t) = \begin{cases} u^2, & u < t-1 \\ t^2, & t-1 \leq u < t+1 \\ u + t^2, & u \geq t+1 \end{cases}$$

Рисунок 1.2 – Функции  $y(u,t)$ , для варианта 2

Код для выполнения задания 1:

##Задание 1, Лаб 5, Вариант 2

```
def y(u, t):
```

```
    if u<t-1:
```

```
        y=u**2
```

```
    elif u>=t-1 and u<t+1:
```

```
        y=t**2
```

```
    elif u>=t+1:
```

```
        y=t**2+u
```

```
    return y
```

```
a =float (input('Введите число а: '))
```

```
x =float (input('Введите число х: '))
```

```
s1=y(a,x)
```

```
s2=y(a-1,x+1)
```

```
s3=y(a**2+x,1)
```

```
w=s1+s2+s3
```

```
print (w)</t+1:
```

```
</t-1:
```

### **3.2. Перечень вопросов для подготовки к зачёту**

1. История и тенденции развития языков программирования
2. Области применения языка программирования Python
3. Переменные в Python. Наименование. Модель памяти Python при работе с переменными
4. Функции в Python. Создание функций
5. Создание программ на языке Python в отдельном файле. Отличие от интерактивного режима
6. Строки и операции над строками в языке Python
7. Операторы отношений в Python. Логические операции над объектами
8. Условная инструкция if
9. Модули в Python
10. Создание собственных модулей в Python
11. Строковые методы в Python. Отличие функций от методов
12. Списки в Python. Создание списка
13. Операции над списками в Python
14. Псевдонимы и копирование списков в Python
15. Методы списка в Python
16. Преобразование типов в Python (списки, строки)
17. Вложенные списки в Python
18. Циклы в Python
19. Цикл for для списков и строк в Python
20. Функция range() и цикл for в Python
21. Способы генерации списка в Python
22. Цикл while в Python
23. Вложенные циклы в Python (на примере вложенных списков)
24. Множества и операции над ними в Python
25. Кортежи и операции над ними в Python

- 26. Словари и операции над ними в Python
- 27. Обработка исключений в Python
- 28. Работа с файлами в Python. Менеджер контекста
- 29. Объектно-ориентированное программирование в Python. Классы, объекты
- 30. Иерархия наследования в Python (класс object)
- 31. Полиморфизм в Python
- 32. Структура оконного приложения на примере модуля tkinter (обработка событий)
- 33. Научные вычисления в Python.
- 34. Работа с xml-файлами в Python.