

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ухтинский государственный технический университет»
(УГТУ)
филиал Ухтинского государственного технического университета
в г. Усинске
(УФ УГТУ)
(среднего профессионального образования)

УТВЕРЖДАЮ
И. о. директора филиала
_____ О. В. Филиппова

«___» _____ 20__ г.

КОМПЛЕКС ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОПЦ. 09 Основы работы с информацией
образовательной программы
среднего профессионального образования
по специальности 09.02.11 Разработка и управление программным
обеспечением

г. Усинск
2026

1. ПАСПОРТ КОМПЛЕКТА ОЦЕНОЧНЫХ СРЕДСТВ

1.1 Область применения

Комплект оценочных средств (далее – КОС) предназначен для контроля и оценки результатов прохождения учебной дисциплины ОПЦ. 09 Основы работы с информацией, образовательной программы среднего профессионального образования (далее – ОП СПО) по специальности 09.02.11 Разработка и управление программным обеспечением.

1.2 Результаты освоения компетенции

В результате проведения промежуточной аттестации по дисциплине осуществляется комплексная оценка овладения следующими профессиональными и общими компетенциями:

Таблица 2.1

Код	Результат освоения компетенции
ПК 2.1	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

2. ФОРМЫ КОНТРОЛЯ И ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В соответствии с учебным планом и рабочей программой предусматривается текущий и промежуточный контроль результатов освоения.

Предметом оценки служат умения и знания, предусмотренные требованиями ФГОС по дисциплине 09.02.11 Разработка и управление программным обеспечением, направленные на формирование общих и профессиональных компетенций, а также личностных результатов в рамках программы воспитания.

2.1 Критерии и нормы оценки знаний обучающихся.

Оценка «5» ставится в случае:

- знания, понимания, глубины усвоения обучающимися всего объема программного материала;

- умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать межпредметные и внутри предметные связи, творчески применять полученные знания в незнакомой ситуации;
- отсутствия ошибок и недочётов при воспроизведении изученного материала, при устных ответах устранения отдельных неточностей с помощью дополнительных вопросов учителя, соблюдения культуры письменной и устной речи, правил оформления письменных работ.

Оценка «4» ставится в случае:

- знания всего изученного программного материала;
- умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать внутри предметные связи, применять полученные знания на практике;
- допущения незначительных (негрубых) ошибок, недочётов при воспроизведении изученного материала; соблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Оценка «3» ставится в случае:

- знания и усвоения материала на уровне минимальных требований программы, затруднения при самостоятельном воспроизведении, возникновения необходимости незначительной помощи преподавателя;
- умения работать на уровне воспроизведения, затруднения при ответах на видоизменённые вопросы;
- наличия грубой ошибки, нескольких грубых ошибок при воспроизведении изученного материала; незначительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Оценка «2» ставится в случае:

- знания и усвоения материала на уровне ниже минимальных требований программы; наличия отдельных представлений об изученном материале;
- отсутствия умения работать на уровне воспроизведения изученного материала, затруднения при ответе на стандартные вопросы;
- наличия нескольких грубых ошибок, большого числа негрубых при воспроизведении изученного материала, значительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

2.2 Критерии оценки за тестирование

Оценка 5 «отлично» (85 % -100 %)

Оценка 4 «хорошо» (75 % - 84 %)

Оценка 3 «удовлетворительно» (55 % -74 %)

Оценка 2 «неудовлетворительно» (менее 54 %)

2.3 Формы и методы оценивания учебной дисциплины «Технология разработки программного обеспечения»

Раздел, тема	Формы и методы контроля					
	Текущий контроль		Промежуточная аттестация		Итоговая аттестация	
	Формы контроля	Проверяемые Знания, умения, ОК, ЛР	Формы контроля	Проверяемые знания, умения ОК, ЛР	Формы контроля	Проверяемые ЗУН, ОК, ЛР
Часть 1. Методы и этапы технологии программирования.						
Тема 1. Эволюция технологии программирования. Основы программирования.	УО, ПР №1	ПК2.1, ПК2.4, ПК2.5.	ДФК, Т	ПК2.1, ПК2.4, ПК2.5.	Эк	ПК2.1, ПК2.4, ПК2.5.
Тема 2. Основные этапы технологии программирования. Основы программирования – условный оператор.	УО, ПР №2.	ПК2.1, ПК2.4, ПК2.5.	ДФК, Т	ПК2.1, ПК2.4, ПК2.5.	Эк	ПК2.1, ПК2.4, ПК2.5.
Тема 3. Пользовательский интерфейс. Основы программирования – типы данных.	УО, ПР №3, СР	ПК2.1, ПК2.4, ПК2.5.		ПК2.1, ПК2.4, ПК2.5.	Эк	ПК2.1, ПК2.4, ПК2.5.
Часть 2. Программирование на языке высокого уровня.						
Тема 4. Программирование на языке высокого уровня Python.	ПР № 4, СР.	ПК2.1, ПК2.4, ПК2.5.		ПК2.1, ПК2.4, ПК2.5.	Эк	ПК2.1, ПК2.4, ПК2.5.

Кодификатор оценочных средств

Функциональный признак оценочного средства (тип контрольного задания)	Код оценочного средства
Устный опрос	УО
Практическая работа	ПР №
Самостоятельная работа	СР
Тестирование	Т
Другая форма контроля	ДФК
Экзамен	Эк

3. Контрольно-оценочные материалы для промежуточной и итоговой аттестации по учебной дисциплине «Основы алгоритмизации и программирования»

3.1 Типовые задания для проведения текущий контроль

3.1.1 Перечень практических работ.

Практическая работа № 1.

Задача № 1. Напишите программу, которая считывает строку-разделитель и три строки, а затем выводит указанные строки через разделитель в следующем формате:

<вторая строка><строка-разделитель><третья строка><строка-разделитель><четвёртая строка>

Решение:

```
sep1=input()
a=input()
b=input()
c=input()
print(a,b,c, sep=sep1)
```

Задача № 2. Напишите программу, которая считает стоимость трёх компьютеров, состоящих из монитора, системного блока, клавиатуры и мыши. На вход программе подаются четыре целых числа (каждое на отдельной строке): стоимость монитора, стоимость системного блока, стоимость клавиатуры, стоимость мыши.

Решение:

```
a, b, c, d=int(input()), int(input()), int(input()), int(input())
print(3*(a+b+c+d))
```

Задача № 3. Арифметической прогрессией называется последовательность чисел $a_1, a_2, \dots, a_n$, каждое из которых, начиная с a_2 , получается из предыдущего путем прибавления к нему одного и того же постоянного числа d (разность прогрессии). Если известен первый член прогрессии (a_1) и её разность (d), то n -ый член арифметической прогрессии находится по формуле: $a_n = a_1 + d(n-1)$.

Решение:

```
a1, d, n=int(input()), int(input()), int(input())
print(a1+d*(n-1))
```

Практическая работа № 2.

Задача № 1. Напишите программу, которая определяет наименьшее из двух чисел. На вход программе подаются два различных целых числа. Программа должна вывести наименьшее из двух чисел.

Решение:

```
n1, n2 = int(input()), int(input())
if n1 < n2:
    print(n1)
else:
    print(n2)
```

Задача № 2. Напишите программу, которая считывает три числа и подсчитывает сумму только положительных чисел. На вход программе подаются три целых числа, каждое на отдельной строке. Программа должна вывести одно число – сумму положительных чисел.

Решение:

```
n1, n2, n3 = int(input()), int(input()), int(input())
sum = 0
if n1 >= 0:
    sum += n1
if n2 >= 0:
    sum += n2
if n3 >= 0:
    sum += n3
print(sum)
```

Задача № 3. Напишите программу, которая принимает целое число x и определяет, принадлежит ли данное число промежутку $(-1; 17]$. На вход программе подаётся целое число x . Программа должна вывести текст: «Не принадлежит» или «Принадлежит».

Решение:

```
x = int(input())
if -1 < x <= 17:
    print("Принадлежит")
else:
    print("Не принадлежит")
```

Задача № 4. Напишите программу, которая классифицирует треугольник на основе длин его сторон. Программа должна принимать три числа, каждое из которых представляет собой длину одной из его сторон. В результате программа должна определить, является ли треугольник равносторонним, равнобедренным или разносторонним. На вход программе подаются три числа (каждое на отдельной строке) – длины сторон существующего треугольника. Программа должна вывести на экран текст – тип треугольника («Равносторонний», «Равнобедренный» или «Разносторонний»).

Решение:

```
a, b, c = int(input()), int(input()), int(input())
if a == b == c:
    print("Равносторонний")
elif a == b or a == c or c == b:
    print("Равнобедренный")
else:
    print("Разносторонний")
```

Практическая работа № 3.

Задача № 1. Дано положительное действительное число. Выведите его дробную часть. На вход программе подаётся положительное действительное число. Программа должна вывести дробную часть числа.

Решение:

```
n = float(input())
n = n - int(n)
print(n)
```

Задача № 2. Напишите программу, определяющую площадь круга и длину окружности по заданному радиусу R . Программа должна вывести два числа (каждое на отдельной строке) – площадь круга и длину окружности радиуса R .

Решение:

```
from math import pi
r = float(input())
s = pi * r ** 2
c = 2 * pi * r
```

```
print(s, c, sep="\n")
```

Задача № 3. Вводятся 3 строки в случайном порядке. Напишите программу, которая выясняет, можно ли из длин этих строк построить арифметическую прогрессию. Программа должна вывести строку «YES» (без кавычек), если из длин введенных слов можно построить арифметическую прогрессию, или «NO» (без кавычек) в противном случае.

Решение:

```
n1, n2, n3 = input(), input(), input()
ln1=len(n1)
ln2=len(n2)
ln3=len(n3)
mn=min(ln1, ln2, ln3)
mx=max(ln1, ln2, ln3)
sr=ln1+ln2+ln3-mn-mx
if mx-sr==sr-mn:
    print("YES")
else:
    print("NO")
```

Практическая работа № 4.

Задача № 1. Даны два целых числа m и n ($m > n$). Напишите программу, которая выводит все нечётные целые числа от m до n (включительно) в порядке убывания. Программа должна вывести числа, каждое на отдельной строке.

Решение:

```
m, n = int(input()), int(input())
for i in range(m, n-1, -1):
    if i%2!=0:
        print(i)
```

Задача № 2. На вход программе подаётся натуральное число n . Напишите программу, которая вычисляет сумму всех его делителей.

Решение:

```
n = int(input())
sm=0
for i in range(1, n+1):
    if n%i==0:
        sm+=i
print(sm)
```

Задача № 3. Напишите программу, которая считывает 1010 чисел и выводит произведение отличных от нуля чисел.

Решение:

```
pr=1
for i in range(10):
    n = int(input())
    if n!=0:
        pr*=n
print(pr)
```

Задача № 4. На вход программе подаётся натуральное число n . Напишите программу, которая вычисляет $n!$.

Решение:

```
n = int(input())
k=1
for i in range(2, n+1):
    k*=i
print(k)
```

Задача № 5. На вход программе подаётся последовательность целых чисел от 11 до 55, характеризующее оценку ученика, каждое число на отдельной строке. Концом последовательности является любое неположительное число либо число, большее 55. Напишите программу, которая выводит количество пятерок.

Решение:

```
n = int(input())
kl=0
while 0 < n < 6:
    if n==5:
        kl+=1
    n = int(input())

print(kl)
```

Задача № 6. Дано натуральное число. Напишите программу, которая выводит его цифры в столбик в обратном порядке. Программа должна вывести цифры введённого числа в столбик в обратном порядке.

Решение:

```
n = int(input())
while n!=0:
    lst = n % 10
    n = n // 10
    print(lst)
```

Задача № 7. На вход программе подаётся число n ($n > 1$). Напишите программу, которая выводит его наименьший отличный от 11 делитель.

Решение:

```
n = int(input())
for i in range(2, n+1):
    if n%i==0:
        print(i)
        break
```

Задача № 8. Дано натуральное число nn . Напишите программу, которая выводит значение суммы: $1!+2!+3!+\dots+n!$

Решение:

```
n=int(input())
sm=1
mp=1
for i in range(2, n+1):
    mp=1
    for j in range(1, i+1):
        mp=mp*j
    sm=sm+mp
print(sm)
```

Задача № 9. На вход программе подаётся одна строка. Напишите программу, которая выводит:

- общее количество символов в строке;
- исходную строку, повторённую 3 раза;
- первый символ строки;
- первые три символа строки;
- последние три символа строки;
- строку в обратном порядке;
- строку с удалённым первым и последним символами.

Решение:

```
s = input()
print(len(s))
print(s*3)
print(s[0])
print(s[:3])
print(s[-3:])
print(s[::-1])
print(s[1:len(s)-1])
```

Задача № 10. На вход программе подаётся строка текста. Напишите программу, которая удаляет все вхождения символа «@».

Решение:

```
s = input()
print(s.replace("@", ""))
```

Задача № 11. На вход программе подаётся натуральное число n. Напишите программу, которая создаёт список, состоящий из делителей введённого числа, а затем выводит его.

Решение:

```
n = int(input())
sp = list()
k = n // 2 + 1
for i in range(1, k):
    if n % i == 0:
        sp.append(i)
sp.append(n)
print(sp)
```

Задача № 12. На вход программе подаётся строка текста, содержащая целые числа. Из данной строки формируется список чисел. Напишите программу, которая сортирует и выводит данный список сначала по возрастанию, а затем по убыванию.

Решение:

```
spis = input().split()
for i in range(len(spis)):
    spis[i] = int(spis[i])
spis.sort()
print(*spis)
spis.sort(reverse = True)
print(*spis)
```

Задача № 13. Напишите функцию print_digit_sum(), которая принимает одно натуральное число num и выводит на печать сумму его цифр.

Решение:

```
# объявление функции
def print_digit_sum(num):
    s = 0
    while num > 0:
        s += num % 10
        num = num // 10
    print(s)

# считываем данные
n = int(input())

# вызываем функцию
print_digit_sum(n)
```

Задача № 14. Напишите функцию merge(list1, list2), которая принимает в качестве аргументов два отсортированных по возрастанию списка, состоящих из целых чисел, и объединяет их в один отсортированный список.

Решение:

```

# объявление функции
def merge(list1, list2):
    t1 = 0
    t2 = 0
    spis = []
    for i in range(len(list1)):
        i = t1
        for j in range(t2, len(list2)):
            if list1[t1] < list2[j]:
                spis.append(list1[t1])
                t1 += 1
                break
            elif list1[t1] > list2[j]:
                spis.append(list2[j])
                t2 = j + 1
            else:
                spis.append(list1[t1])
                t1 += 1
                spis.append(list2[j])
                t2 = j + 1
                break

        if t2 >= len(list2):
            break

    if t2 >= len(list2) and t1 < len(list1):
        spis.extend(list1[t1:])
    elif t2 < len(list2) and t1 >= len(list1):
        spis.extend(list2[t2:])
    return spis

# считываем данные
numbers1 = [int(c) for c in input().split()]
numbers2 = [int(c) for c in input().split()]

# вызываем функцию
print(merge(numbers1, numbers2))

```

Задача № 15. Напишите функцию `is_valid_triangle(side1, side2, side3)`, которая принимает в качестве аргументов три натуральных числа, и возвращает значение `True`, если существует невырожденный треугольник со сторонами `side1`, `side2`, `side3`, или `False` в противном случае.

Решение:

```

# объявление функции
def is_valid_triangle(side1, side2, side3):
    M = max(side1, side2, side3)
    result = False
    if (side1 + side2 + side3 - M) > M:
        result = True
    return result

# считываем данные
a, b, c = int(input()), int(input()), int(input())

# вызываем функцию
print(is_valid_triangle(a, b, c))

```

Задача № 16. Напишите функцию `get_circle(radius)`, которая принимает в качестве аргумента радиус окружности и возвращает два значения: длину окружности и площадь круга, ограниченного данной окружностью.

Решение:

```

import math

# объявление функции

```

```
def get_circle(radius):
    c = 2 * math.pi * radius
    s = math.pi * (radius ** 2)
    return c, s

# считываем данные
r = float(input())

# вызываем функцию
length, square = get_circle(r)
print(length, square)
```

Задача № 17. Описание проекта: программа генерирует заданное количество паролей и включает в себя умную настройку на длину пароля, а также на то, какие символы требуется в него включить, а какие исключить.

Программа должна запрашивать у пользователя следующую информацию:

- Количество паролей для генерации;
- Длину одного пароля;
- Включать ли цифры 0123456789?
- Включать ли прописные буквы ABCDEFGHIJKLMNOPQRSTUVWXYZ?
- Включать ли строчные буквы abcdefghijklmnopqrstuvwxyz?
- Включать ли символы !#\$%&*+,-=?@^_?
- Исключать ли неоднозначные символы il1Lo0O?

Решение:

```
from random import *
digits = '0123456789'
lowercase_letters = 'abcdefghijklmnopqrstuvwxyz'
uppercase_letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
punctuation = '!#$%&*+,-=?@^_ '
chars = ''
def gen_pas(d, ch):
    password = ''
    for j in range(d):
        password += choice(ch)
    return password
print('Могу сгенерировать достаточно безопасный пароль!')
k = int(input('Количество паролей для генерации: '))
dl = int(input('Длина одного пароля: '))
zi = input('Включать ли цифры 0123456789 ? Наберите да или нет ')
sb = input('Включать ли строчные буквы abcdefghijklmnopqrstuvwxyz ? Наберите да или нет ')
pb = input('Включать ли прописные буквы ABCDEFGHIJKLMNOPQRSTUVWXYZ ? Наберите да или нет ')
sim = input('Включать ли символы !#$%&*+,-=?@^_ ? Наберите да или нет ')
ins = input('Исключать ли неоднозначные символы il1Lo0O ? Наберите да или нет ')
if zi == 'да': chars += digits
if sb == 'да': chars += lowercase_letters
if pb == 'да': chars += uppercase_letters
if sim == 'да': chars += punctuation
if ins == 'да':
    for i in 'il1Lo0O':
        chars = chars.replace(i, '')
for i in range(k):
    print(gen_pas(dl, chars))
```

3.1.2. Примерные вопросы для устного опроса обучающихся по учебной дисциплине

Тема 1. Эволюция технологии программирования

- 1.1. Какие этапы эволюции прошли технологии программирования?
- 1.2. Какие языки и методы программирования вы знаете?

- 1.3. Какие языки программирования называются языками высокого уровня?
- 1.4. Какая модель построения программ лежит в основе технологии процедурного программирования?
- 1.5. В чем преимущества и недостатки языков сценария?
- 1.6. Какова область применения языков параллельных вычислений?

Тема 2. Основные этапы технологии программирования

- 2.1. Каковы основные этапы решения задач на ЭВМ?
- 2.2. Что такое жизненный цикл программного обеспечения?
- 2.3. Что такое спецификации, какие сведения они содержат?
- 2.4. Что представляет собой структурный анализ?
- 2.5. В чем заключается этап реализации программного обеспечения?
- 2.6. Какие методы оценки трудоемкости разработки программного обеспечения вы знаете?
- 2.7. Какие способы записи алгоритма вы знаете?
- 2.8. Какие работы выполняются в процессе обеспечения качества программного продукта?
- 2.9. Какие основные группы ошибок в программных продуктах вы знаете?
- 2.10. Какие стадии тестирования ПО вы знаете?
- 2.11. Каковы основные методы отладки?
- 2.12. Что подразумевается под защитным программированием?
- 2.13. Каковы основные группы документации программного обеспечения?
- 2.14. Какие существуют формы записи алгоритма?
- 2.15. По каким критериям оценивается качество программы?
- 2.16. Какие факторы влияют на качество программ?
- 2.17. Что представляет собой инструментарий технологии программирования?

Тема 3. Пользовательский интерфейс

- 3.1. Каковы основные типы пользовательских интерфейсов?
- 3.2. В чем преимущество интерфейса со свободной навигацией по сравнению с интерфейс-меню?
- 3.3. Какие интерфейсы называются графическими?
- 3.4. Какие интерфейсы используются при объектно-ориентированном подходе к программированию?
- 3.5. Что такое диалог?
- 3.6. Какие элементы пользовательских интерфейсов относятся к интеллектуальным?

3.1.3 Перечень самостоятельных работ.

Задача № 1. Напишите программу, которая определяет, является ли год с данным номером високосным. Если год является високосным, то выведите «YES» (без кавычек), иначе выведите «NO» (без кавычек). Год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400.

Решение:

```
g=int(input())
if g%4==0 and g%100!=0 or g%400==0:
    print("YES")
else:
    print("NO")
```

Задача № 2. Напишите программу, которая считывает с клавиатуры два целых числа и

строку. Если эта строка является обозначением одной из четырёх математических операций (+, -, *, /), то выведите результат применения этой операции к введённым ранее числам, в противном случае выведите «Неверная операция» (без кавычек). Если пользователь захочет поделить на ноль, выведите текст «На ноль делить нельзя!» (без кавычек).

Решение:

```
a= int(input())
b= int(input())
oper=input()
if oper=="-":
    print(a-b)
elif oper=="+":
    print(a+b)
elif oper=="*":
    print(a*b)
elif oper=="/":
    if b==0:
        print("На ноль делить нельзя!")
    else:
        print(a/b)
else:
    print("Неверная операция")
```

Задача № 3. Дано натуральное число. Используя цикл while, напишите программу, которая вычисляет:

- сумму его цифр;
- количество цифр в нем;
- произведение его цифр;
- среднее арифметическое его цифр;
- его первую цифру;
- сумму его первой и последней цифры.

Решение:

```
n = int(input())
ns = n
kl=0
sm=0
pr=1
while n!=0:
    lst=n%10
    sm+=lst
    pr*=lst
    kl+=1
    n = n // 10
n1 = ns // 10**(kl-1)
nk = ns%10
sm1 = n1 + nk

print(sm, kl, pr, sm/kl, n1, sm1, sep="\n")
```

Задача № 4. Дано натуральное число n ($n \leq 9$). Напишите программу, которая печатает таблицу сложения для всех чисел от 1 до n (включительно). Таблицу сложения подразумеваем от 1 до 9 (включительно).

Решение:

```
n = int(input())
for i in range(1, n+1):
    for j in range(1, 10):
        print(i, "+", j, "=", i+j)
    print()
```

Задача № 5. Напишите программу, которая выводит «Correct» (без кавычек), если никнейм соответствует всем правилам, или «Incorrect» (без кавычек) в противном случае. Правила для корректного никнейма в соцсети Y следующие:

- никнейм должен начинаться с символа @
- никнейм должен содержать от 5 до 15 (включительно) символов (включая первый символ @)
- никнейм должен содержать только строчные буквы и цифры (помимо первого символа @)

Решение:

```
s = input()
n = len(s)
if 5 <= n <= 15 and s[0] == "@" and s[1:].isalnum() == True:
    if s[1:].isdigit() == True:
        print("Correct")
    elif s.islower() == True:
        print("Correct")
    else:
        print("Incorrect")
else:
    print("Incorrect")
```

3.2. Типовые задания для проведения промежуточной аттестации

Другая форма контроля

Часть 1. Методы и этапы технологии программирования.

Тема 1. «Эволюция технологии программирования»

Выберите один правильный ответ

- 1) Как называется совокупность методов и средств, используемых в процессе разработки программного обеспечения?
 - a) структура программирования
 - b) условия программирования
 - c) технология программирования
 - d) операции программирования
- 2) Что позволило программистам вместо кодов использовать mnemonic обозначения кодов операций и символические имена данных?
 - a) разработка аппаратных средств
 - b) появление ассемблеров
 - c) создание совместной разработки ПО
 - d) возникновение библиотеки подпрограмм
- 3) Что означает инкапсуляция?
 - a) каждый объект полностью описывается совокупностью своих свойств и методов
 - b) каждый субъект объемно описывается совокупностью своих свойств и методов
 - c) каждый предмет частично описывается совокупностью своих свойств и методов
 - d) каждый субъект выборочно описывается совокупностью своих свойств и методов
 - e) каждый объект фрагментарно описывается совокупностью своих свойств и методов
- 4) Как называется совокупность объектов, имеющих одинаковые свойства и методы?
 - a) класс
 - b) вид
 - c) атрибут

- d) тип
- 5) **Какое программирование возникло как упрощение функционального программирования для математиков и лингвистов, решающих задачи символьной обработки?**
- a) логическое
 - b) многослойное
 - c) ориентированное
 - d) наследственное
- 6) **В рамках какого стиля программирования программа представляет собой совокупность утверждений, описывающих фрагмент предметной области или сложившуюся ситуацию?**
- a) декларативный стиль программирования
 - b) полиморфный стиль программирования
 - c) императивный стиль программирования
 - d) абстракционный стиль программирования
- 7) **Что является основой среды JavaBeans?**
- a) объединённая программная модель
 - b) межкомпонентная предметная модель
 - c) компонентная объектная модель
 - d) компонентная субъектная модель
- 8) **Какое программирование предполагает наличие языковых средств, позволяющих выделять сквозную функциональность в отдельные модули?**
- a) объектно-ориентированное
 - b) предметно-ориентированное
 - c) субъектно-ориентированное
 - d) аспектно-ориентированное
- 9) **Как определяется приложение, способное самостоятельно реализовывать задачу, имеющую несколько путей решения?**
- a) аспект
 - b) субъект
 - c) объект
 - d) предмет
- 10) **В чем заключается метод структурного программирования процедур? Выберите один или несколько вариантов:**
- a) использование операторов ветвления и циклов
 - b) разделение процедур на вложенные блоки
 - c) сохранение текста процедуры
 - d) объединение процедур из вложенных блоков
 - e) форматирование текста процедуры
 - f) исключение операторов ветвления и циклов
- 11) **Какой технологии соответствует модель взаимодействия типа «клиент — сервер»? Выберите один или несколько вариантов:**
- a) технология CORBA
 - b) технология CASE
 - c) технология EJB
 - d) технология COM

Тема 2. «Основные этапы технологии программирования»

- 12) Расположите цифры в таком порядке, чтобы критерия качества программы соответствовали их содержанию. Перечислите цифры подряд, без пробелов.

1. Надежность	Оценка расходов вычислительных ресурсов
2. Дружественность	Способность к внесению изменений и расширений функций обработки
3. Мобильность	Возможность интеграции с другими программами
4. Эффективность	Обеспечение удобного интерфейса для работы пользователя
5. Коммуникативность	Независимость программы от программных и технических средств обработки данных
6. Модифицируемость	Устойчивость в работе программы, точность выполнения функций обработки

Ответ:
465231

- 13) Расположите цифры в таком порядке, чтобы основные свойства алгоритмов соответствовали их содержанию. Перечислите цифры подряд, без пробелов.

1. результативность	Получение искомого ответа через конечное число шагов
2. массовость	Расчлененность алгоритма на отдельные этапы
3. детерминированность	Пригодность алгоритма для решения всех задач данного типа
4. дискретность	Точность и понятность, обеспечивающие однозначность результата

Ответ:
1423

- 14) Установите верный порядок «стадии разработки» ПО по ГОСТ 19.102—77.

- (1) Стадия «Техническое задание»
- (2) Стадия «Эскизный проект»
- (3) Стадия «Технический проект»
- (4) Стадия «Рабочий проект»

Ответ:
4321

- 15) Какая модель жизненного цикла программного обеспечения предполагает переход на следующий этап после завершения всех операций предыдущего этапа?

- a) ядерная модель
- b) спиральная модель
- c) модель с промежуточным контролем
- d) каскадная модель

- 16) Как называется совокупность взаимосвязанных программ, совместно решающих небольшой класс задач некоторой прикладной области?

- a) программные комплексы
- b) программные системы
- c) пакеты программ
- d) архитектура программ

- 17) Как именуется процесс формулировки назначения программного обеспечения и основных требований к нему?

- a) постановка задачи
 - b) реализация
 - c) проектирование
 - d) определение спецификаций
- 18) Как называется процесс поиска и исправления ошибок, обнаруженных при тестировании программы? (в поле ответа введите одно слово)
Ответ: отладка
- 19) Установите верную последовательность стадий тестирования согласно рекомендациям Microsoft.
- (1) компоновочное тестирование
 - (2) бета-тестирование
 - (3) стресс-тестирование
 - (4) модульное тестирование
 - (5) приемно-сдаточное тестирование
 - (6) системное тестирование
- Ответ: 416325
- 20) Какие процессы разработки программного обеспечения сопровождаются документами? Выберите один или несколько вариантов:
- a) передачи информации между пользователями программного обеспечения
 - b) передачи пользователям информации, необходимой для эксплуатации ПО
 - c) передачи информации между разработчиками программного обеспечения
 - d) передачи разработчикам информации, необходимой для эксплуатации ПО
 - e) управления использованием программного обеспечения
 - f) управления разработкой программного обеспечения
- 21) Какие документы включаются в состав документации по сопровождению? Выберите один или несколько вариантов
- a) документы, описывающие разработку ПО
 - b) документы, описывающие структуру ПО
 - c) документы, описывающие настройки ПО
 - d) документы, описывающие процессы установки ПО
 - e) документы, описывающие состав ПО

Часть 2. Основы программирования.

- 22) Какая команда используется для вывода (печати) данных:
- a) `print()`
 - b) `cout()`
 - c) `printf()`
 - d) `Console.WriteLine()`
- 23) Как называют значения для вывода, указываемые через запятую в команде `print()`?
- a) строками
 - b) параметрами
 - c) символами
 - d) аргументами
- 24) Что покажет приведённый ниже код?
- ```
print('Hello, it's me!')
```
- a) Hello, it's me!

- b) 'Hello, it's me!'
- c) Hello, it
- d) произойдёт ошибка

25) Какая команда используется для считывания данных с клавиатуры:

- a) scanf()
- b) cin
- c) Console.ReadLine()
- d) input()

26) Какая из указанных строк считывает целое число в переменную n?

- a) n = integer(input())
- b) n = int(input())
- c) n = input()

27) Укажите подряд цифры с верными утверждениями:

- (1) Имя переменной не может начинаться с цифры
- (2) Имя переменной не может оканчиваться цифрой
- (3) Имя переменной может начинаться с символа подчёркивания (\_)
- (4) Имя переменной не должно совпадать с ключевым (зарезервированным) словом

**Ответ: 134**

28) Что покажет приведённый ниже код?

```
a = 4
print(a, 'a')
```

- a) 4 4
- b) a a
- c) 4 a
- d) a 4

29) Что покажет приведённый ниже код?

```
print(4 ** 2 * 3 - 1)
```

- a) 23
- b) 47
- c) 32

30) Вызов команды print() с пустыми скобками:

- a) вставит один пробел
- b) выполнит перевод строки
- c) ничего не выведет

31) Параметр sep в команде print() выполняет:

- a) вставку символов в конец всех аргументов;
- b) разделение между аргументов;
- c) перевод строки.

32) Укажите количество видов деления в Python:

- a) 1
- b) 2
- c) 3
- d) 4

33) Какой арифметических операторов имеет высший приоритет после скобок:

- a) Деление и умножение
- b) Унарный минус

- с) **Возведение в степень**
- 34) Какой арифметический оператор выполняет целочисленное деление:
- а) знак процента (%)
  - б) слэш (/)
  - с) двойной слэш (//)
- 35) Какой арифметический оператор находит остаток от деления:
- а) знак процента (%)
  - б) слэш (/)
  - с) двойной слэш (//)
- 36) Какой логический оператор имеет наивысший приоритет:
- а) логическое сложение or
  - б) логическое отрицание not
  - с) логическое умножение and
- 37) Чтобы значение выражения с оператором and было истинным, должны быть:
- а) Истинно хотя бы одно из выражений
  - б) Истины все выражения
  - с) Истинно одно выражение
- 38) Чтобы значение выражения с оператором or было истинным, должны быть
- а) Истинно хотя бы одно выражение
  - б) Истины все выражения
  - с)
- 39) Верно ли указан оператор сравнения:
- ```
if age >= 7 and <= 9:
```
- а) Верно
 - б) Неверно

3.2 Организация и проведение итоговой аттестации

3.2.1 Условия проведения итоговой аттестации. Подготовка к проведению итоговой аттестации.

Итоговая аттестация проводится в период экзаменационной сессии, установленной календарным графиком учебного процесса рабочего учебного плана. С формами проведения промежуточной аттестации обучающиеся знакомятся в течение двух месяцев с начала обучения.

Для подготовки к итоговой аттестации студентом (не позднее чем за 20 дней до проведения экзамена в соответствии с календарным графиком учебного процесса) выдается перечень тем к практическим заданиям, составленный исходя из требований ФГОС и рабочей программы предмета к уровню умений и знаний. Количество вопросов и практических задач в перечне для подготовки к промежуточной аттестации превышает количество вопросов и практических задач, необходимых для составления контрольно-измерительных материалов.

На основе разработанного и объявленного обучающимся перечня вопросов и практических задач, рекомендуемых для подготовки к итоговой аттестации, составлены задания, содержание которых до обучающихся не доводится. Вопросы и практические

задачи носят равноценный характер. Формулировки вопросов четкие, краткие, понятные, исключают двойное толкование.

3.2.2 Проведение итоговой аттестации

Студенты для сдачи итоговой аттестации распределяются по времени. На выполнение задания студенту отводится не более двух академических часов. Оценка, полученная на итоговой аттестации, заносится преподавателем в зачетную книжку студента (кроме неудовлетворительной) и экзаменационную ведомость (в том числе и неудовлетворительные). Экзаменационная оценка по предмету за данный семестр является определяющей, независимо от полученных в семестре оценок текущего контроля по предмету

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Теоретические вопросы:

Часть 1. Методы и этапы технологии программирования.

1. Какая модель построения программ лежит в основе технологии процедурного программирования? Каковы основные методы процедурного программирования?
2. На чем основывается концепция объектно-ориентированного программирования? Назовите основные принципы объектно-ориентированного программирования.
3. Что такое компонентные технологии и САЭ-технологии?
4. В чем преимущества и недостатки языков сценария?
5. Что такое жизненный цикл программного обеспечения? Какие модели жизненного цикла программного обеспечения вы знаете?
6. Что называется архитектурой программного обеспечения?
7. Каковы основные типы пользовательских интерфейсов?
8. Что такое спецификации, какие сведения они содержат?
9. Какие диаграммы включает язык UML?
10. Что представляет собой структурный анализ? Какие правила лежат в основе структурного анализа?
11. Какая модель построения программы используется при объектно – ориентированном подходе?
12. В чем заключается этап реализации программного обеспечения?
13. Назовите методы оценки трудоемкости разработки программного обеспечения.
14. Назовите виды организации коллектива разработчиков программного обеспечения.
15. Какие работы выполняются в процессе обеспечения качества программного продукта?
16. Назовите основные группы ошибок в программных продуктах?
17. Назовите стадии тестирования ПО? Каковы основные методы отладки?
18. Что подразумевается под защитным программированием?
19. Каковы основные группы документации программного обеспечения?
20. Какими свойствами обладают алгоритмы? Какие существуют формы записи алгоритма?
21. По каким критериям оценивается качество программы? Какие факторы влияют на качество программ?
22. Что представляет собой инструментарий технологии программирования?
23. Каковы основные типы пользовательских интерфейсов?
24. В чем преимущество интерфейса со свободной навигацией по сравнению с интерфейс-меню?

25. Какие интерфейсы называются графическими?
26. Какие интерфейсы используются при объектно-ориентированном подходе к программированию?
27. Что такое диалог? Какие типы диалога вы знаете? Какие формы диалога вы знаете?
28. Каковы основные компоненты графических пользовательских интерфейсов?
29. Перечислите виды пиктограмм.
30. Какие элементы пользовательских интерфейсов относятся к интеллектуальным?

Часть 2. Программирование на языке высокого уровня - Python.

1. Что выполняют команды `print()` и `input()`? Как влияет на команду `print()` параметры `sep` и `end`?
2. Условный оператор. Чем отличается вложенные от каскадных операторов.
3. Условный оператор. Какие логические операции возможно использовать в условном операторе.
4. Целочисленный тип данных и числа с плавающей точкой. Арифметические операторы.
5. Целочисленный тип данных и числа с плавающей точкой. Явное и неявное преобразование типов данных.
6. Строковый тип данных. Как определить длину строки. Конкатенация строк. Умножение строки на число.
7. Модуль `math`. Особенности подключения модулей. Основные функции и константы модуля `math`.
8. Цикл `for`. Какие параметры используются в функции `range()`.
9. Какое назначение имеют переменные: счетчик, сумматор, мультипликатор, Алгоритм обмена значений переменных.
10. Какое назначение имеют переменные: счетчик, сумматор, мультипликатор. Расширенные операторы присваивания.
11. Операторы `break`, `continue`. Блок `else` в циклах.
12. Вложенные циклы. Операторы `break` и `continue` во вложенных циклах.
13. Индексация строк. Итерирование строк.
14. Срезы строк. Параметра среза строк. Срез до конца, от начала. Отрицательные индексы в срезе.
15. Срезы строк. Параметра среза строк. Как выполнить изменение символа строки.
16. Строковый тип данных, основные методы конвертации регистра.
17. Строковый тип данных, основные методы поиска и замены.
18. Строковый тип данных, основные методы классификации символов.
19. Строковый тип данных, основные методы форматирования строк.
20. Сравнение строк единичной длины. Сравнение строк не единичной длины.
21. Списки. Создание и вывод списка. Встроенная функция `list()`.
22. Основы работы со списками. Функция `len()`. Оператор `in`. Индексация.
23. Основы работы со списками. Срезы. Операция конкатенации и умножения на число. Встроенные функции `sum()`, `min()`, `max()`.
24. Добавление элементов в список. Удаление элементов из списка.
25. Способы вывод элементов списка и строки.
26. Строковые методы `split()` и `join()`.
27. Основные назначения методов списков: `insert()`, `index()`, `remove()`, `pop()`, `reverse()`.
28. Основные назначения методов списков: `count()`, `clear()`, `copy()`, `sort()`, `reverse()`.
29. Создание пользовательских функций без параметров.
30. Создание пользовательских функций с параметрами.

Практические задания:

1. Напишите программу, которая принимает три положительных числа и определяет, существует ли невырожденный треугольник с такими сторонами. Треугольник существует, если выполняется неравенство треугольника: $a+b > c$; $a+c > b$; $b+c > a$, где a, b, c - стороны треугольника.

Решение:

```
a, b, c, = int(input()), int(input()), int(input())
if a+b>c and b+c>a and c+a>b:
    print("YES")
else:
    print("NO")
```

2. Напишите программу, которая упорядочивает три числа в порядке убывания.

Решение:

```
a1, a2, a3 = int(input()), int(input()), int(input())
mx=max(a1, a2, a3)
mn=min(a1, a2, a3)
sr=a1+a2+a3-mn-mx
print(mx, sr, mn, sep="\n")
```

3. На вход программе подаётся натуральное число n . Напишите программу, которая вычисляет сумму всех его делителей.

Решение:

```
n = int(input())
sm=0
for i in range(1, n+1):
    if n%i==0:
        sm+=i
print(sm)
```

4. На вход программе подаётся натуральное число n . Напишите программу, которая для каждого из чисел от 0 до n (включительно) выводит текст в следующем формате: Квадрат числа <текущее число> равен <квадрат текущего числа> .

Решение:

```
n= int(input())
for i in range(n+1):
    print("Квадрат числа", i, "равен", i*i)
```

5. На вход программе подаётся последовательность слов, каждое слово на отдельной строке. Концом последовательности является слово «КОНЕЦ» (без кавычек). При этом само слово «КОНЕЦ» не входит в последовательность, лишь символизируя её окончание. Напишите программу, которая выводит члены данной последовательности.

Решение:

```
ln = input()
while ln!="КОНЕЦ":
    print(ln)
    ln = input()
```

6. На вход программе подаётся число n ($n>1$). Напишите программу, которая выводит его наименьший отличный от 1 делитель.

Решение:

```
n = int(input())
for i in range(2, n+1):
    if n%i==0:
        print(i)
```

`break`

7. Даны два целых числа m и n ($m \leq n$). Напишите программу, которая выводит все целые числа от m до n включительно.

Решение:

```
m, n = int(input()), int(input())
for i in range(m, n+1):
    print(i)
```

8. На вход программе подаются два целых числа a и b ($a \leq b$). Напишите программу, которая подсчитывает количество чисел в диапазоне от a до b (включительно), куб которых оканчивается на 4 или 9.

Решение:

```
a, b = int(input()), int(input())
b+=1
k=0
for i in range(a, b):
    if i**3%10==4 or i**3%10==9:
        k+=1
print(k)
```

9. На вход программе подаётся натуральное число n . Напишите программу, которая подсчитывает сумму тех чисел от 1 до n (включительно), квадрат которых оканчивается на 2, 5 или 8.

Решение:

```
n=int(input())
sm=0
for i in range(1, n+1):
    if i**2%10==2 or i**2%10==5 or i**2%10==8:
        sm+=i

print(sm)
```

10. Напишите программу, которая считывает 8 чисел и выводит произведение отличных от нуля чисел.

Решение:

```
pr=1
for i in range(8):
    n = int(input())
    if n!=0:
        pr*=n
print(pr)
```

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ухтинский государственный технический университет»
(УГТУ)
филиал Ухтинского государственного технического университета
в г. Усинске
(УФ УГТУ)
(среднего профессионального образования)

УТВЕРЖДАЮ
И. о. директора филиала
О. В. Филиппова



КОМПЛЕКС ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОПЦ. 09 Основы работы с информацией
образовательной программы
среднего профессионального образования
по специальности 09.02.11 Разработка и управление программным
обеспечением

г. Усинск
2026

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ухтинский государственный технический университет»
(УГТУ)
филиал Ухтинского государственного технического университета
в г. Усинске
(УФ УГТУ)
(среднего профессионального образования)

УТВЕРЖДАЮ
И. о. директора филиала

О. В. Филиппова
« 4 » 20 26 г.

(подпись) (И. О. Фамилия)

« » 20 г.

(подпись) (И. О. Фамилия)

« » 20 г.

РАБОЧАЯ ПРОГРАММА

Дисциплина: Основы работы с информацией

Индекс: ОПЦ.09

Специальность: 09.02.11 Разработка и управление программным
обеспечением

Форма
обучения: очная

Курс (ы) 2

Семестр (ы): 4

г. Усинск
2026

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением приказ Минпросвещения России от 24.02.2025 N 138.

Разработчик Халафеев А.И. преподаватель СПО УФ УГТУ.

Рассмотрено на заседании ученого совета УФ УГТУ (протокол от 26.05.2026 № 06).

СОГЛАСОВАНО

Заместитель директора УФ УГТУ
по учебной работе

О.В. Филиппова