**Московский Государственный Университет Геодезии и Картографии**

**Факультет геоинформатики и информационной безопасности**

**Кафедра информационно-измерительных систем (ИС)**

Направление: Прикладная информатика

**Отчет**

**о выполненных практических работах по дисциплине ИНФОРМАТИКА**

*(индивидуальные задания: №4а, №4b, №5a=№3, №5b, №5c №6, №7)*

Проверила: Выполнил:

доц. кафедры ИС студент 2024 ПИ-1(б)

Лапчинская М.П. Струков Артемий Викторович­­

**МОСКВА – 2024**

**СОДЕРЖАНИЕ**

**стр.**

**ВВЕДЕНИЕ**

**1. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ №4 (ВАРИАНТ …) 3**

**1.1. Структурное (императивное) программирование на Python 4**

**1.1.1. Программирование без пользовательских функций (№4a)**

1.1.1.1. Постановка задачи №4а

1.1.1.2. Используемые инструкции в программе

1.1.1.3. Блок-схема решения задачи

1.1.1.4. Результаты работы программы (скриншоты)

1.1.1.5. Текст отлаженной программы

**1.1.2. Программирование с пользовательскими функциями (№4b)**

1.1.2.1. Постановка задачи №4b

1.1.2.2. Используемые инструкции в программе

1.1.2.3. Блок-схема решения задачи

1.1.2.4. Результаты работы программы (скриншоты)

1.1.2.5. Текст отлаженной программы

**2. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ № 5 (ВАРИАНТ …)**

**2.1. Структурное (императивное) программирование на Python**

**2.1.1. Программирование без пользовательских функций (№5a=№3)**

2.1.1.1. Используемые инструкции в программе

2.1.1.2. Блок-схема решения задачи

2.1.1.3. Результаты работы программы (скриншоты)

2.1.1.4. Текст отлаженной программы

**2.1.2. Программирование с пользовательскими функциями (№5b)**

2.1.2.1. Используемые инструкции в программе

2.1.2.2. Блок-схема решения задачи

2.1.2.3. Результаты работы программы (скриншоты)

2.1.2.4. Текст отлаженной программы

**2.2. Объектно-ориентированное программирование на Python (№5c)**

2.2.1. Используемые инструкции в программе

2.2.2. Результаты работы программы (скриншоты)

2.2.3. Текст отлаженной программы

**3. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ № 6 (ВАРИАНТ…)**

**3.2. Объектно-ориентированное программирование на Python (№6)**

3.2.1. Используемые инструкции в программе

3.2.2. Результаты работы программы (скриншоты)

3.2.3. Текст отлаженной программы

**3. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ № 7 (ВАРИАНТ …)**

**3.1. Объектно-ориентированное программирование на Python**

3.2.1. Описание алгоритма вычисления определенного интеграла по методу средних прямоугольников

3.2.2. Разработка пользовательского интерфейса программы с использованием   
виджетов PyQt

3.2.3. Используемые библиотеки в программе

3.2.4. Результаты работы программы (скриншоты)

3.2.5. Текст отлаженной программы

**ВЫВОДЫ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ**

**1. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ №4 (ВАРИАНТ 25)**

Дан двумерный массив **A** размером **n\*m** элементов, заполненный целыми случайными целыми числами из диапазона **(r1, r2)**. Вывести на экран исходный массив и результаты вычислений/преобразований с точностью до **3-го** знака.

**1.1. Структурное (императивное) программирование (№4a)**

**1.1.1. Программирование без пользовательских функций**

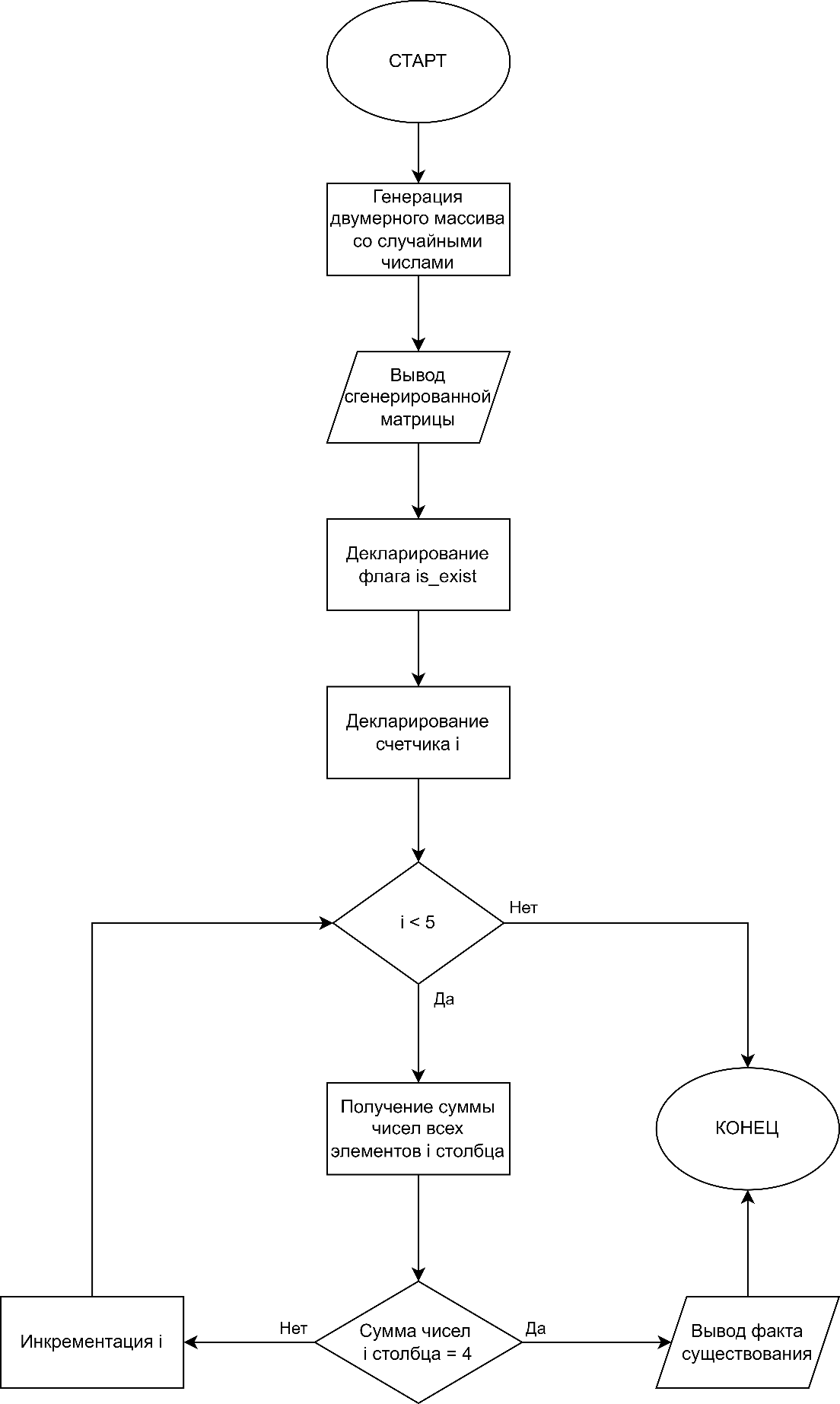
**1.1.1.1. Постановка задачи №4а**

Дан двумерный массив A (Двумерный массив описать с использованием списочного типа данных размером 5\*8 элементов, заполненный случайными целыми числами из диапазона (-100,100). Определить, есть ли в данном массиве столбец, в котором равное количество положительных и отрицательных элементов.

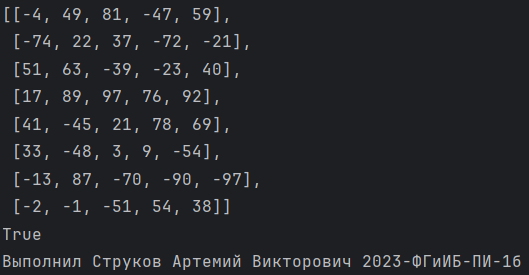
**1.1.1.2. Используемые инструкции в программе**

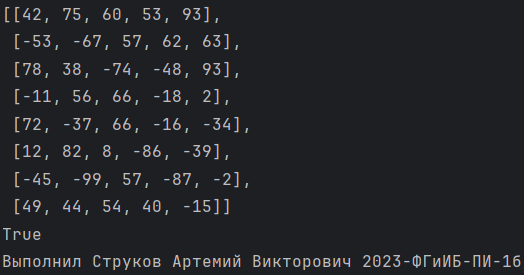
|  |  |  |
| --- | --- | --- |
| **№** | **Инструкция  (из кода программы)** | **Название и назначение инструкции** |
| **1.** | **is\_exist = False** | **Инструкция присваивания. Присваивание переменной is\_exist значения логического типа False с помощью оператора равенства =** |
| **2.** | **randint(-100, 100)** | **Инструкция вызова функции. Выполнение ранее написанного кода выделенного в функцию с передачей входных аргументов через запятую** |
| **3.** | **for i in range(5):** | **Цикл. Цикл с переменной i, которая выполняет роль счетчика, принимает значения от 0 до 5 (не включительно)** |
| **4.** | **if is\_exist:**  **pass**  **else: pass** | **Условная конструкция. Принимает логическое значение is\_exist, если оно верно вызывает тело if, иначе тело else** |
| **5.** | **break** | **Инструкция остановки выполнения цикла.** |
| **6.** | **data = [[randint(-100, 100) for \_ in range(5)] for \_ in range(8)]** | **Двумерное списочное выражение. Возвращает двумерный список размера 8 строк на 5 колонок** |

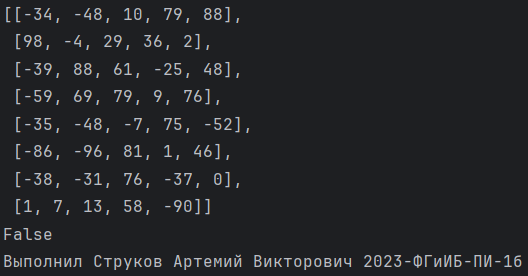
**1.1.1.3. Блок-схема решения задачи**



**1.1.1.4. Результаты работы программы (скриншоты)**







**1.1.1.5. Текст отлаженной программы**

from pprint import pprint  
from random import randint  
  
data = [[randint(-100, 100) for \_ in range(5)] for \_ in range(8)]  
  
pprint(data)  
is\_exist = False  
for i in range(5):  
 is\_exist = sum([int(elem > 0) for elem in [data[j][i] for j in range(8)]]) == 4  
 if is\_exist:  
 break  
print(is\_exist)  
  
print("Выполнил Струков Артемий Викторович 2023-ФГиИБ-ПИ-1б")

**1.1.2. Программирование с пользовательскими функциями (№4b)**

**Дан двумерный массив A размером n\*m элементов, заполненный целыми случайными целыми числами из**

**диапазона (r1, r2). Вывести на экран исходный массив и результаты вычислений/преобразований с точностью до 3-го знака.**

**Указания:**

**1. Двумерный массив описать с использованием списочного типа данных**

**2. Не использовать стандартные функции min, max, sum и т.п.**

**(написать свои функции).**

**3. Результат представить в виде пользовательской функции.**

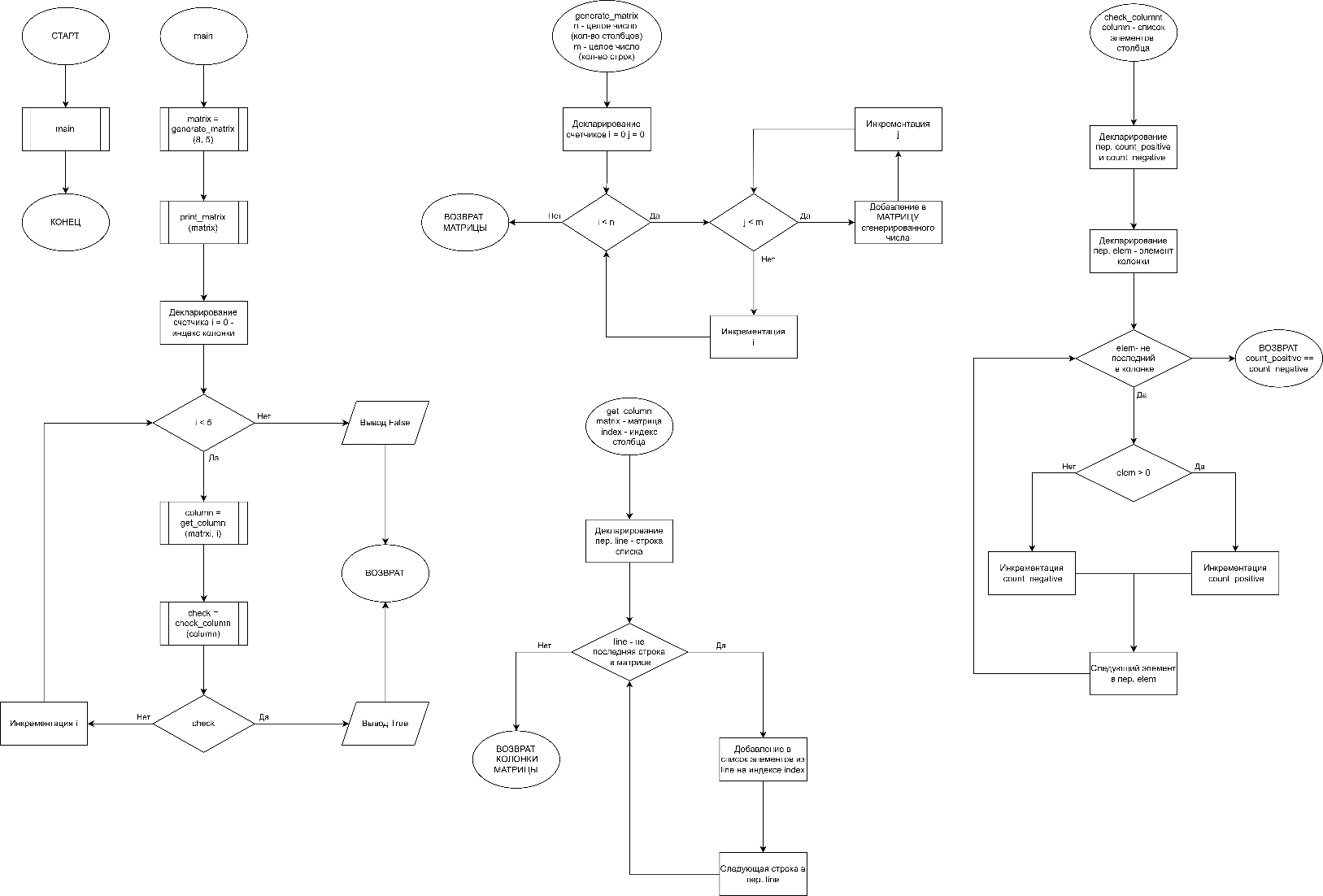
**1.1.2.1. Постановка задачи №4b**

**Дан двумерный массив A размером 8\*5 элементов, заполненный случайными целыми числами из диапазона (-100,100). Определить, есть ли в данном массиве столбец, в котором равное количество положительных и отрицательных элементов.**

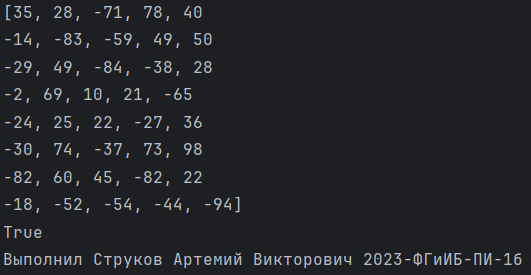
**1.1.2.2. Используемые инструкции в программе**

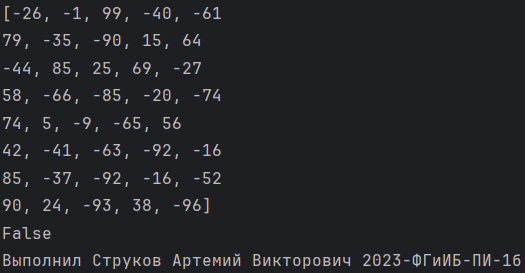
|  |  |  |
| --- | --- | --- |
| **№** | **Инструкция  (из кода программы)** | **Название и назначение инструкции** |
| **1.** | **is\_exist = False** | **Инструкция присваивания. Присваивание переменной is\_exist значения логического типа False с помощью оператора равенства =** |
| **2.** | **randint(-100, 100)** | **Инструкция вызова функции. Выполнение ранее написанного кода выделенного в функцию с передачей входных аргументов через запятую** |
| **3.** | **for i in range(5):** | **Цикл. Цикл с переменной i, которая выполняет роль счетчика, принимает значения от 0 до 5 (не включительно)** |
| **4.** | **if is\_exist:**  **pass**  **else: pass** | **Условная конструкция. Принимает логическое значение is\_exist, если оно верно вызывает тело if, иначе тело else** |
| **5.** | **break** | **Инструкция остановки выполнения цикла.** |
| **6.** | **data = [[randint(-100, 100) for \_ in range(5)] for \_ in range(8)]** | **Двумерное списочное выражение. Возвращает двумерный список размера 8 строк на 5 колонок** |
| **7.** | **def name (n: int, m: int) -> int** | **Декларирование функции с названием name с входными аргументами и указанием их типов, и указанием возвращающегося типа функции.** |

**1.1.2.3. Блок-схема решения задачи**



**1.1.2.4. Результаты работы программы (скриншоты)**

****

****

**1.1.2.5. Текст отлаженной программы**

from random import randint  
  
  
def generate\_matrix(n: int, m: int) -> list[list[int]]:  
 return [[randint(-100, 100) for j in range(m)] for i in range(n)]  
  
  
def print\_matrix(matrix: list[list[int]]):  
 print("[" + "\n".join([", ".join([str(elem) for elem in line]) for line in matrix]) + "]")  
  
  
def check\_column(column: list[int]) -> bool:  
 count\_positive, count\_negative = 0, 0  
 for elem in column:  
 if elem < 0:  
 count\_negative += 1  
 else:  
 count\_positive += 1  
 return count\_positive == count\_negative  
  
  
def get\_column(matrix: list[list[int]], index\_column: int) -> list[int]:  
 return [line[index\_column] for line in matrix]  
  
  
def main():  
 data = generate\_matrix(8, 5)  
  
 print\_matrix(data)  
 for i in range(5):  
 column = get\_column(data, i)  
 if check\_column(column):  
 print(True)  
 return  
 print(False)  
  
  
main()  
  
print("Выполнил Струков Артемий Викторович 2023-ФГиИБ-ПИ-1б")

**2. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ № 5 (ВАРИАНТ 24)**

Разработать программу табулирования заданной функции f(x) с учетом

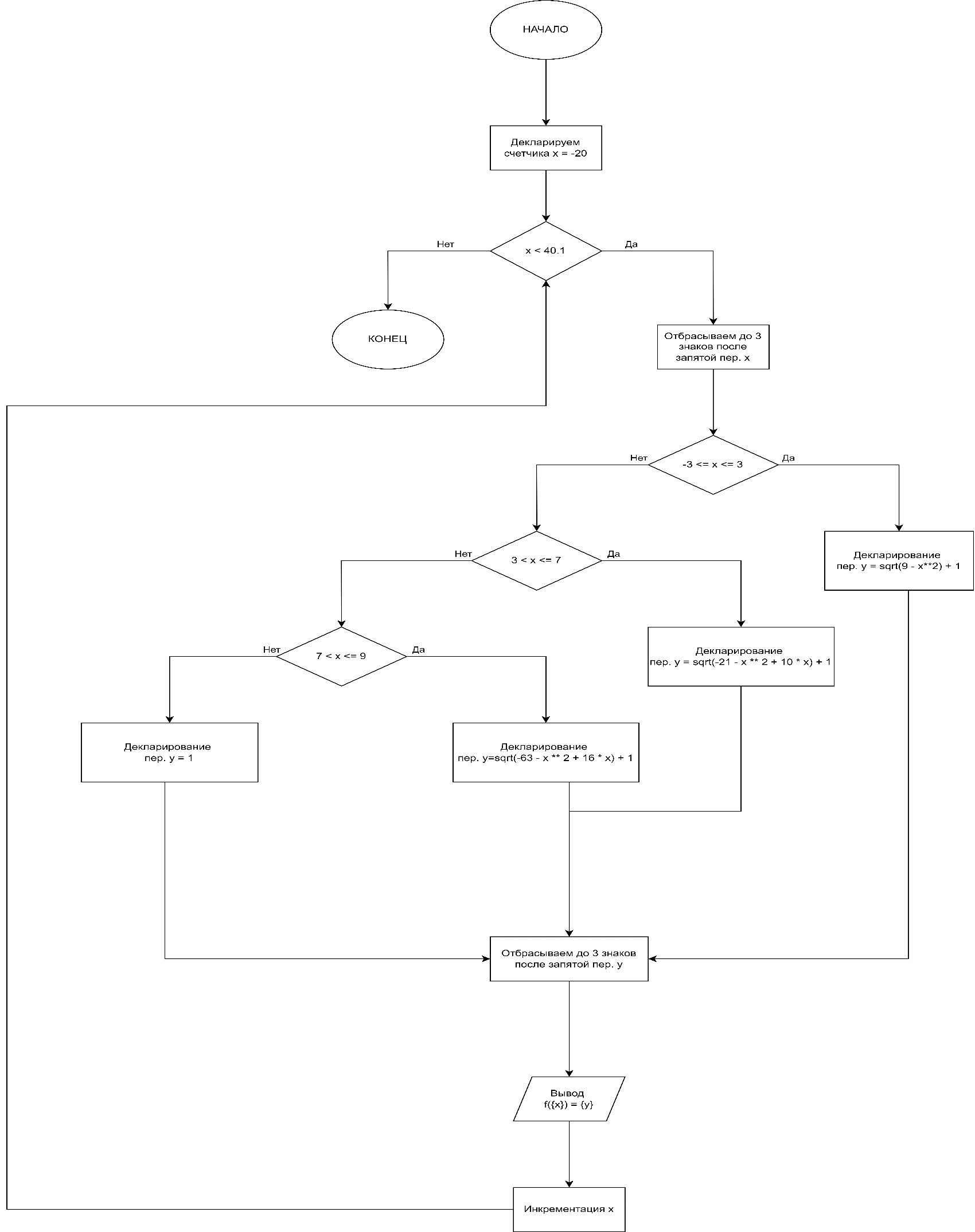
области ее определения при изменении аргумента х от начального значения X0 до конечного значения Xn постоянным шагом hx. Значения функции выводить с точностью З знака после запятой.

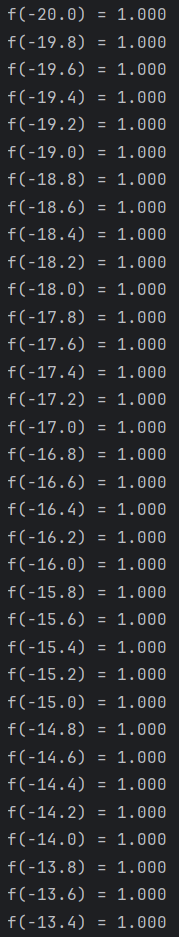
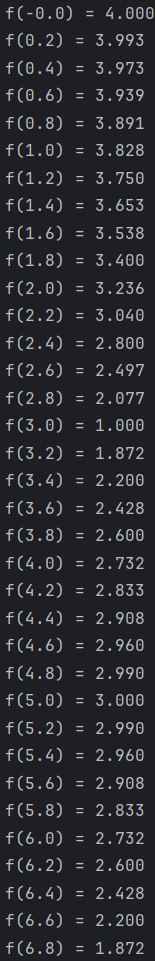
**2.1. Структурное (императивное) программирование**

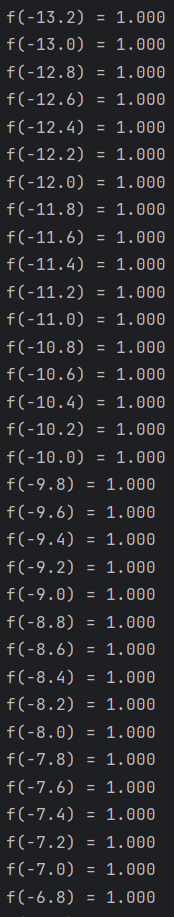
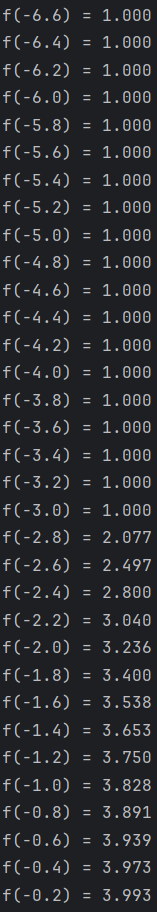
**2.1.1. Программирование без пользовательских функций (№5a)**

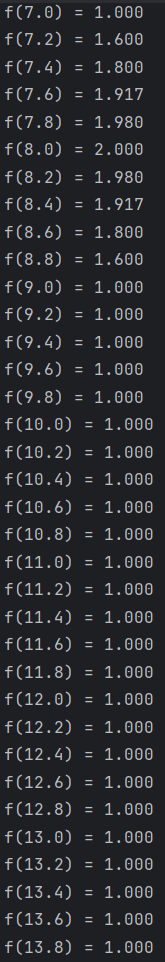
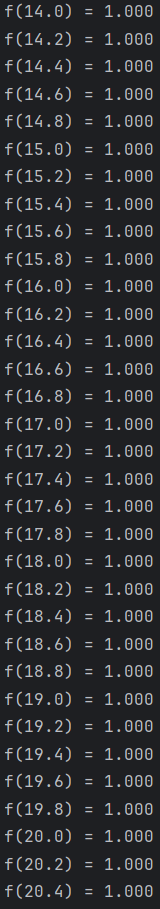
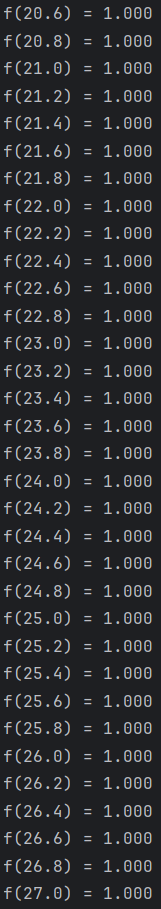
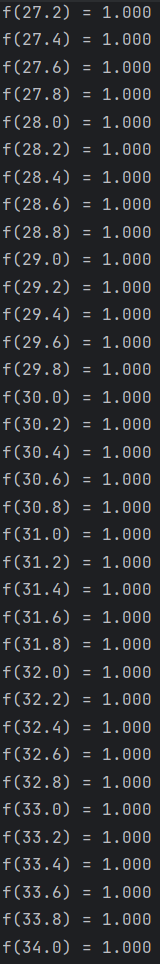
**2.1.1.1. Используемые инструкции в программе**

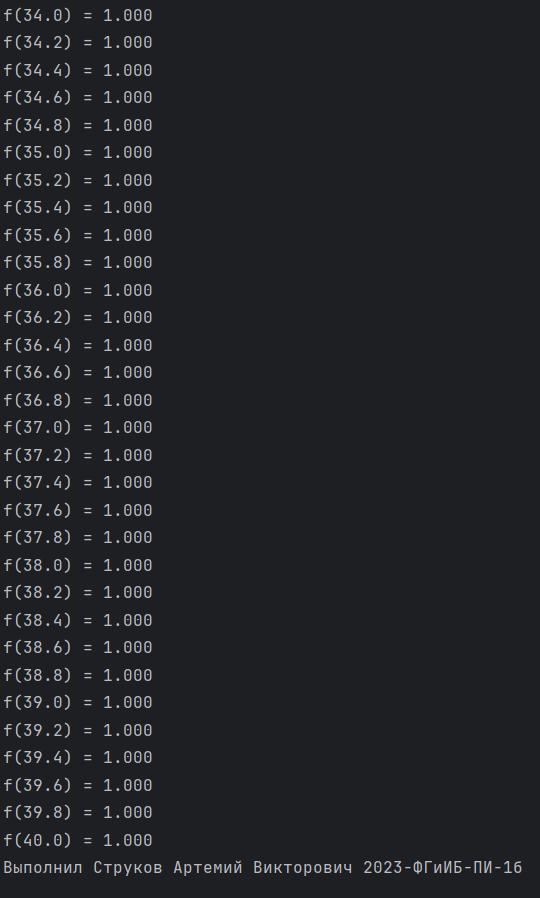
|  |  |  |
| --- | --- | --- |
| **№** | **Инструкция  (из кода программы)** | **Название и назначение инструкции** |
| **1.** | **а = 10** | **Инструкция присваивания. Присваивание переменной a значения целого типа 10 с помощью оператора равенства =** |
| **2.** | **m, n = 0, 1** | **Инструкция присваивания. Присваивание нескольким переменным m, n целочисленных значений, соответственно 0 и 1.** |
| **3.** | **x = float("{0:.3f}".format(x))** | **Отбрасывание до 3 знаков после запятой у переменной x.** |
| **4.** | **-3 <= x <= 3** |  |
| **5.** | **print(f"f({x}) = {y}")** | **Интерполяция строк. Строка помечается символом f в начале, что позволяет встраивать переменные в строку обособляя их фигурными скобками.** |

**2.1.1.2. Блок-схема решения задачи**

**2.1.1.3. Результаты работы программы (скриншоты)**





****

**2.1.1.4. Текст отлаженной программы**

from numpy import arange  
from math import sqrt  
  
for x in arange(-20, 40.1, 0.2):  
 x = float("{0:.3f}".format(x))  
 if -3 <= x <= 3:   
 y = sqrt(9 - x\*\*2) + 1  
 elif 3 < x <= 7:   
 y = sqrt(-21 - x \*\* 2 + 10 \* x) + 1  
 elif 7 < x <= 9:   
 y = sqrt(-63 - x \*\* 2 + 16 \* x) + 1  
 else:  
 y = 1.0  
 y = "{0:.3f}".format(y)  
 print(f"f({x}) = {y}")  
  
print("Выполнил Струков Артемий Викторович 2023-ФГиИБ-ПИ-1б")

**2.1.2. Программирование с пользовательскими функциями (№5b)**

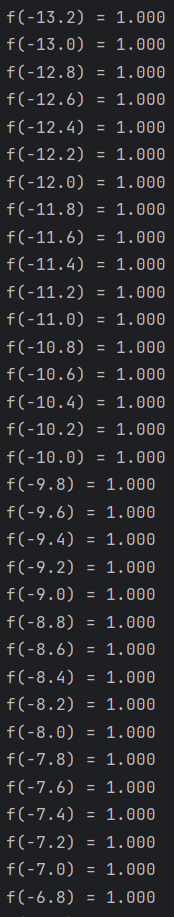
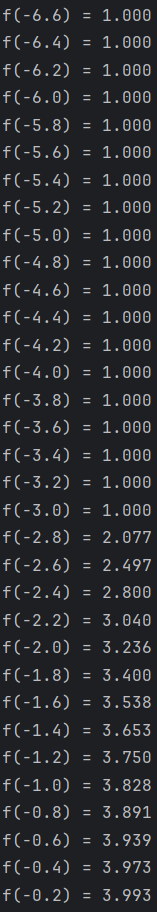
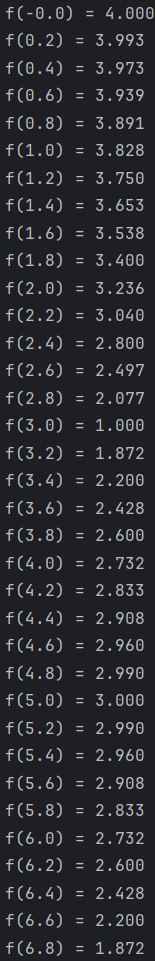
**2.1.2.1. Используемые инструкции в программе**

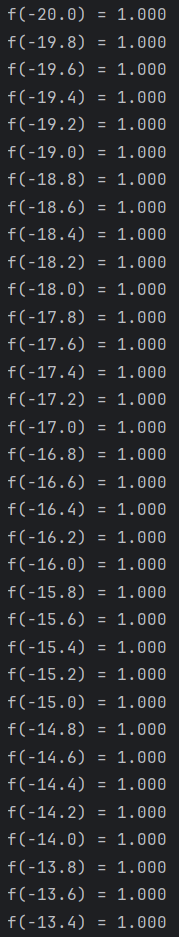
|  |  |  |
| --- | --- | --- |
| **№** | **Инструкция  (из кода программы)** | **Название и назначение инструкции** |
| **1.** | **is\_exist = False** | **Инструкция присваивания. Присваивание переменной is\_exist значения логического типа False с помощью оператора равенства =** |
| **2.** | **randint(-100, 100)** | **Инструкция вызова функции. Выполнение ранее написанного кода выделенного в функцию с передачей входных аргументов через запятую** |
| **3.** | **for i in range(5):** | **Цикл. Цикл с переменной i, которая выполняет роль счетчика, принимает значения от 0 до 5 (не включительно)** |
| **4.** | **if is\_exist:**  **pass**  **else: pass** | **Условная конструкция. Принимает логическое значение is\_exist, если оно верно вызывает тело if, иначе тело else** |
| **5.** | **if \_\_name\_\_ == "\_\_main\_\_":** | **Условие на то, что файл является исполняемым** |
| **6.** | **data = [[randint(-100, 100) for \_ in range(5)] for \_ in range(8)]** | **Двумерное списочное выражение. Возвращает двумерный список размера 8 строк на 5 колонок** |
| **7.** | **def name (n: int, m: int) -> int** | **Декларирование функции с названием name с входными аргументами и указанием их типов, и указанием возвращающегося типа функции.** |
| **8.** | **try:**  **pass**  **except ValueError as e:**  **pass** | **Обработка ошибок. В случае возникновения исключения ValueError в блоке try будет вызван блок except** |
| **9.** | **for x in arange(-20, 40.1, 0.2):** | **Цикл с вещественным счетчиком** |

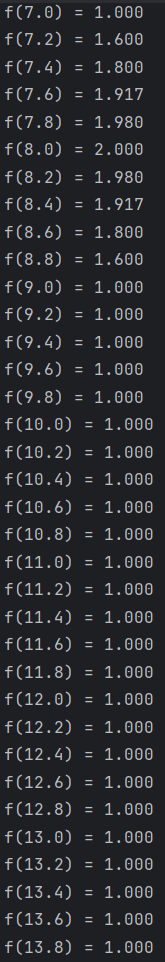
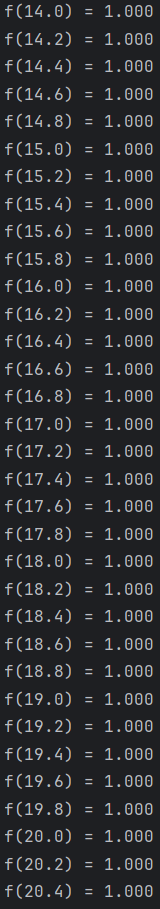
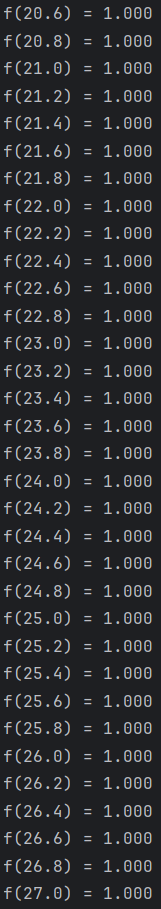
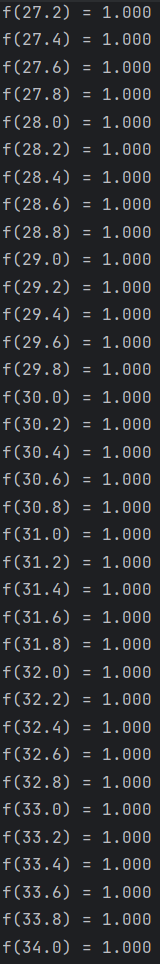
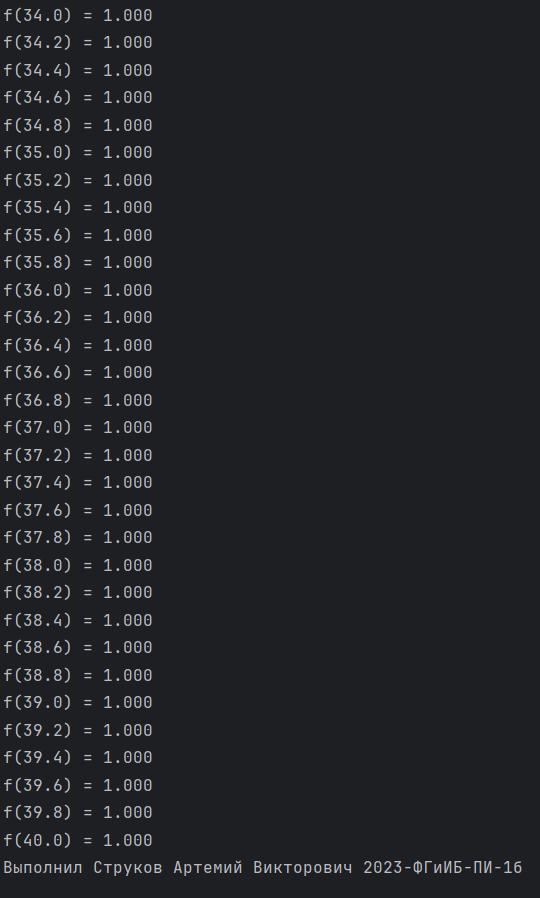
**2.1.2.2. Блок-схема решения задачи**

*Самостоятельно!*

**2.1.2.3. Результаты работы программы (скриншоты)**





****

**2.1.2.4. Текст отлаженной программы**

from numpy import arange  
from math import sqrt  
  
  
def calc\_first\_circle(x: float) -> float:   
 try:  
 return sqrt(9 - x \*\* 2) + 1  
 except ValueError:  
 return 1  
  
  
def calc\_second\_circle(x: float) -> float:   
 try:  
 return sqrt(-21 - x \*\* 2 + 10 \* x) + 1  
 except ValueError:  
 return 1  
  
  
def calc\_third\_circle(x: float) -> float:   
 try:  
 return sqrt(-63 - x \*\* 2 + 16 \* x) + 1  
 except ValueError as e:  
 return 1  
  
  
def pprint(x: float, y: float):  
 y = "{0:.3f}".format(y)  
 print(f"f({x}) = {y}")  
  
  
def main():  
 for x in arange(-20, 40.1, 0.2):  
 x = float("{0:.3f}".format(x))  
 if -3 <= x <= 3:  
 y = calc\_third\_circle(x)  
 elif 3 < x <= 7:  
 y = calc\_second\_circle(x)  
 elif 7 < x <= 9:  
 y = calc\_first\_circle(x)  
 else:  
 y = 1.0  
 pprint(x, y)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()  
  
print("Выполнил Струков Артемий Викторович 2023-ФГиИБ-ПИ-1б")

**2.2.1. Программирование с пользовательскими функциями (№5с)**

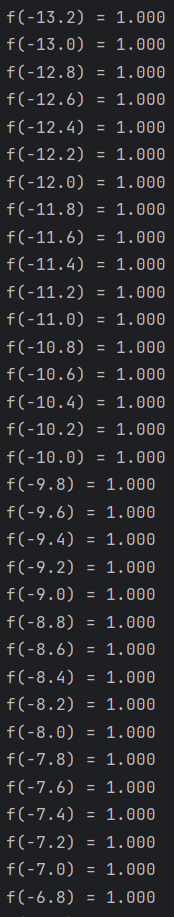
**2.2.1.1. Используемые инструкции в программе**

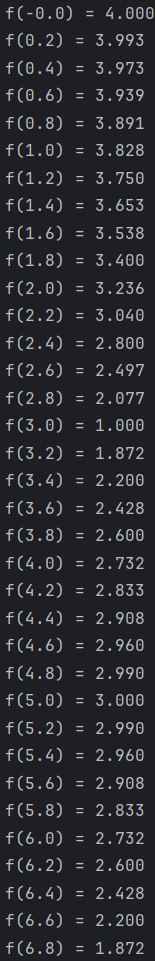
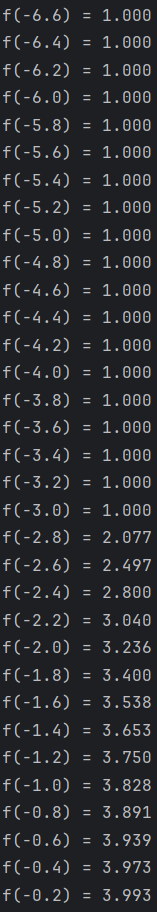
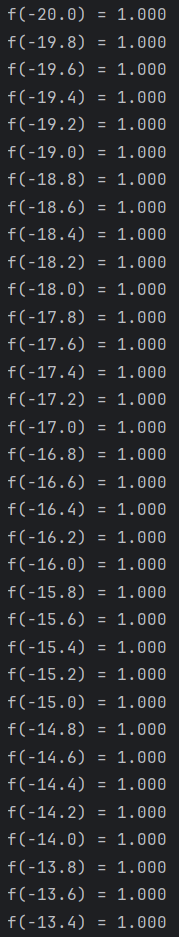
|  |  |  |
| --- | --- | --- |
| **№** | **Инструкция  (из кода программы)** | **Название и назначение инструкции** |
| **1.** | **is\_exist = False** | **Инструкция присваивания. Присваивание переменной is\_exist значения логического типа False с помощью оператора равенства =** |
| **2.** | **randint(-100, 100)** | **Инструкция вызова функции. Выполнение ранее написанного кода выделенного в функцию с передачей входных аргументов через запятую** |
| **3.** | **for i in range(5):** | **Цикл. Цикл с переменной i, которая выполняет роль счетчика, принимает значения от 0 до 5 (не включительно)** |
| **4.** | **if is\_exist:**  **pass**  **else: pass** | **Условная конструкция. Принимает логическое значение is\_exist, если оно верно вызывает тело if, иначе тело else** |
| **5.** | **if \_\_name\_\_ == "\_\_main\_\_":** | **Условие на то, что файл является исполняемым** |
| **6.** | **data = [[randint(-100, 100) for \_ in range(5)] for \_ in range(8)]** | **Двумерное списочное выражение. Возвращает**  **двумерный список размера 8 строк на 5 колонок** |
| **7.** | **def name (self, n: int, m: int) -> int** | **Декларирование метода класса (self) с названием name с входными аргументами и указанием их типов, и указанием возвращающегося типа функции.** |
| **8.** | **class Calculator():** | **Объявление класса Calculator** |
| **9.** | **for x in arange(-20, 40.1, 0.2):** | **Цикл с вещественным счетчиком** |
| **10.** | **@staticmethod**  **def run():**  **Main().main()** | **Статический метод класса. Статический метод можно вызывать без создания экземпляра класса** |

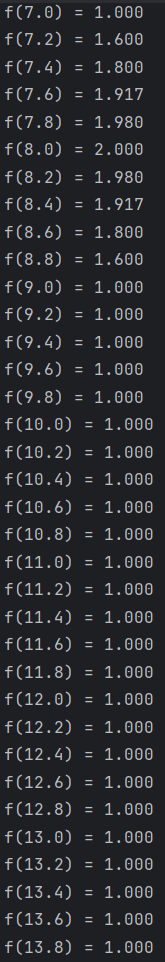
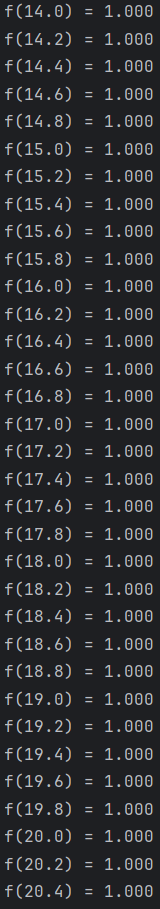
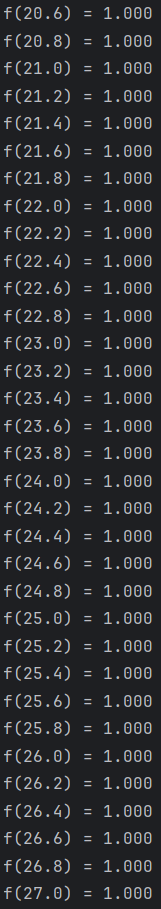
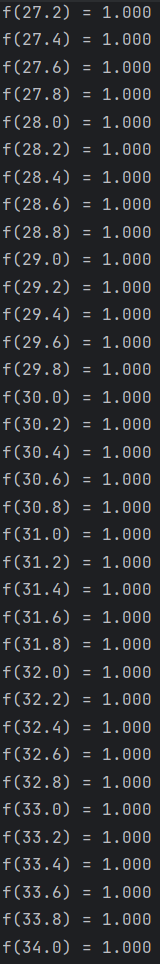
**2.2.1.2. Блок-схема решения задачи**

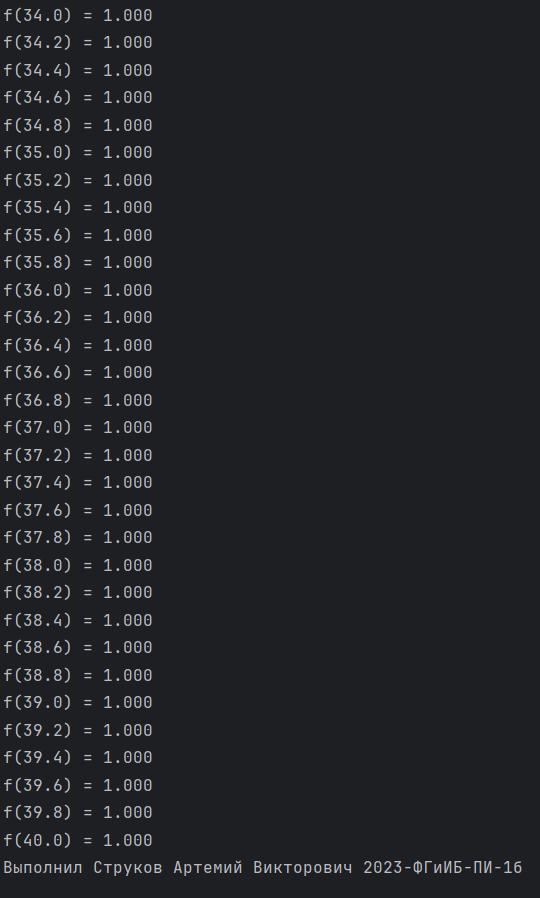
*Самостоятельно!*

**2.2.1.3. Результаты работы программы (скриншоты)**

****

****

****

****

**2.2.1.4. Текст отлаженной программы**

class Calculator():  
 def \_\_init\_\_(self, x: float):  
 self.x = x  
  
 def calc\_first\_circle(self) -> float:   
 return sqrt(9 - self.x \*\* 2) + 1  
  
 def calc\_second\_circle(self) -> float:   
 return sqrt(-21 - self.x \*\* 2 + 10 \* self.x) + 1  
  
 def calc\_third\_circle(self) -> float:   
 return sqrt(-63 - self.x \*\* 2 + 16 \* self.x) + 1  
  
 def calc(self) -> float:  
 if -3 <= self.x <= 3:  
 return self.calc\_third\_circle()  
 if 3 < self.x <= 7:  
 return self.calc\_second\_circle()  
 if 7 < self.x <= 9:  
 return self.calc\_third\_circle()  
 return 1.0  
  
  
class Main():  
 def \_\_init\_\_(self):  
 self.x = None  
 self.y = None  
  
 def pprint(self):  
 y = "{0:.3f}".format(self.y)  
 print(f"f({self.x}) = {y}")  
  
 def main(self):  
 for x in arange(-20, 40.1, 0.2):  
 self.x = float("{:.3f}".format(x))  
 self.y = float("{:.3f}".format(Calculator(self.x).calc()))  
 self.pprint()  
  
 @staticmethod  
 def run():  
 Main().main()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 Main.run()  
  
print("Выполнил Струков Артемий Викторович 2023-ФГиИБ-ПИ-1б")

**3. ПОСТАНОВКА ЗАДАЧИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ №4 (ВАРИАНТ 25)**

Написать программу на Python с использованием пользовательских функций согласно конкретным формулировкам,

указанным перед каждым из 5 индивидуальных заданий Варианта.

Задание 1. Укажите числа, равные 1111\_2 (проставьте галочки и обязательно приведите вывод):

[] 11\_4 (11 в 4 системе счисления)

[] 11\_10

[] F\_16

[] 17\_8

Задание 2. Вычислите разность (обязательно приведите вывод)

1A\_16 - 6\_16 = []\_16

Задание 3. Восстановите пропущенный знак операции сравнения (больше, меньше, больше или равно, меньше или равно)

и обязательно приведите вывод:

F\_16 [] 16\_8

Задание 4. Расположите числа в порядке возрастания (обязательно приведите вывод):

F1\_16; 12\_8; 12\_10; F\_16

Задание 5. Впишите пропущенное число (обязательно приведите вывод):

2BF8\_16 = []\_2

**3.2. Объектно-ориентированное программирование**

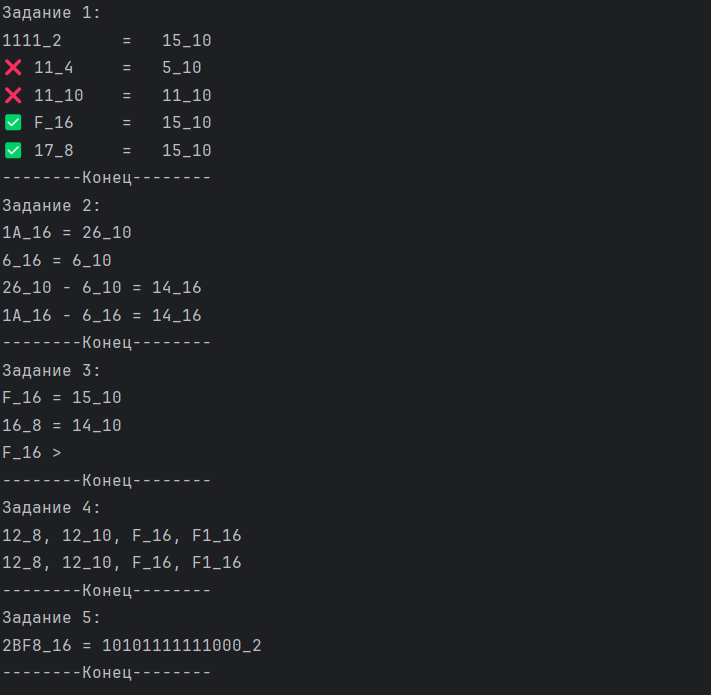
**3.2.1. Используемые инструкции в программе**

|  |  |  |
| --- | --- | --- |
| **№** | **Инструкция  (из кода программы)** | **Название и назначение инструкции** |
| **1.** | **is\_exist = False** | **Инструкция присваивания. Присваивание переменной is\_exist значения логического типа False с помощью оператора равенства =** |
| **2.** | **bin(self.value)[2:]** | **Приведение числа в десятичной системе счисления к двоичной. Возвращает строку.** |
| **3.** | **for i in range(5):** | **Цикл. Цикл с переменной i, которая выполняет роль счетчика, принимает значения от 0 до 5 (не включительно)** |
| **4.** | **if is\_exist:**  **pass**  **else: pass** | **Условная конструкция. Принимает логическое значение is\_exist, если оно верно вызывает тело if, иначе тело else** |
| **5.** | **if \_\_name\_\_ == "\_\_main\_\_":** | **Условие на то, что файл является исполняемым** |
| **6.** | **isinstance(other, Num)** | **Проверка на то, что переменная other является экземпляром класса Num** |
| **7.** | **def name (self, n: int, m: int) -> int** | **Декларирование метода класса (self) с названием name с входными аргументами и указанием их типов, и указанием возвращающегося типа функции.** |
| **8.** | **class Num():** | **Объявление класса Num** |
| **9.** | **for x in arange(-20, 40.1, 0.2):** | **Цикл с вещественным счетчиком** |
| **10.** | **raise Exception("Num сравнивается только с Num!")** | **Вызов исключения с текстовой ошибкой** |

**3.2.2. Блок-схема решения задачи**

*Самостоятельно!*

**3.2.3. Результаты работы программы (скриншоты)**

****

**3.2.4. Текст отлаженной программы**

class Num():  
 def \_\_init\_\_(self, value: int | str, base: int = 10):  
 self.value, self.base = value, base  
  
 def set\_decimal(self):  
 if self.base == 10:  
 return  
 self.value = int(self.value, self.base)  
 self.base = 10  
  
 def set\_hex(self):  
 if self.base == 16:  
 return  
 if self.base != 10:  
 self.set\_decimal()  
 self.value = hex(self.value)[2:].upper()  
 self.base = 16  
  
 def set\_bin(self):  
 if self.base != 10:  
 self.set\_decimal()  
 self.value = bin(self.value)[2:]  
 self.base = 2  
  
 def \_\_eq\_\_(self, other) -> bool:  
 if isinstance(other, Num):  
 return self.value == other.value and self.base == other.base  
 raise Exception("Num сравнивается только с Num!")  
  
 def \_\_lt\_\_(self, other):  
 if isinstance(other, Num):  
 if other.base != self.base:  
 raise Exception("Num'ы сравниваются только когда имеют одну систему счисления")  
 return self.value < other.value  
 raise Exception("Num сравнивается только с Num!")  
  
 def \_\_le\_\_(self, other):  
 return self < other and self == other  
  
 def \_\_gt\_\_(self, other):  
 if isinstance(other, Num):  
 if other.base != self.base:  
 raise Exception("Num'ы сравниваются только когда имеют одну систему счисления")  
 return self.value > other.value  
 raise Exception("Num сравнивается только с Num!")  
  
 def \_\_ge\_\_(self, other):  
 return self > other and self == other  
  
 def \_\_add\_\_(self, other):  
 if isinstance(other, Num):  
 if other.base != self.base:  
 raise Exception(  
 "Арифметические действия с Num осуществляются только когда они имеют одну систему счисления"  
 )  
 return Num(self.value + other.value, self.base)  
 raise Exception("Арифметические действия с Num возможно только с другим Num")  
  
 def \_\_neg\_\_(self):  
 return Num(self.value \* (-1), self.base)  
  
 def \_\_sub\_\_(self, other):  
 return self + (-other)  
  
 def \_\_str\_\_(self):  
 return f"{self.value}\_{self.base}"  
  
  
class Task():  
 def run(self):  
 pass  
  
  
class Task1(Task):  
 def run(self):  
 print("Задание 1:")  
 target = Num("1111", 2)  
 target\_str = str(target)  
 target.set\_decimal()  
 print(f"{target\_str}\t\t=\t{target}")  
  
 def get\_check(value: Num) -> str:  
 return "✅" if value == target else "❌"  
  
 first = Num("11", 4)  
 first.set\_decimal()  
 print(f"{get\_check(first)} 11\_4\t\t=\t{first}")  
  
 second = Num("11", 10)  
 second.set\_decimal()  
 print(f"{get\_check(second)} 11\_10\t=\t{second}")  
  
 third = Num("F", 16)  
 third.set\_decimal()  
 print(f"{get\_check(third)} F\_16\t\t=\t{third}")  
  
 four = Num("17", 8)  
 four.set\_decimal()  
 print(f"{get\_check(four)} 17\_8\t\t=\t{four}")  
  
 print("-" \* 8 + "Конец" + "-" \* 8)  
  
  
class Task2(Task):  
 def run(self):  
 print("Задание 2:")  
 first = Num("1A", 16)  
 first.set\_decimal()  
 print(f"1A\_16 = {first}")  
 second = Num("6", 16)  
 second.set\_decimal()  
 print(f"6\_16 = {second}")  
 result = first - second  
 result.set\_hex()  
 print(f"{first} - {second} = {result}")  
 print(f"1A\_16 - 6\_16 = {result}")  
 print("-" \* 8 + "Конец" + "-" \* 8)  
  
  
class Task3(Task):  
 def run(self):  
 print("Задание 3:")  
 first = Num("F", 16)  
 first.set\_decimal()  
 second = Num("16", 8)  
 second.set\_decimal()  
 if first > second:  
 operation = ">"  
 elif first < second:  
 operation = "<"  
 else:  
 operation = "="  
 print(f"F\_16 = {first}")  
 print(f"16\_8 = {second}")  
 print(f"F\_16 {operation}")  
 print("-" \* 8 + "Конец" + "-" \* 8)  
  
  
class Task4(Task):  
 def run(self):  
 print("Задание 4:")  
 first = Num("F1", 16)  
 second = Num("12", 8)  
 third = Num("12", 10)  
 four = Num("F", 16)  
 data = sorted([first, second, third, four], key=lambda num: num.value)  
 print(", ".join([str(i) for i in data]))  
 print(", ".join(map(str, data)))  
 print("-" \* 8 + "Конец" + "-" \* 8)  
  
  
class Task5(Task):  
 def run(self):  
 print("Задание 5:")  
 value = Num("2BF8", 16)  
 value.set\_bin()  
 print(f"2BF8\_16 = {value}")  
 print("-" \* 8 + "Конец" + "-" \* 8)  
  
  
def main():  
 Task1().run()  
 Task2().run()  
 Task3().run()  
 Task4().run()  
 Task5().run()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**3.3. Объектно-ориентированное программирование**

**3.3.1. Описание алгоритма вычисления определенного интеграла   
по методу средних прямоугольников**

*[скопируете текст по теории из методических указаний].*

**3.3.2. Разработка пользовательского интерфейса программы с использованием виджетов PyQt**

*[приведите пошаговые скрины разработки своего интерфейса в среде Qt Designer].*

**3.3.3. Используемые библиотеки в программе**

*[приведите таблицу из 2-х столбцов, где в левой части дайте название библиотеки, справа – ее назначение;   
самостоятельно разберитесь в назначении!].*

**3.3.4. Результаты работы программы (скриншоты)**

*[приведите скрины своей программы – при ее запуске с различными видами: начальный вид, при вводе значений и результата расчетов и окно с авторством].*

**3.3.5. Текст отлаженной программы**

*[копируете весь текст программного кода из окна редактора в этот раздел.*

*Шрифт для кода программы –* ***12 pt****,* ***Arial*** *и через* ***1*** *интервал]*

**ВЫВОДЫ**

1.

2.

3.

*[ здесь вы должны написать кратко:*

*1. какие задания вами были выполнены,*

*2. кратко – какие ошибки были обнаружены и отлажены и*

*3. дать заключительную фразу о том, какой вид программирования вам больше всего и почему понравился]*

**СПИСОК ИСПОЛЬЗОВАННЫХ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ**

1.

2.

3.

*[можно указать электронные цифровые книги и интернет-ресурсы, которыми пользовались]*