МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

Методические указания к практическим занятиям

для направления подготовки:

09.03.02 «Информационные системы и технологии»,

09.03.04 «Программная инженерия»

Квалификация выпускника: бакалавр

Ставрополь, 2022

Н 63 **Информационные технологии и программирование:** методические указания к практическим занятиям для студентов направления 09.03.02 «Информационные системы и технологии», 09.04.02 «Программная инженерия» / Николаев Е.И. – Ставрополь: Изд-во СКФУ, 2022. – 70 с.

Автор:

канд. техн. наук, доцент **Е.И. Николаев**

ОГЛАВЛЕНИЕ

Лабораторная работа 1. Простейший скрипт Python 4
Лабораторная работа 2. Основные конструкции языка 8
Лабораторная работа 3. Обработка данных из файла 12
Лабораторная работа 4. Данные в формате CSV 17
Лабораторная работа 5. Объектно-ориентированное программирование 26
Лабораторная работа 6. Работа с изображениями. Библиотека Pillow 33
Лабораторная работа 7. Основы Numpy. Построение графиков 35
Лабораторная работа 8. Основы программирования на C# 37
Лабораторная работа 9. Двумерная графика в оконных приложениях 53
Список литературы 68

ЛАБОРАТОРНАЯ РАБОТА 1. ПРОСТЕЙШИЙ СКРИПТ PYTHON

1. Цель и содержание

Цель лабораторной работы: научиться писать простейшие скрипты Python, используя простейшие возможности языка.

Задачи лабораторной работы:

– применение функции type( ),

– применение функции print( ),

– выполнение скрипта и решение простейших задач.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Задание: напишите скрипт на языке Python для вычисления значения функции:

$$Z = \frac{x^y - \sqrt[3]{x}}{a \cdot (k_to + 120 \cdot |x - y|)}$$

Для каждой используемой переменной необходимо вывести ее тип.

Решение

https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_1.ipynb

2.2. Индивидуальное задание

Индивидуальное задание состоит из двух частей, которые выполняются в одном блокноте Python Notebook:

1. Измените скрипт, созданный в ходе выполнения данной практической работы таким образом, чтобы в блокноте Python Notebook выводилась следующая информация (каждый студент должен использовать персональную информацию о себе):

– ФИО студента;

– группа студента и шифр специальности;

– перечень языков программирования и ИТ-технологий, которые студент считает наиболее перспективными для изучения;

– краткое описание личных увлечений, хобби, интересов.

2. Объявите требуемые переменные, присвойте им начальные значения (определите самостоятельно, значения какого типа могут принимать переменные), выведите на экран с использованием форматной строки значения переменных и результат вычисления выражения в соответствие с вариантом:

Вариант	Выражение для вычисления
1	$F = a_1 + b - a \cdot (x + y5)$
2	$Se = w111 + bt - x + y \cdot w$
3	$R_x = a \cdot b + b / t - x + f \cdot i_2$
4	$c = gh + b \cdot q3 - x + y / w$
5	$H = \frac{g \cdot h}{d17} + b / h1 - \frac{x + y}{4}$
6	$Z = \frac{35}{f} + y \cdot f - \frac{f + y}{4}$
7	$U = \frac{35}{a} \cdot z + z \cdot a - \frac{a + E \cdot t}{4}$
8	$A0 = \frac{35}{G_1} \cdot Zvcw + G_1 \cdot a - \frac{a + Zvcw}{a}$
9	$Se = w111 + bt - x + y \cdot w$
10	$R_x = a \cdot b + b / t - x + f \cdot i_2$
11	$g = w \cdot h + b \cdot q3 - q3 + y / w$
12	$ee = \frac{g \cdot h}{d17} + d17 / h - \frac{g + d17 + h}{4}$
13	$Zze = \frac{35}{f} + y \cdot f - \frac{f + y}{4}$
14	$t = \frac{35}{a} \cdot z + z \cdot a - \frac{a + Et}{4}$
15	$A0 = \frac{35}{G_1} \cdot Zvcw + G_1 \cdot a - \frac{a + Zvcw}{a}$
16	$Zxy = a_5 + b - a_5 \cdot (b + y)$

17	$$ch = \frac{6}{f1} \cdot z + z \cdot f1 - \frac{f1 + Et}{6}$$
18	$$ch = rx \cdot z + z \cdot f1 - \frac{f1 + Et}{rx}$$
19	$$_H = k1 + k2 - k3 + \frac{k1 \cdot k2}{k3}$$
20	$$Fy = a_1 + b - a_1 \cdot (x + y)$$
21	$$_G_1 = y1 + y2 + y3 - \frac{y1 + y2 + y3}{3}$$
22	$$R = 2 \cdot x + 2 \cdot y - 4 \cdot x \cdot y + z$$
23	$$g11 = t \cdot z + z \cdot f1 - \frac{f1 + U}{t}$$
24	$$Y = Z \cdot (z + x) + z / Z - \frac{z + Z}{x + X}$$
25	$$U = rx \cdot z + z \cdot Y - \frac{Y + z}{rx}$$
26	$$c = gh + b \cdot q3 - x + y / w$$
27	$$H = \frac{g \cdot h}{d17} + b / h1 - \frac{x + y}{4}$$
28	$$Z = \frac{35}{f} + y \cdot f - \frac{f + y}{4}$$
29	$$W_1 = rx \cdot z + z \cdot (rx + z) - \frac{Y + z}{rx}$$
30	$$A0 = \frac{35}{G_1} \cdot Zvcw + G_1 \cdot a - \frac{a + Zvcw}{a}$$
31	$$Se = w111 + bt - x + y \cdot w$$
32	$$R_x = 15 + a \cdot b + b / t - x + f \cdot i_2$$
33	$$g = w \cdot h + b \cdot q3 - q3 + y / w$$
34	$$ee = \frac{g \cdot h}{d17} + d17 / h - \frac{g + d17 + h}{4}$$
35	$$Pwd = \frac{12 + f}{f} + y \cdot f - \frac{f + y}{4}$$
36	$$t = \frac{35}{a} \cdot z + z \cdot a - \frac{a + Et}{4}$$
37	$$Se = w111 + bt - x + y \cdot w$$
38	$$R_x = a \cdot b + b / t - x + f \cdot i_2$$
39	$$t = xx \cdot h + b \cdot q3 - q3 + y / xx$$

40	$X = \frac{g \cdot h}{d17} + d17 / h - \frac{g + d17 + h}{4}$

3. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Чем отличается интерпретируемый язык программирования от компилируемого?

2. Перечислите компилируемые языки программирования.

3. Перечислите интерпретируемые языки программирования.

4. Как объявляется переменные в языке Python?

5. Каким образом определить тип переменной в языке Python?

6. Для чего используется функция **`print( )`** ?

7. Какая функция предназначена для вычисления модуля в Python? Какую библиотеку необходимо для этого подключить?

8. Какие из перечисленных идентификаторов нельзя использовать в качестве имен пользовательских переменных?

`_1_01, b100, int, double_1, _b200, MyVar, create-var, 4perem, _5elem, zo0, wodoo, UserCount, system_call, string, System.Double.`

9. Перечислите основные типы в Python.

ЛАБОРАТОРНАЯ РАБОТА 2. ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА

1. Цель и содержание

Цель лабораторной работы: создание скриптов с использованием базовых возможностей языка.

Задачи лабораторной работы: типы данных, числа, строки, операции условия и цикла, список, применение функции print( ), выполнение скрипта и решение простейших задач.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Задание: напишите скрипт поиска чисел, кратных 23 в списке, состоящем из элементов произвольного типа.

Решение

https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_2.ipynb

2.2. Индивидуальное задание

Напишите скрипт для решения задачи в блокноте Python Notebook в соответствии с индивидуальным вариантом:

Вариант	Выражение для вычисления
1	Напишите скрипт поиска чисел, кратных 7, но не кратных 13 в списке, состоящем из элементов произвольного типа.
2	Напишите скрипт для подсчета количества элементов не целочисленного типа в списке, состоящем из элементов произвольного типа.
3	Напишите скрипт для подсчета количества элементов списочного типа в списке, состоящем из элементов произвольного типа.
4	Напишите скрипт для подсчета суммы элементов целочисленного типа в списке, состоящем из элементов произвольного типа.
5	Напишите скрипт для подсчета среднего арифметического элементов целочисленного типа в списке, состоящем из элементов произвольного типа.

6	Напишите скрипт для поиска максимального из элементов вещественного типа в списке, состоящем из элементов произвольного типа.
7	Напишите скрипт для поиска среднего арифметического элементов вещественного типа в списке, состоящем из элементов произвольного типа.
8	Напишите скрипт для поиска среднего геометрического элементов вещественного типа в списке, состоящем из элементов произвольного типа.
9	напишите скрипт поиска чисел, кратных 4, но не кратных 150 в списке, состоящем из элементов произвольного типа.
10	Напишите скрипт для подсчета количества строковых элементов в списке, состоящем из элементов произвольного типа.
11	Напишите скрипт для подсчета количества пар рядом стоящих четных целочисленных элементов в списке, состоящем из элементов произвольного типа.
12	Напишите скрипт для подсчета количества троек рядом стоящих нечетных целочисленных элементов в списке, состоящем из элементов произвольного типа.
13	Напишите скрипт для подсчета количества троек рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа. В каждой тройке средний элемент должен быть больше крайних.
14	Напишите скрипт для подсчета количества пар рядом стоящих целочисленных элементов в списке, состоящем из элементов произвольного типа. В каждой паре второй элемент должен быть больше первого.
15	Напишите скрипт поиска чисел, кратных 7, но не кратных 5 в списке, состоящем из элементов произвольного типа.
16	Напишите скрипт для подсчета количества комплексных элементов в списке, состоящем из элементов произвольного типа.
17	Напишите скрипт для подсчета количества пар рядом стоящих нечетных целочисленных элементов в списке, состоящем из элементов произвольного типа.
18	Напишите скрипт для подсчета количества троек рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа.
19	Напишите скрипт для подсчета количества пар рядом стоящих целочисленных элементов в списке, состоящем из элементов произвольного типа. В каждой паре второй элемент не превышает первого.
20	Напишите скрипт для подсчета количества пар рядом стоящих четных целочисленных элементов в списке, состоящем из элементов

	произвольного типа. В каждой паре элементы должны иметь различную четность.
21	Напишите скрипт для подсчета количества комплексных элементов, справа и слева от которых расположены элементы строкового типа, в списке, состоящем из элементов произвольного типа.
22	Напишите скрипт для поиска вещественных элементов, больших 10 в списке, состоящем из элементов произвольного типа. Необходимо вывести количество таких элементов и максимальный из них.
23	Напишите скрипт для подсчета количества троек рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа. В каждой тройке средний элемент должен быть больше 10, а два соседних меньше 10.
24	Напишите скрипт поиска чисел, кратных 2, но не кратных 4 в списке, состоящем из элементов произвольного типа. Необходимо вывести количество и максимальный из найденных элементов.
25	Напишите скрипт для подсчета количества пар рядом стоящих четных целочисленных элементов в списке, состоящем из элементов произвольного типа. В каждой паре элементы должны иметь одинаковую четность.
26	Напишите скрипт для поиска целочисленных элементов, больших 150 в списке, состоящем из элементов произвольного типа. Необходимо вывести количество таких элементов и минимальный из них.
27	Напишите скрипт для поиска вещественных элементов, больших 10 и кратных 4 в списке, состоящем из элементов произвольного типа. Необходимо вывести количество таких элементов и максимальный из них.
28	Напишите скрипт для вычисления суммы нечетных целочисленных элементов, расположенных на четных индексах, в списке, состоящем из элементов произвольного типа.
29	Напишите скрипт поиска чисел, кратных 3, но не кратных 9 в списке, состоящем из элементов произвольного типа. Необходимо вывести количество и минимальный из найденных элементов.
30	Напишите скрипт для подсчета количества троек рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа. В каждой тройке любые соседние элементы должны иметь различную четность.
31	Напишите скрипт для подсчета количества троек рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа. В каждой тройке средний элемент должен быть больше 30, а два соседних меньше 10.
32	Напишите скрипт для подсчета количества комплексных элементов, справа и слева от которых расположены элементы целочисленного типа, в списке, состоящем из элементов произвольного типа.

33	Напишите скрипт для подсчета количества троек рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа. Все элементы в тройке должны иметь одинаковую четность.
34	Напишите скрипт для вычисления суммы четных целочисленных элементов не кратных 16, в списке, состоящем из элементов произвольного типа.
35	Напишите скрипт для подсчета количества целочисленных элементов, стоящих на четных позициях, в списке, состоящем из элементов произвольного типа.
36	Напишите скрипт для вычисления суммы четных целочисленных элементов, расположенных на нечетных индексах, в списке, состоящем из элементов произвольного типа.
37	Напишите скрипт для подсчета количества строковых элементов, справа и слева от которых расположены элементы не строкового типа, в списке, состоящем из элементов произвольного типа.
38	Напишите скрипт для подсчета количества пар рядом стоящих вещественных элементов в списке, состоящем из элементов произвольного типа. В каждой паре хотя бы один из элементов должен быть больше 11.
39	Напишите скрипт для поиска вещественных элементов, больших 10 и кратных 13 в списке, состоящем из элементов произвольного типа. Необходимо вывести количество таких элементов и максимальный из них.
40	Напишите скрипт для подсчета количества троек рядом стоящих целочисленных элементов в списке, состоящем из элементов произвольного типа. В каждой тройке хотя бы один из элементов должен быть кратен 5.

3. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Какой оператор предназначен для многократного повторения блока кода?
2. Перечислите операторы, применяемые в работе скрипта из учебной задачи.
3. Перечислите параметры функции **`print( )`**. Опишите их назначение.
4. Что такое список в языке Python?
5. Объясните механизм обращения к элементам списка.
6. Поясните чем список отличается от массива (например в C++).

ЛАБОРАТОРНАЯ РАБОТА 3. ОБРАБОТКА ДАННЫХ ИЗ ФАЙЛА

1. Цель и содержание

Цель лабораторной работы: научиться обрабатывать данные из текстового файла.

Задачи лабораторной работы:

– научиться считывать символьные строки из файла;

– научиться преобразовывать строковые данные к прочим типа данных;

– научиться решать прикладные задачи с использованием простых скриптов.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Задание: в файле data_v1.txt содержится последовательность символов, содержащая буквы латинского алфавита и десятичные цифры. Необходимо решить следующие задачи:

1. Найдите максимальное четное число в строке (под числом понимается последовательность цифр, ограниченная буквами).

2. Найдите буквенную подпоследовательность максимальной длины (то есть подпоследовательность не содержащую цифр). В ответе укажите ее длину.

3. Найдите символ в файле, который встречается чаще всего. Если таких символов несколько, то выведите все эти символы, в конце укажите одно число – сколько раз встречаются эти символы.

Решение

https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_3.ipynb

2.2. Индивидуальное задание

Напишите скрипт для решения задачи в блокноте Python Notebook или с использованием локального интерпретатора в соответствии с индивидуальным вариантом:

Вариант	Выражение для вычисления
1	В файле data_v2.txt посчитайте количество цифр, расположенных рядом с символом «Z»
2	В файле data_v2.txt посчитайте количество символов, расположенных между четными цифрами
3	В файле data_v2.txt найдите количество чисел не кратных 3 (под числом понимается непрерывная последовательность цифр). В ответе укажите количество таких чисел и минимальное из них.
4	В файле data_v1.txt найдите количество чисел кратных 3 но не кратных 9 (под числом понимается непрерывная последовательность цифр). В ответе укажите количество таких чисел и максимальное из них.
5	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую заглавных букв и цифр, кратных 3. В ответе укажите саму подпоследовательность и ее длину.
6	В файле data_v2.txt найдите минимальное число кратное 3 (под числом понимается непрерывная последовательность цифр)
7	В файле data_v2.txt найдите количество чисел кратных 10 (под числом понимается непрерывная последовательность цифр). В ответе укажите количество таких чисел и их среднее арифметическое.
8	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую гласных букв, и при этом сумма цифр подпоследовательности должна быть четной. В ответе укажите саму подпоследовательность и ее длину.
9	В файле data_v2.txt найдите самый частотный заглавный символ. В ответе укажите сам символ и количество. Если таких символов несколько укажите все из них.
10	В файле data_v2.txt посчитайте количество символов, расположенных между нечетными цифра
11	В файле data_v2.txt найдите минимальное число не кратное 4 (под числом понимается непрерывная последовательность цифр)
12	В файле data_v2.txt найдите количество чисел кратных 9 (под числом понимается непрерывная последовательность цифр). В ответе укажите количество таких чисел и максимальное из них.

13	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую прописных букв и цифр, кратных 4. В ответе укажите саму подпоследовательность и ее длину.
14	В файле data_v1.txt найдите самый частотный прописной символ. В ответе укажите сам символ и количество. Если таких символов несколько укажите все из них.
15	В файле data_v2.txt посчитайте количество символов, расположенных между заглавными буквами
16	В файле data_v2.txt найдите количество чисел, у которых поровну четных и нечетных цифр (под числом понимается непрерывная последовательность цифр). В ответе укажите количество таких чисел и максимальное из них.
17	В файле data_v2.txt найдите самый частотный символ-цифру. В ответе укажите сам символ и количество. Если таких символов несколько укажите все из них.
18	В файле data_v2.txt найдите самую длинную подпоследовательность не содержащую нечетных цифр и прописных букв. В ответе укажите саму подпоследовательность и ее длину.
19	В файле data_v2.txt найдите минимальное число кратное 20 (под числом понимается непрерывная последовательность цифр)
20	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую прописных гласных букв, и при этом сумма цифр подпоследовательности должна быть четной. В ответе укажите саму подпоследовательность и ее длину.
21	В файле data_v2.txt найдите самую длинную подпоследовательность из букв, в которой заглавных букв больше чем прописных. В ответе укажите саму строку и ее длину.
22	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую букв, и при этом сумма цифр подпоследовательности должна быть четной. В ответе укажите саму подпоследовательность и ее длину.
23	В файле data_v2.txt посчитайте количество символов, расположенных между гласными буквами
24	В файле data_v1.txt найдите самую длинную подпоследовательность состоящую из нечетных цифр. В ответе укажите саму подпоследовательность и ее длину.
25	В файле data_v1.txt найдите количество заглавных букв, стоящих между двумя цифрами с разной четностью.
26	В файле data_v2.txt найдите количество чисел (под числом понимается непрерывная последовательность цифр)
27	В файле data_v2.txt найдите самую длинную подпоследовательность из букв, в которой заглавных букв меньше чем прописных. В ответе укажите саму подпоследовательность и ее длину.

28	В файле data_v1.txt найдите количество прописных букв, стоящих между двумя цифрами кратными 3.
29	В файле data_v2.txt найдите самую длинную подпоследовательность не содержащую цифр кратных 3, 4 или 5 и прописных букв. В ответе укажите саму подпоследовательность и ее длину
30	В файле data_v2.txt найдите самую длинную подпоследовательность не содержащую букв, и при этом сумма цифр подпоследовательности должна быть нечетной. В ответе укажите саму подпоследовательность и ее длину.
31	В файле data_v2.txt посчитайте количество символов, расположенных между согласными буквами
32	В файле data_v1.txt найдите самую длинную подпоследовательность состоящую из четных цифр и(или) прописных букв. В ответе укажите саму подпоследовательность и ее длину.
33	В файле data_v2.txt найдите самую длинную подпоследовательность не содержащую четных цифр и прописных букв. В ответе укажите саму подпоследовательность и ее длину.
34	В файле data_v2.txt посчитайте количество символов, расположенных между прописными буквами
35	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую прописных букв, и при этом сумма цифр подпоследовательности должна быть четной. В ответе укажите саму подпоследовательность и ее длину.
36	В файле data_v1.txt найдите самую длинную подпоследовательность не содержащую заглавных букв. В ответе укажите саму подпоследовательность и ее длину.
37	В файле data_v1.txt найдите самую длинную подпоследовательность состоящую из цифр, кратных 4 или 5 и(или) прописных букв. В ответе укажите саму подпоследовательность и ее длину.
38	файле data_v2.txt найдите самую длинную подпоследовательность состоящую из четных цифр и(или) гласных заглавных букв. В ответе укажите саму подпоследовательность и ее длину.
39	В файле data_v1.txt найдите количество символов, стоящих между двумя цифрами с одинаковой четностью.
40	В файле data_v2.txt найдите самую длинную подпоследовательность не содержащую заглавных букв, и при этом сумма цифр в подпоследовательности должна быть нечетной. В ответе укажите саму подпоследовательность и ее длину.

3. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Что общего у строки и списка в Python?

2. Перечислите и охарактеризуйте функции класса **`str`** в Python.

3. Охарактеризуйте тип данных **`dict`** в Python.

4. Перечислите и опишите использование функций класса **`dict`**.

5. Опишите механизм чтения данных из текстового файла.

6. Опишите назначение, сходства и различия функций чтения из текстового файла: **`read( )`**, **`readline( )`**, **`readlines( )`**.

7. как прочитать многострочный текстовый файл не используя явно методы **`read( )`**, **`readline( )`**, **`readlines( )`**?

ЛАБОРАТОРНАЯ РАБОТА 4. ДАННЫЕ В ФОРМАТЕ CSV

1. Цель и содержание

Цель лабораторной работы: научиться обрабатывать данные из файла csv.

Задачи лабораторной работы:

– научиться считывать строки из файла csv;

– научиться преобразовывать типы данных;

– научиться решать прикладные задачи с использованием простых скриптов.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Задание:

На рисунке 4.1 показан фрагмент файла csv, содержащего информацию о держателях финансовых карт.

1001 lines (1001 sloc) | 90.6 KB — Raw | Blame

Search this file...

	ID	LastName	FirstName	email	Plus	Minus	credit_card	Tax
2	12-9321320	Pickle	Réjane		$41564.80	$46160.22	3557725016917448	$16408.20
3	14-4213773	Hartford	Léa		$46310.29	$13116.21	3544948067779642	$13397.19
4	92-0579703	Caldecott	Adélaïde	lcaldecott2@com.com	$44330.64	$9725.39	4508863396652307	$35768.27
5	43-3066209	Dohrmann	Cloé	odohrmann3@gov.uk	$32271.10	$33467.96	6398709109212517	$3876.95
6	39-1753452	Pes	Zhì	dpes4@trellian.com	$24467.01	$5220.55	3562917312487449	$5559.11
7	98-1640163	Satyford	Pénélope	asatyford5@friendfeed.com	$31883.05	$11529.92	3564171803659243	$5939.26
8	58-9143184	Corballis	Clémentine	scorballis6@mapquest.com	$25340.90	$3241.86	341075109903237	$45625.23
9	87-3594410	Hiseman	Bénédicte	chiseman7@sciencedirect.com	$46081.76	$36954.21	3548420880126084	$19464.18
10	74-7031522	Easton	Mà		$37139.51	$14322.80	3528412919441844	$44003.80

Рисунок 4.1 – Структура файла csv, просмотр через github

В csv присутствуют следующие поля: ID – уникальный ключ записи; LastName, firstName – фамилия и имя держателя карты; email – электронная почта; Plus и Minus – суммы поступлений и списаний по карте соответственно в долларах; credit_card – номер карты; Tax – уплаченный налог.

Необходимо ответить на следующий вопросы:

1. Найдите держателя карты с максимальной прибылью. В ответе необходимо указать ФИО держателя и максимальную прибыль (прибыль = поступления – списания).

2. Найдите количество держателей карт, у которых сумма уплаченного налога больше среднего арифметического значения налога по всей выборке.

Решение

https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_4.ipynb

2.2. Индивидуальное задание

Напишите скрипт для решения задачи в блокноте Python Notebook или с использованием локального интерпретатора в соответствии с индивидуальным вариантом.

В каждом варианте необходимо использовать свой набор данных. Файлы с данными необходимо получить у ведущего преподавателя или скачать из репозитория github:

https://github.com/enikolaev/IT-and-Programming/tree/main/data/pract4

Например, для варианта 1 используется файл p4_data_01.csv, для варианта 2 – p4_data_02.csv, и так далее.

Вариант	Задание
1	1. Количество сотрудников в каждой из 3-х категорий; 2. Суммарный фонд з/п и сумму по каждой категории; 3. Количество сотрудников с дипломом; 4. Максимальную зарплату среди сотрудников без диплома.
2	1. Количество продуктов в каждой из 3-х категорий; 2. Суммарный объем поставок в валюте по каждой категории; 3. Суммарный объем в валюте всей выборки; 4. Количество просроченных продуктов.
3	1. Самый дорогой продукт для каждого из 4-х поставщиков; 2. Товар с минимальным количеством в каждой категории; 3. Объем просрочки в валюте; 4. Среднюю цену у каждого поставщика.

4	1. Количество сотрудников в каждой из 4-х компаний; 2. Среднюю з/п в каждой из 4-х компаний; 3. Количество сотрудников, владеющих английским языком; 4. Количество сотрудников, получающих более $3000,00.
5	1. Количество билетов с пунктом отправления из Ставрополя; 2. Средняя цена билета для каждого их четырех пунктов отправления; 3. Количество билетов, купленных женщинами и мужчинами; 4. Количество билетов с одинаковыми пунктами отправления и прибытия.
6	Дан список целей, которые описываются параметрами. 1. Определите среднее значение Param1 по каждой из категорий; 2. Определите сумму параметров Param2 для каждой категории; 3. Сколько целей снабжены оружием, а сколько – нет; 4. Найти объект с максимальным значением Param1.
7	Дан список сотрудников. 1. Количество сотрудников без логина; 2. Средняя заработная плата по каждому из 4-х отделов; 3. Максимальная з/п сотрудников у которых есть собака или кошка; 4. Средняя з/п у сотрудников с оружием и без.
8	Дан список доменов. Требуется определить: 1. Количество доменов для которых не указан domain или user; 2. Средняя цена домена по каждой из 4-х категорий; 3. Максимальная цена домена среди продающихся и не продающихся; 4. Суммарная стоимость доменов без доменного имени.
9	В файле содержатся сведения о сотрудниках компании, их зарплатах и сумме налога с зарплаты. 1. У какого сотрудника минимальная з/п? Какой сотрудник получает минимальную сумму после уплаты налогов? 2. У скольких сотрудников не указан навык (skill)? 3. Определите среднюю з/п для сотрудников с e-mail и без. 4. Определите сумму налога по всей выборке.
10	В файле содержатся данные о счетах клиентов банка, которые отражают приход и расход по счету, а также сумму налога, которую следует оплатить. 1. Сколько клиентов имеют отрицательный баланс счета, т.е. приход меньше расхода? 2. Вычислите среднюю сумму налогов по счетам. 3. Найдите клиента с максимальными расходами. 4. Вычислите средний доход по счету для клиентов без e-mail.

11	В файле содержатся сведения о товарах на складе (ID, наименование, поставщик, цена, количество, категория, скидка в %). 1. Определите количество товаров со скидкой более 25%. 2. Вычислите суммарный объем (в валюте) товаров, для которых не указа поставщик. 3. Для каждого из 4-х поставщиков определите среднюю цену товаров. 4. Найдите минимальную цену товара с учетом скидки.
12	В файле содержатся сведения о производстве товаров работниками (ID, сотрудник, категория товара, з/п рабочего, количество произведенных товаров, цена за единицу товара). 1. Определите сколько рабочих получают больше, чем вырабатывают продукции. 2. Суммарный объем произведенной продукции (в валюте) без категории. 3. Суммарный объем в валюте произведенной продукции по каждой из 4-х категорий. 4. Самый эффективный сотрудник: с максимальной разницей между произведенным продуктом и зарплатой.
13	В файле содержатся сведения о сотрудниках компании, их зарплатах и сумме налога с зарплаты. 1. Максимальная и минимальная з/п. Вывести соответствующих сотрудников. 2. Средняя з/п сотрудников без указанного навыка (skill)? 3. Количество сотрудников без e-mail. 4. Суммарный налог.
14	В файле содержатся данные о товарах (ID, наименование, категория (4 шт.), количество, цена, признак просрочки, признак наличия скидки). 1. Средняя цена товаров для просроченных и непросроченных товаров. 2. Суммарный объем в валюте товаров без категории. 3. Самый дорогой товар в каждой категории. 4. Количество наименований товаров без скидки.
15	В файле содержатся данные о счетах клиентов банка, которые отражают доход (plus) и расход (minus) по счету, а также сумму налога, которую следует оплатить (tax). 1. Суммарный расход по счетам клиентов, у которых не указан e-mail. 2. Клиент с самым большим балансом с учетом уплаты налогов. 3. Средний доход по счетам с положительным балансом. 4. Суммарная сумма налогов у клиентов с отрицательным балансом.
16	1. Сумма фонда з/п в каждой из 3-х категорий сотрудников; 2. Количество сотрудников с дипломом и без; 3. Количество сотрудников с з/п менее $50000,00; 4. Максимальную зарплату среди сотрудников без диплома и минимальную у сотрудников с дипломом.

17	1. Средняя цена в каждой из 3-х категорий; 2. Товар с максимальной ценой; 3. Суммарный объем в валюте непросроченных продуктов; 4. Количество просроченных продуктов.
18	В файле содержатся сведения о товарах на складе (ID, наименование, поставщик, цена, количество, категория, скидка в %). 1. Определите среднюю цену с учетом скидки по каждому из 4-х поставщиков. 2. Найдите самый дорогой и самый дешевый товары. 3. По каждой из 4-х категорий найдите товар с минимальным остатком. 4. Количество товаров без указания поставщика.
19	Дан список сотрудников. 1. Сколько сотрудников не имеют логина или пароля; 2. Средняя з/п сотрудников с семьей и без; 3. Сколько сотрудников имеют машину, но не имеют семьи; 4. Минимальная и максимальная з/п в каждом из 4-х отделов.
20	1. Суммарный объем в валюте для каждого из 4-х поставщиков; 2. Количество наименований товаров дешевле $15,00; 3. Суммарный объем просрочки в валюте у каждого поставщика; 4. Количество товаров с ценой более $50,00.
21	В файле содержатся данные о счетах клиентов банка, которые отражают приход и расход по счету, а также сумму налога, которую следует оплатить. 1. Вычислите средний баланс по счетам с положительным балансом. 2. Количество счетов с отрицательным балансом. 3. Сколько клиентов имеют баланс, недостаточный для уплаты налога? 4. Найдите клиента с максимальным балансом счета.
22	1. Максимальную з/п в каждой из 4-х компаний; 2. Количество сотрудников в каждой из 4-х компаний; 3. Количество сотрудников в каждой языковой группе; 4. Количество сотрудников, получающих зарплату выше средней по всей выборке.
23	В файле содержатся сведения о производстве товаров работниками (ID, сотрудник, категория товара, з/п рабочего, количество произведенных товаров, цена за единицу товара). 1. Определите сколько рабочих получают меньше, чем вырабатывают продукции. 2. Количество единиц произведенной продукции без категории. 3. Суммарный объем в валюте произведенной продукции по каждой из 4-х категорий. 4. Самый неэффективный сотрудник: с минимальной разницей между произведенным продуктом и зарплатой.

24	В файле содержатся данные о товарах (ID, наименование, категория (4 шт.), количество, цена, признак просрочки, признак наличия скидки). 1. Максимальная цена товара для просроченных и непросроченных товаров. 2. Количество товаров без категории. 3. Самый дорогой товар с учетом скидки в каждой категории. 4. Объем товаров в валюте с учетом скидки.
25	В файле содержатся сведения о сотрудниках компании, их зарплатах и сумме налога с зарплаты. 1. Максимальная и минимальная з/п после уплаты налога. Вывести соответствующих сотрудников. 2. Средняя з/п сотрудников без указанного навыка (skill). 3. Количество сотрудников с e-mail. 4. Объем фонда заработной платы.
26	В файле содержатся сведения о товарах на складе (ID, наименование, поставщик, цена, количество, категория, скидка в %). 1. Определите количество наименований товаров в каждой из 4-х категорий. 2. Вычислите суммарный объем (в валюте) товаров по каждому из 4-х поставщиков, о также по товарам, у которых не указан поставщик. 3. Найдите самый дорогой товар с учетом скидки. 4. Найдите товар с минимальным количеством.
27	В файле содержатся сведения о производстве товаров работниками (ID, сотрудник, категория товара, з/п рабочего, количество произведенных товаров, цена за единицу товара). 1. Определите сколько рабочих получают меньше, чем вырабатывают продукции. 2. Количество единиц произведенной продукции без категории. 3. Суммарный объем в валюте произведенной продукции по каждой из 4-х категорий. 4. Количество сотрудников, получающих более 50% от суммы производимого продукта.
28	1. Максимальную з/п среди англоговорящих сотрудников; 2. Средняя з/п в каждой из 4-х компаний; 3. Количество сотрудников в каждой языковой группе; 4. Сотрудник с максимальной з/п.
29	В файле содержатся данные о счетах клиентов банка, которые отражают приход (plus) и расход (minus) по счету, а также сумму налога, которую следует оплатить (tax). 1. Максимальный расход по всем клиентам. 2. Количество счетов с положительным балансом. 3. Суммарный положительный и отрицательный балансы. 4. Средний баланс для клиентов с отрицательным балансом.

30	В файле содержатся данные о товарах (ID, наименование, категория (4 шт.), количество, цена, признак просрочки, признак наличия скидки). 1. Максимальная цена товара для просроченных и непросроченных товаров. 2. Суммарный объем в валюте товаров без категории с учетом скидки. 3. Самый дорогой товар с учетом скидки в каждой категории. 4. Количество наименований товаров без скидки.
31	В файле содержатся сведения о товарах на складе (ID, наименование, поставщик, цена, количество, категория, скидка в %). 1. Определите количество наименований товаров в каждой из 4-х категорий. 2. Вычислите суммарный объем (в валюте) по товарам, у которых не указан поставщик. 3. Найдите самый дорогой товар в каждой категории. 4. Найдите товар с максимальной скидкой.
32	1. Средняя цена для каждого из 4-х поставщиков; 2. Суммарный объем товаров от каждого поставщика; 3. Количество просрочки от каждого поставщика; 4. Количество товаров с ценой более $50,00.
33	В файле содержатся данные о счетах клиентов банка, которые отражают приход (plus) и расход (minus) по счету, а также сумму налога, которую следует оплатить (tax). 1. Максимальный доход по всем клиентам. 2. Количество счетов с отрицательным балансом. 3. Суммарный положительный и отрицательный балансы. 4. Средний баланс для клиентов с отрицательным балансом.
34	Дан список целей, которые описываются параметрами. 1. Определите максимальное значение Param1 по каждой из категорий; 2. Определите сумму параметров Param3 для каждой категории4 3. Сколько целей снабжены оружием, а сколько – нет; 4. Найти объект с максимальным значением Param2.
35	В файле содержатся сведения о товарах на складе (ID, наименование, поставщик, цена, количество, категория, скидка в %). 1. Определите среднюю цену товаров со скидкой более 25%. 2. Вычислите максимальную цену среди товаров, для которых не указа поставщик. Выведите этот товар. 3. Для каждого из 4-х поставщиков определите товар с максимальной ценой. 4. Найдите товар с максимальной ценой с учетом скидки и без учета скидки.

36	1. Средняя цена билета с отправлением из Ставрополя; 2. Количество билетов для каждого их четырех пунктов отправления; 3. Суммарный объем проданных билетов, оформленных на женщин и мужчин; 4. Максимальная и минимальная цена билета в Москву.
37	В файле содержатся данные о товарах (ID, наименование, категория (4 шт.), количество, цена, признак просрочки, признак наличия скидки). 1. Минимальная цена товара для просроченных и непросроченных товаров. 2. Суммарный объем в валюте товаров без категории с учетом скидки. 3. Самый дорогой товар с учетом скидки в каждой категории. 4. Средняя цена товаров в каждой категории.
38	В файле содержатся данные о счетах клиентов банка, которые отражают доход (plus) и расход (minus) по счету, а также сумму налога, которую следует оплатить (tax). 1. Количество клиентов с отрицательным балансом. 2. Клиент с самым большим балансом с учетом уплаты налогов. 3. Средний доход по счетам с отрицательным балансом. 4. Суммарная сумма налогов.
39	Даны сведения о товарах на складах компании (ID, наименование товара, категория товара, количество товара, цена за единицу, склад размещения). 1. Объем товаров в валюте по каждому из складов, а также для товаров без склада. 2. Максимальную цену по каждой категории. 3. Средняя цена товаров, для которых не указана категория. 4. Самый дешевый товар.
40	Дан список доменов. Требуется определить: 1. Суммарная стоимость доменов для которых не указан domain или user; 2. максимальная цена домена по каждой из 4-х категорий; 3. Средняя цена домена среди продающихся и не продающихся; 4. Количество доменов без доменного имени.

3. Содержание отчета и его форма

Отчет по практическому занятию необходимо реализовать в формате Python Notebook. Отчет по практическому занятию должен содержать:

1. Фамилию, Имя и Отчество разработчика, название группы, номер и наименование лабораторной работы.

2. Выполненное индивидуальное задание.

2. Ответы на контрольные вопросы (если требуется, то исходный код к вопросам).

После завершения работы необходимо экспортировать Python Notebook в формат pdf и прикрепить отчет в СЭО.

4. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Напишите код для печати суммы элементов списка **p**:

```
p = [10, 100, 34, 56, 45, 45]
```

2. Напишите код для печати количества элементов списка **p**:

```
p = [10, 100, 34, 56, 45, 45]
```

1. Напишите код для печати суммы элементов кратных 7:

```
p = [10, 100, 35, 34, 56, 45, 45, 21]
```

4. Напишите код для поиска ключей словаря **d**, для которых значения превышают 45:

```
p = {4:10, 12:100, 35:3, 34:56, 56:11, 45:'abc'}
```

5. Напишите код, который выводит список, составленный из квадратов чисел от 1 до 100. Сам список также необходимо сгенерировать.

6. Опишите структуру файла csv. Поясните последовательность действий при чтении файла csv.

ЛАБОРАТОРНАЯ РАБОТА 5. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

1. Цель и содержание

Цель лабораторной работы: научиться конструировать пользовательские типы данных и решать с их использованием практические задачи.

Задачи лабораторной работы:

– научиться объявлять тип данных класс;

– научиться работать с объектами классов;

– научиться объявлять методы и поля класса.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Задание:

Разработать класс для представления объекта «Прямоугольный параллелепипед». Реализуйте все необходимые поля данных (закрытые) и методы позволяющие:

– создавать объекты-параллелепипеды;

– вычислять объем прямоугольного параллелепипеда;

– вычислять площадь поверхности прямоугольного параллелепипеда;

– выводить полную информацию об объекте в выходной файл.

Решение.

Решение данной задачи состоит из следующих этапов: 1) проектирование класса; 2) объявление класса (кодирование); 3) демонстрация использования класса (создание объекта и вызов методов).

Проектирование класса можно производить с использованием любого средства проектирования (UML-редактор, онлайн средства построения диаграмм

(https://www.draw.io), обычный графический редактор). На рисунке 5.1 показана диаграмма классов для спроектированного класса Par.

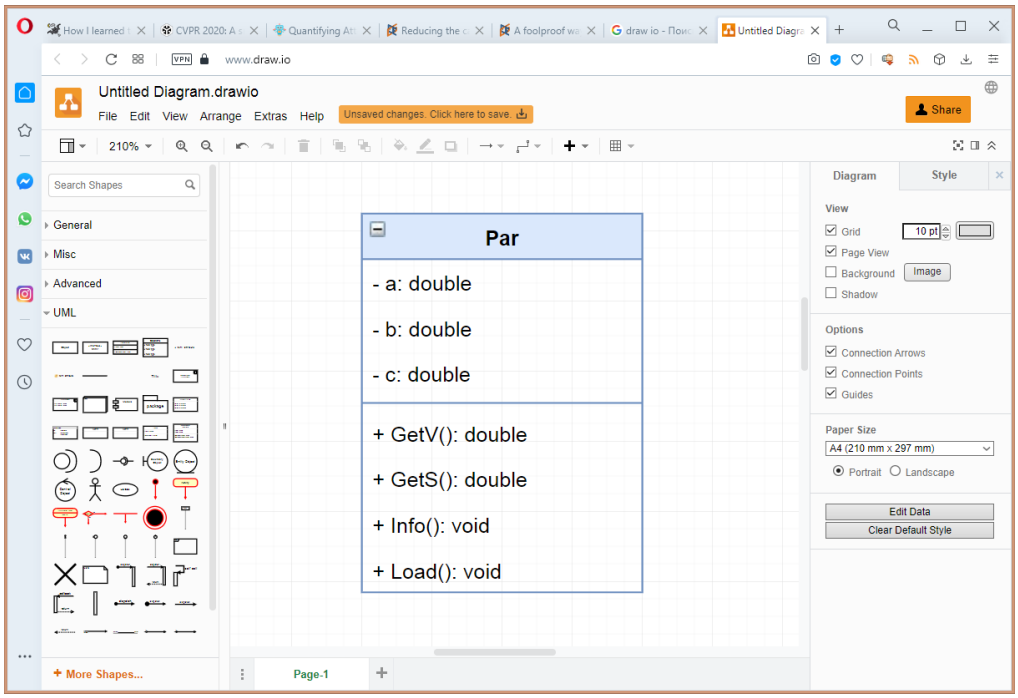


Рисунок 5.1 – Класс Par в редакторе диаграмм классов

Проект класса учитывает все данные и методы, которые можно выявить, проанализировав задание:

1) в классе определены три закрытые переменные a, b, c, описывающие стороны прямоугольного параллелепипед;

2) в классе должны быть следующие открытые методы:

GetV() – возвращает объем прямоугольного параллелепипеда;

GetS() – возвращает площадь поверхности прямоугольного параллелепипеда;

Info() – выводит в стандартный поток информацию о параллелепипеде;

Load() – производит загрузку данных из файла и инициализирует поля объекта данными из файла.

Решение

https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_5.ipynb

2.2. Индивидуальное задание

Напишите скрипт для решения задачи в блокноте Python Notebook или с использованием локального интерпретатора в соответствии с индивидуальным вариантом. Спроектируйте класс, наполните его требуемой функциональностью, продемонстрируйте работоспособность класса.

Вариант	Выражение для вычисления
1	Класс «Шар». Реализовать ввод и вывод полей данных, вычисление объема, диаметра и площади поверхности, а также вывод информации об объекте.
2	Класс «Куб». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, длины диагонали, а также вывод информации об объекте.
3	Класс «Сфера». Реализовать ввод и вывод полей данных, вычисление объема, диаметра и площади поверхности, а также вывод информации об объекте.
4	Класс «Точка в пространстве». Реализовать ввод и вывод полей данных, вычисление расстояния до введенной пользователем точки, расстояния от начала координат, а также вывод информации об объекте.
5	Класс «График y=x». Реализовать ввод и вывод полей данных, вычисление интеграла функции от a до b (вводятся пользователем), длины отрезка функции от (a, y(a)) до (b, y(b)), а также вывод информации об объекте.
6	Класс «Шар». Реализовать ввод и вывод полей данных, вычисление объема, диаметра и площади поверхности, а также вывод информации об объекте.
7	Класс «Матрица MxN». Реализовать инициализацию элементов матрицы случайными числами, вывод матрицы, нахождение максимального и минимального элементов, а также вывод информации об объекте.
8	Класс «Прямоугольный треугольник». Реализовать ввод и вывод полей данных, вычисление гипотенузы, площади и периметра, а также вывод информации об объекте.
9	Класс «Отрезок». Реализовать ввод и вывод полей данных (координаты начала и координаты конца отрезка), вычисление длины, расстояний начала и конца отрезка от начала координат, а также вывод информации об объекте.
10.	Класс «Цилиндр». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.

11	Класс «Ромб». Реализовать ввод и вывод полей данных (диагонали ромба), вычисление площади, периметра, а также вывод информации об объекте.
12	Класс «Точка в пространстве». Реализовать ввод и вывод полей данных, вычисление расстояния до введенной пользователем точки, расстояния от начала координат, а также вывод информации об объекте.
13	Класс «График y=3x+5». Реализовать ввод и вывод полей данных, вычисление интеграла функции от a до b (вводятся пользователем), длины отрезка функции от (a, y(a)) до (b, y(b)), а также вывод информации об объекте.
14	Класс «Матрица MxN». Реализовать инициализацию элементов матрицы случайными числами, вывод транспонированной матрицы, нахождение среднего арифметического всех элементов, а также вывод информации об объекте.
15	Класс «Отрезок в пространстве». Реализовать ввод и вывод полей данных (координаты начала и координаты конца отрезка), вычисление длины, расстояний начала и конца отрезка от начала координат, а также вывод информации об объекте.
16	Класс «График y=x-10». Реализовать ввод и вывод полей данных, вычисление интеграла функции от a до b (вводятся пользователем), длины отрезка функции от (a, y(a)) до (b, y(b)), а также вывод информации об объекте.
17	Класс «Матрица MxN». Реализовать инициализацию элементов матрицы случайными числами, вывод транспонированной матрицы, нахождение и вывод среднего арифметического элементов в каждом столбце.
18	Класс «Куб». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, длины диагонали, а также вывод информации об объекте.
19	Класс «Сфера». Реализовать ввод и вывод полей данных, вычисление объема, диаметра и площади поверхности, а также вывод информации об объекте.
20	Класс «Точка в пространстве». Реализовать ввод и вывод полей данных, вычисление расстояния до введенной пользователем точки, расстояния от начала координат, а также вывод информации об объекте.
21	Класс «График y=x». Реализовать ввод и вывод полей данных, вычисление интеграла функции от a до b (вводятся пользователем), длины отрезка функции от (a, y(a)) до (b, y(b)), а также вывод информации об объекте.
22	Класс «Точка в пространстве». Реализовать ввод и вывод полей данных, вычисление расстояния до введенной пользователем точки,

	расстояния от начала координат, а также вывод информации об объекте.
23	Класс «График y=-x». Реализовать ввод и вывод полей данных, вычисление интеграла функции от a до b (вводятся пользователем), длины отрезка функции от (a, y(a)) до (b, y(b)), а также вывод информации об объекте.
24	Класс «Цилиндр». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
25	Класс «Отрезок в пространстве». Реализовать ввод и вывод полей данных (координаты начала и координаты конца отрезка), вычисление длины, расстояний начала и конца отрезка от начала координат, а также вывод информации об объекте.
26	Класс «Правильная треугольная пирамида». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
27	Класс «Куб». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
28	Класс «Конус». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
29	Класс «Усеченный конус». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
30	Класс «Четырехугольная пирамида». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
31	Класс «Шар». Реализовать ввод и вывод полей данных, вычисление объема, площади поверхности, а также вывод информации об объекте.
32	Класс «Круг». Реализовать ввод и вывод полей данных, вычисление длины окружности, площади, а также вывод информации об объекте.
33	Класс «Параллелограмм». Реализовать ввод и вывод полей данных, вычисление периметра, площади, а также вывод информации об объекте.
34	Класс «Прямоугольный треугольник». Реализовать ввод и вывод полей данных, вычисление площади, гипотенузы, а также вывод информации об объекте.
35	Класс «Товар» с полями цена, количество, скидка. Реализовать ввод и вывод полей данных, вычисление суммарной стоимости со скидкой и без, а также вывод информации об объекте.

36	Класс «Точка на плоскости». Реализовать ввод и вывод полей данных, вычисление расстояния до начала координат, расстояния до заданной пользователем точки, а также вывод информации об объекте.
37	Класс «Книга» с полями: название, автор, количество страниц, количество строк на странице, количество символов в строке. Реализовать ввод и вывод полей данных, вычисление количества строк во всей книге, количества символов во всей книге, а также вывод информации об объекте.
38	Класс «Транспорт» с полями: скорость, объем бака, расход топлива на 1 километр. Реализовать ввод и вывод полей данных, вычисление времени в пути по переданному расстоянию, количества затрачиваемого топлива по переданному расстоянию, хватит или нет топлива на заданное расстояние, а также вывод информации об объекте.
39	Класс «Тест», содержащий поля: тема, вопрос, ответ. Реализовать ввод и вывод полей данных, функционал вывода вопроса, получения ответа, вывода информации верен ответ или нет, а также вывод информации об объекте.
40	Класс «Равноускоренное движение», содержащий поля начальная скорость, ускорение. Реализовать ввод и вывод полей данных, вычисление пройденного расстояния за заданное время, скорость в заданный момент времени после начала движения, а также вывод информации об объекте.

3. Содержание отчета и его форма

Отчет по практическому занятию необходимо реализовать в формате Python Notebook. Отчет по практическому занятию должен содержать:

1. Фамилию, Имя и Отчество разработчика, название группы, номер и наименование лабораторной работы.

2. Выполненное индивидуальное задание.

2. Ответы на контрольные вопросы (если требуется, то исходный код к вопросам).

После завершения работы необходимо экспортировать Python Notebook в формат pdf и прикрепить отчет в СЭО.

4. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Дайте определение понятию класс.

2. Продемонстрируйте использование метода __init__. Поясните назначение данного метода.

3. Поясните назначение статических методов и их отличия от методов класса.

4. Для чего используется параметр self в каждом методе класса?

5. Сколько методов __init__ может быть объявлено внутри класса?

ЛАБОРАТОРНАЯ РАБОТА 6. РАБОТА С ИЗОБРАЖЕНИЯМИ. БИБЛИОТЕКА PILLOW

1. Цель и содержание

Цель лабораторной работы: научиться обрабатывать изображения с использованием библиотеки Pillow.

Задачи лабораторной работы:

– научиться открывать, сохранять и конвертировать изображения;

– научиться работать каналами изображений;

– научиться фильтровать, трансформировать изображения.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Изучите код практической работы:
https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_6.ipynb

2.2. Индивидуальное задание

Изучите инструментарий библиотеки Pillow. Спроектируйте свой собственный фильтр. Продемонстрируйте его работу и поясните назначение предложенного фильтра.

3. Содержание отчета и его форма

Отчет по практическому занятию необходимо реализовать в формате Python Notebook. Отчет по практическому занятию должен содержать:

1. Фамилию, Имя и Отчество разработчика, название группы, номер и наименование лабораторной работы.

2. Выполненное индивидуальное задание.

2. Ответы на контрольные вопросы (если требуется, то исходный код к вопросам).

После завершения работы необходимо экспортировать Python Notebook в формат pdf и прикрепить отчет в СЭО.

4. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Перечислите функции для трансформации изображения: вращение, обрезка и т.д.

2. Поясните назначение метода **`convert()`**. Поясните возможные значения параметра данного метода.

3. Что такое фильтрация изображений? Приведите примеры практического применения фильтров.

4. Поясните формат изображения RGB.

5. Поясните значение терминов: растр, формат изображения, палитра, глубина, разрешение изображения.

ЛАБОРАТОРНАЯ РАБОТА 7. ОСНОВЫ NUMPY. ПОСТРОЕНИЕ ГРАФИКОВ

1. Цель и содержание

Цель лабораторной работы: научиться манипулировать векторно-матричными данными с использованием библиотеки NumPy.

Задачи лабораторной работы:

– научиться создавать массивы numpy;

– научиться визуализировать данные с использование matplotlib.

2. Методика и порядок выполнения работы

2.1. Пример учебного практического задания

Задание:

Необходимо построить график функции, заданный выражениями:

Решение

https://github.com/enikolaev/IT-and-Programming/blob/main/Практическая_работа_7.ipynb

2.2. Индивидуальное задание

Напишите скрипт для решения задачи в блокноте Python Notebook или с использованием локального интерпретатора в соответствии с индивидуальным вариантом.

Файлы для индивидуального задания:

https://github.com/enikolaev/IT-and-Programming/tree/main/data/pract4

В каждом варианте необходимо выполнить визуализацию данных, хранящихся в файле с использованием библиотеки matplotlib. Необходимо реализовать не менее 2 типов графиков. Учащийся самостоятельно выбирает

отображаемые данные: зарплату, численность сотрудников по отделам, гистограмма заработной платы и т.д.

3. Содержание отчета и его форма

Отчет по практическому занятию необходимо реализовать в формате Python Notebook. Отчет по практическому занятию должен содержать:

1. Фамилию, Имя и Отчество разработчика, название группы, номер и наименование лабораторной работы.

2. Выполненное индивидуальное задание.

2. Ответы на контрольные вопросы (если требуется, то исходный код к вопросам).

После завершения работы необходимо экспортировать Python Notebook в формат pdf и прикрепить отчет в СЭО.

4. Контрольные вопросы

Ответы на контрольные вопросы также необходимо предоставить в блокноте Python Notebook:

1. Чем список python отличается от массива numpy?

2. Продемонстрируйте варианты создания массива numpy: из списка, как диапазон с заданным шагом, массив нуле, единиц.

3. Продемонстрируйте выполнение векторных операций с массивами numpy: умножение вектора на число, сложение векторов.

4. Реализуйте построение векторов numpy с использованием генераторов случайных чисел.

5. Перечислите основные типы диаграмм matplotlib и функции, которые реализуют построение графиков данных типов.

ЛАБОРАТОРНАЯ РАБОТА 8. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА C#

1. Цель и содержание

Цель лабораторной работы: изучить операторы, позволяющие организовывать непоследовательное выполнение программного кода.

Задачи лабораторной работы:

– научиться объявлять переменные простых типов в языке C#;

– научиться объявлять константы простых типов в языке C#;

– научиться выполнять простейшие действия с переменными и константами;

– научиться применять условныq оператор if .. else;

– научиться применять операторs цикла for, while, do … while;

– научиться применять операторы выбора switch.

2. Теоретическая часть

2.1. Структура консольного приложения

Рассмотрим структуру консольного приложения на языке C#, созданного с использованием средств MS Visual Studio (рис. 8.1).

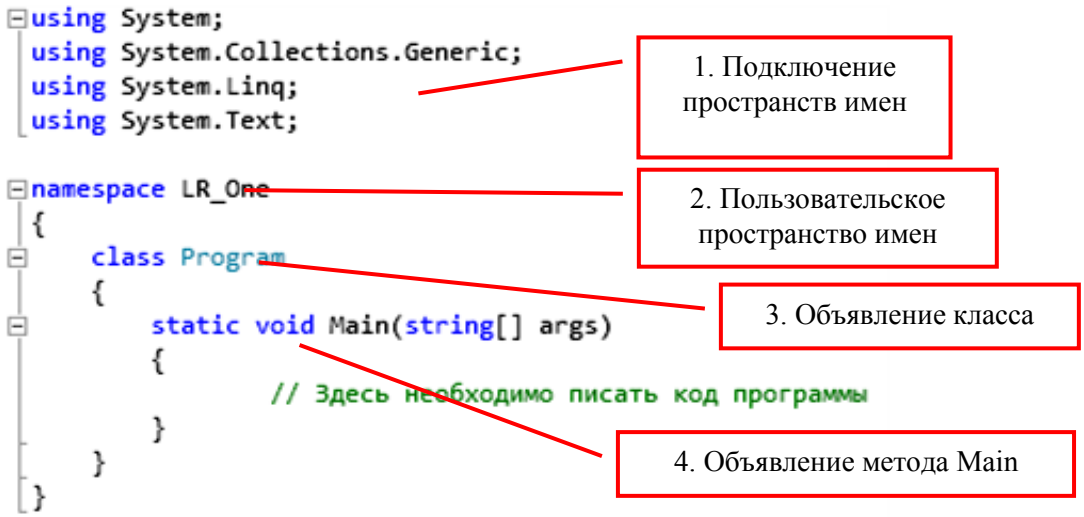


Рисунок 8.1 – Исходный код консольного приложения

В интерактивном редакторе кода среды разработки код может быть свернут/развернут с использованием кнопок +/-, внедренных в код. На рис. 8.1 в области 1 показано подключение пространств имен (библиотек, содержащих стандартные инструменты) с использованием зарезервированного слова using.

Программист сам создает свое пространство имен с именем LR_One (2), в котором объявляется класс с именем Program.

В классе Program объявлен один метод – функция Main (параметры функции не рассматриваем).

Функция Main имеет особенное значение в программировании на языках C, C++ и C#.

Функцию Main называют «точкой входа», то есть началом выполнения программы. Далее мы рассмотрим приложения, содержащие множество функций. Операционная система знает с какой именно функции начать выполнение программы – с функции Main. Очевидно, что имя этой функции менять нельзя. Это должен быть статический метод класса (или структуры), возвращающий либо значение типа int, либо void. Хотя нередко модификатор public указывается явно, поскольку по определению этот метод должен быть вызван извне программы, на самом деле неважно, какой уровень доступа вы назначите методу точки входа. Он запустится, даже если вы пометите его как private.

Когда компилируется консольное или Windows-приложение C#, по умолчанию компилятор ищет в точности один метод Main () с описанной выше сигнатурой в любом классе и делает его точкой входа программы. Если существует более одного метода Main ( ), компилятор возвращает сообщение об ошибке.

Обратите внимание на вложенность конструкций на рис. 1.1: в пространство имен LR_One вложен класс Program, в класс Program вложена функция Main. В свою очередь, в функции Main содержатся инструкции на языке C#-код, который начнет выполняться при старте программы.

В C#, как и в других C-подобных языках, большинство операторов завершаются точкой с запятой (;) и могут продолжаться в нескольких строках без необходимости указания знака переноса. Операторы могут быть объединены в

блоки с помощью фигурных скобок ({ }). Однострочные комментарии начинаются с двойного слеша (//), а многострочные – сослеша со звездочкой (/*) и заканчиваются противоположной комбинацией (*/). В этих аспектах язык C# идентичен C++ и Java.

Следует также помнить о том, что язык C# чувствителен к регистру символов. Это значит, что переменные с именами myVar и MyVar являются разными.

Причина присутствия оператора using в файле Program.cs связана с использованием библиотечных классов.

Весь код C# должен содержаться внутри класса. Объявление класса состоит из ключевого слова class, за которым следует имя класса и пара фигурных скобок. Весь код, ассоциированный с этим классом, размещается между этими скобками.

2.2. Типы значений C#

Язык C# поддерживает 8 предопределенных целочисленных типов (таблица 8.1).

Таблица 8.1 – Целочисленные типы C#

Имя типа	Тип CTS	Описание	Диапазон (минимум : максимум)
sbyte	System.SByte	8-битное целое со знаком	-128 : 127
short	System.Int16	16-битное целое со знаком	-32 768 : 32 767
int	System.Int32	32-битное целое со знаком	-2 147 483 648 : 2 147 483 647
long	System.Int64	64-битное целое со знаком	$-2^{63} : 2^{63}-1$
byte	System.Byte	8-битное целое без знака	0 : 255
ushort	System.UInt16	16-битное целое без знака	0 : 65 535
uint	System.UInt32	32-битное целое без знака	$0 : 2^{32}-1$
ulong	System.UInt64	64-битное целое без знака	$0 : 2^{64}-1$

Язык C# также поддерживает и типы с плавающей точкой (таблица 8.2).

Таблица 8.2 – Типы с плавающей точкой C#

Имя типа	Тип CTS	Описание	Кол-во знаков	Диапазон (минимум : максимум)
float	System.Single	32-битное с плавающей точкой одинарной точности	7	от $\pm 1.5\times10^{-45}$ до $\pm 3.4\times10^{38}$
double	System.Double	64-битное с плавающей точкой двойной точности	15/16	от $\pm 5.0\times10^{-324}$ до $\pm 1.7\times10^{308}$

В таблице 8.3 представлен десятичный тип C#. Данный тип реализован для финансовых операций.

Таблица 8.3 – Десятичный тип C#

Имя типа	Тип CTS	Описание	Кол-во знаков	Диапазон (минимум : максимум)
decimal	System.Decimal	128-битное с плавающей точкой в десятичной нотации с высокой точностью	28	от $\pm 1.0\times10^{-28}$ до $\pm 7.9\times10^{28}$

Как и во многих языках программирования существует булевский тип (таблица 8.4).

Таблица 8.4 – Булевский тип.

Имя типа	Тип CTS	Значения
bool	System.Boolean	true или false

Для хранения одиночных символов в языке C# используется тип char (таблица 8.5)

Таблица 8.5 – Булевский тип.

Имя типа	Тип CTS	Значения
char	System.Char	Представляет отдельный 16-битный (Unicode) символ

Литералы типа char записываются как одиночные, заключенные в одинарные кавычки символы: ‘F’, ‘w’, ‘ц’, ‘Я’ и т.д.

В переменных типа char можно хранить и специальные символы в виде управляющих последовательностей (таблица 8.6).

Таблица 8.6 – Представление символов в виде управляющих последовательностей.

Управляющая последовательность	Символ
\’	Одиночная кавычка
\”	Двойная кавычка
\\	Обратный слэш
\0	Пусто
\a	Предупреждение (звуковой сигнал)
\b	Забой
\f	Подача формы
\n	Новая строка
\r	Возврат каретки
\t	Символ табуляции
\v	Вертикальная табуляция

Если отдельные символы объединены в строку, то необходимо использовать тип string, который отображается на тип CTS – System.String.

2.3. Объявление и инициализация переменных в C#

Синтаксис объявления переменных в C# выглядит следующим образом:

ТипДанных ИдентификаторПеременной;

Например:

```
int a;
```

Этот код объявляет переменную типа int с именем a. Компилятор не позволит использовать эту переменную до тех пор, пока она не будет инициализирована (т.е. пока ей не будет присвоено значение).

Для инициализации переменной a необходимо написать следующий код:

```
a = 123;
```

Переменную можно инициализировать во время объявления:

```
int b = 7;
```

или

```
string str = "Hello, World!!!";
```

Синтаксис C# позволяет объявить несколько переменных (и инициализировать их) одного типа в одной синтаксической конструкции.

Например:

```
float b, i, myPerem, U_t = 0.3F, z_11 = 23.56F;
```

В данном примере объявляется 5 переменных типа float, некоторые из них инициализируются в процессе объявления.

2.4. Объявление и инициализация констант в C#

Константа – это переменная, значение которой не меняется за время выполнения программы. Для объявления константы необходимо воспользоваться ключевым словом const. Пример:

```
const char simv = 'A';
const double pi = 3.14;
```

Очевидно, что при таком объявлении, поменять значения simv и pi в дальнейшем будет нельзя.

2.5. Условный оператор

В предыдущих лабораторных работах были рассмотрены вопросы программирования на языке C# с использованием только последовательного выполнения операторов программы.

В языке C# (как и в большинстве языков) можно вводить условия, циклы, ветвления.

Условный оператор в C# позволяет организовать условие типа «если …, то …, иначе …».

Синтаксис условного оператора:

if (*условие*)
    оператор
[else if (*условие*)
    оператор
...
else
    оператор]

В данном синтаксисе *оператор* может быть составным оператором (группа операторов языка, заключенные в фигурные скобки). Условный оператор содержит только одно обязательное зарезервированное слово – if. Все остальные конструкции не являются обязательными (в синтаксическом описании на это указывают скобки [ ]).

Пример использования условного оператора:

```
int a = 100, b = 13, c = 23, d = 0, e = 0;


if (a == 100) // Если a равно 100
{
    // Увеличиваем b на 1000
    b += 1000;
}
else if (a < 100) // иначе если a меньше 100
{
    // изменяем значение d
    d = b + c;
    // и изменяем значение e
    e = a + b;
}
else // иначе (остальные варианты не предусмотренные первыми двумя условиями)
    // edtkbxbdftv a на 1
    a++;

// продолжается последовательное выполнение программы
Console.WriteLine("{0}", a);
```

В представленном примере сначала проверяется является ли значение переменной a равным 100.

1. Если «Да», то выполняется единственный оператор (увеличение b на 1000). На этом выполнение всего условного оператора завершено и программа переходит к выполнению операторов, следующих за ним, то есть к выводу значения a.

2. Если первое условие не выполнилось, проверяется второе – является ли a меньше 100 (конструкция else if). Если это утверждение истинно, то выполняется составной оператор, состоящий из двух – изменение значений переменных d и e. Затем осуществляется переход к выводу переменной a.

3. Если не выполнилось ни одно условие с оператором if, то выполняется оператор, указанный после зарезервированного слова else. В данном случае это несоставной оператор (a увеличивается на 1), поэтому фигурные скобки можно не писать.

Во всей этой конструкции только оператор if является обязательным. Конструкций else if может быть любое количество, а if и else – только по одному.

Еще один пример условного оператора:

```
int a = 100;

if (a >= 100)
{
    Console.WriteLine("Значение переменной a больше или равно 100");
}
```

В данном случае вывод в консоль выполнится только если a больше либо равно 100.Иначе условный оператор ничего не выведет. Фигурные скобки в данном примере можно опустить.

Обратите внимание, что для проверки на равенство используется оператор ==, а не =. Знак «равно» используется в C# для присваивания значений. Следует понимать, что условное выражение, стоящее в конструкции if должно возвращать булево значение.

Например, следующий условный оператор всегда будет выполняться:

```
if (true)
    Console.WriteLine("Всегда выводится");
else
    Console.WriteLine("Никогда не выводится");
```

2.6. Оператор выбора

Синтаксис оператора:

switch (*перем*)
{
    case *константа*1 :
        *оператор _*1;
        break;
    case *константа*2 :
        оператор _ 2;
        break;
...
    default :
        оператор _ n;
        break;
}

В данном операторе осуществляется выбор оператора (который может быть составным) в зависимости от значения переменной *перем*. Если *перем* равна значению *константа*1, то выполнится *оператор _*1, если *константа*2 – *оператор _* 2, и т.д. Если ни одно значение, указанное в каком-либо операторе case, не совпало со значением *перем*, то выполнится оператор, указанный в секции default. Внутри каждой секции case следует указать оператор break, который предотвращает проверку других условий после выполнения данного оператора. Следует обратить внимание, что в секциях case следует использовать только константы (переменные не допускаются).

Пример использование оператора switch … case:

```
// Организация ответа на действие пользователя:
string text = "1 - Вывод группы \n" +
    "2 - Вывод фамилии преподавателя \n" +
    "3 - Вывод названия предмета \n" +
    "4 - Вывод приветствия \n";

Console.Write(text + "Введите команду > ");
int result = Convert.ToInt32(Console.ReadLine());

switch (result)
{
    case 1: { Console.WriteLine("ИСТБ - 101"); break; }
    case 2: Console.WriteLine("Николаев Евгений Иванович"); break;
    case 3: { Console.WriteLine("Технологии программирования"); break; }
    case 4: Console.WriteLine("Привет !"); break;
    default: Console.WriteLine("Недопустимый вариант !!!"); break;
}
```

Изучите представленный пример самостоятельно.

2.7. Цикл for

Циклы позволяют выполнять определенную последовательность операторов до тех пор, пока выполняется некоторое условие. Каждый «круг» выполнения этого блока называется итерацией.

Синтаксис оператора:

for (*инициализатор*; *условие*; *итератор*)
    оператор

инициализатор – выражение, выполняемое перед первой итерацией цикла.

условие – выражение булевого типа, которое проверяется перед каждой итерацией. Если оно истинно то итерация выполняется, иначе – цикл завершается.

итератор – выражение, вычисляемое после каждой итерации.

Пример применения оператора цикла for (пример 1.1):

```
/*
 * Вычислить сумму чисел от 1 до 1000
 */

// Объявляем переменную, которая хранит значение суммы чисел
System.Int32 Sum = 0;
// Объявляем счетчик цикла
int i;

for (i = 1; i <= 1000; i++)
    Sum += i;

Console.WriteLine("Итог: {0}", Sum);
```

Изучите код самостоятельно (создайте такую программу в VS). Еще один пример для самостоятельного изучения (пример 1.2):

```
/*
* вычислить сумму всех чисел, делящихся на 3 без остатка
* и находящихся в диапазоне от 1 до 1000000
*/

long Sum = 0;
int i;

for (i = 1; i <= 1000000; i++)
    if (i % 3 == 0)
        Sum += i;
```

Пример, отображающий возможность применения итератора (пример 1.3):

```
/*
* вычислить сумму всех чисел, делящихся на 3 без остатка
* и находящихся в диапазоне от 1 до 1000000
*/

long Sum = 0;
int i;

for (i = 0; i <= 1000000; i += 3)
        Sum += i;
```

Обратите внимание, что примеры 2 и 3 решают одну и ту же задачу.

2.8. Цикл while

Цикл while похож на for тем, что является конструкцией с предварительной проверкой условия продолжения цикла. Но синтаксис цикла while более лаконичен:

while (*условие*)
    оператор

В данном синтаксисе *оператор* может быть составным оператором (группа операторов языка, заключенные в фигурные скобки).

Следует понимать, что while используется для заранее неизвестного количества повторных выполнений операторов.

Пример выполнения цикла (пример 2.1):

```
/*
 * Цикл будет выполняться пока пользователь не введет 0
 */

bool Stop = false;
int chislo;

while (!Stop)
{
    Console.Write("Введите число > ");
    chislo = Convert.ToInt32(Console.ReadLine());

    if (chislo == 0)
        Stop = true;
}
```

Оператор break уже встречался в конструкции case оператора switch для выхода из case. Но break часто применяется внутри циклов for, while и do … while. Данный оператор прерывает выполнение цикла и передает управление оператору, следующему за циклом.

Пример 2.2, выполняющий то же самое, что и пример 2.1:

```
/*
 * Цикл будет выполняться пока пользователь не введет 0
 */

bool Stop = true;
int chislo;

while(Stop)
{
    Console.Write("Введите число > ");
    chislo = Convert.ToInt32(Console.ReadLine());

    if (chislo == 0)
        break;
}
```

Следует обратить внимание, что цикл является бесконечным (логическая переменная Stop всегда является true), но выполнение цикла все равно прервется, если пользователь введет 0.

Оператор continue также применяется внутри цикла – он останавливает выполнение текущей итерации и немедленно переходит к выполнению следующей итерации.

2.9. Цикл do … while

Цикл do … while полностью повторяет функциональные возможности while, но предполагает проверку условия окончания цикла после выполнения тела цикла. То есть блок операторов тела цикла всегда выполнится хотя бы один раз.

Синтаксис оператора:

do
{
    оператор (*операторы*)
}
while (*условие*);

Пример использования цикла (пример 3.1):

```
/*
 * Цикл будет выполняться пока пользователь не введет 0
 */

bool Stop = false;
int chislo;

do
{
    Console.Write("Введите число > ");
    chislo = Convert.ToInt32(Console.ReadLine());

    if (chislo == 0)
        Stop = true;
} while (!Stop);
```

3. Методика и порядок выполнения работы

1. Создайте консольное приложение.

2. Выполните индивидуальное задание. Пользователю предлагается выбрать метод расчета значения выражения: 1) с помощью цикла for; 2) с помощью цикла while; или 3) с помощью цикла do ... while. После выбора способа расчета, запрашиваются входные данные и производится расчет с использованием данного вида цикла.

3. Во всех заданиях переменные X, Y являются вещественными и вводятся пользователем. Количество слагаемых также вводится пользователем. Программа должна вывести сумму заданного числа членов последовательности.

Индивидуальное задание.

Перед выполнением задания требуется самостоятельно определить закономерность изменения членов последовательности, чтобы применить цикл, условный оператор или, если потребуется, оператор выбора.

Вариант	Выражение для вычисления
1	$j = -\frac{\sin^3(Y)}{1\cdot 3} + \frac{\sin^5(X^2)}{3\cdot 5} - \frac{\sin^7(Y^3)}{5\cdot 7} + \frac{\sin^9(X^4)}{7\cdot 9} - \ldots$
2	$Z = \frac{1}{1\cdot 3\cdot 5} - \frac{X}{2\cdot 4\cdot 6} + \frac{Y^2}{3\cdot 5\cdot 7} - \frac{X^3}{4\cdot 6\cdot 8} + \frac{Y^4}{5\cdot 7\cdot 9} - \ldots$
3	$Z = \frac{X^2}{1\cdot 3} - \frac{Y^4}{3\cdot 5} + \frac{X^6}{5\cdot 7} - \frac{Y^8}{7\cdot 9} + \ldots$
4	$Z = \frac{Y^2\cdot X}{2} - \frac{Y^2\cdot X^4}{4} + \frac{X^4\cdot Y^6}{6} - \frac{X^8\cdot Y^6}{8} + \ldots$
5	$A = -\frac{\sin(X)\cdot\lg(Y)}{1!} + \frac{\ln(X^3)\cdot\cos(Y^2)}{3!} - \frac{\sin(X^5)\cdot\lg(Y^3)}{5!} + \frac{\ln(X^7)\cdot\cos(Y^4)}{7!} - \ldots$
6	$Z = 1 - \frac{X}{2} + \frac{Y^2}{6} - \frac{X^3}{24} + \frac{Y^4}{120} - \ldots$ (в знаменателе факториал)
7	$s = -\frac{Y\cdot X^2}{1\cdot 3} + \frac{X\cdot Y^3}{2\cdot 4} - \frac{Y\cdot X^4}{3\cdot 5} + \frac{X\cdot Y^5}{4\cdot 6} - \ldots$
8	$Z = 1 - \frac{\sin(X^2)}{2} + \frac{\cos(Y^3)}{3} - \frac{\sin(X^4)}{4} + \frac{\cos(Y^5)}{5} - \ldots$
9	$Z = \frac{1}{1\cdot 3} - \frac{X}{2\cdot 4} + \frac{X^2}{3\cdot 5} - \frac{X^3}{4\cdot 6} + \ldots$
10	$Z = \frac{X^2}{1\cdot 3} - \frac{Y^4}{3\cdot 5} + \frac{X^6}{5\cdot 7} - \frac{Y^8}{7\cdot 9} + \ldots$
11	$p = -\frac{X\cdot\sin(Y)}{1\cdot 2} + \frac{X^3\cdot\cos(Y^2)}{3\cdot 4} - \frac{X^5\cdot\sin(Y^3)}{5\cdot 6} + \frac{X^7\cdot\cos(Y^4)}{7\cdot 8} - \ldots$
12	$p = -\frac{X}{1\cdot 2\cdot 3} + \frac{Y^3}{3\cdot 4\cdot 5} - \frac{X^5}{5\cdot 6\cdot 7} + \frac{Y^7}{7\cdot 8\cdot 9} - \ldots$
13	$Z = 1 - \frac{\sin(X^2)+Y}{2} + \frac{\cos(Y^3)+X}{3} - \frac{\sin(X^4)+Y}{4} + \frac{\cos(Y^5)+X}{5} - \ldots$

14	$Z = \frac{Y^2 + X}{1 \cdot 2} - \frac{Y^2 - X^4}{2 \cdot 4} + \frac{X^4 + Y^6}{4 \cdot 6} - \frac{X^8 - Y^6}{6 \cdot 8} + \ldots$
15	$p = 1 - \frac{\cos(X) \cdot \sin^2(Y)}{1 \cdot 2} + \frac{\cos^4(X) \cdot \sin^3(Y)}{3 \cdot 4} - \frac{\cos^5(X) \cdot \sin^6(Y)}{5 \cdot 6} + \ldots$
16	$s = -\frac{\sin(Y) \cdot X^2}{1 \cdot 3} + \frac{X \cdot \cos(Y^3)}{2 \cdot 4} - \frac{\sin(Y) \cdot X^4}{3 \cdot 5} + \frac{X \cdot \cos(Y^5)}{4 \cdot 6} - \ldots$
17	$Z = -\frac{tg(Y) \cdot ctg(X^2)}{1 \cdot 3} + \frac{tg(X) \cdot ctg(Y^3)}{2 \cdot 4} - \frac{tg(Y) \cdot ctg(X^4)}{3 \cdot 5} + \frac{tg(X) \cdot ctg(Y^5)}{4 \cdot 6} - \ldots$
18	$Z = \frac{1}{1!} - \frac{X}{2!} + \frac{Y^2}{3!} - \frac{X^3}{4!} + \frac{Y^4}{5!} - \ldots$
19	$Z = \frac{\cos(X) + \lvert Y \rvert}{1!} - \frac{\cos^2(Y) + \lvert X \rvert}{2!} + \frac{\cos^3(X) + \lvert Y \rvert}{3!} - \frac{\cos^4(X) + \lvert Y \rvert}{4!} + \ldots$
20	$Z = \frac{Y^2 \cdot X}{2 + 1} - \frac{Y^2 \cdot X^4}{4 - 2} + \frac{X^4 \cdot Y^6}{6 + 4} - \frac{X^8 \cdot Y^6}{8 - 6} + \ldots$
21	$Z = 1 + \frac{1 - \cos(Y)}{1 \cdot 3 \cdot 5} - \frac{Y + \cos^2(X)}{2 \cdot 4 \cdot 6} + \frac{X^2 - \cos^3(Y)}{3 \cdot 5 \cdot 7} - \frac{Y^3 + \cos^4(X)}{4 \cdot 6 \cdot 8} + \ldots$
22	$Z = \frac{X - Y}{1 \cdot 3} - \frac{Y^2 - X^2}{2 \cdot 4} + \frac{X^3 - Y^3}{3 \cdot 5} - \frac{Y^4 - X^4}{4 \cdot 6} + \ldots$
23	$Z = 1 + \frac{1 - \cos(Y)}{1 \cdot 3 \cdot 5} - \frac{Y + \cos^2(X)}{2 \cdot 4 \cdot 6} + \frac{X^2 - \cos^3(Y)}{3 \cdot 5 \cdot 7} - \frac{Y^3 + \cos^4(X)}{4 \cdot 6 \cdot 8} + \ldots$
24	$Z = \frac{(X - Y)^2}{1 \cdot 3} - \frac{(Y - X)^4}{3 \cdot 5} + \frac{(X - Y)^6}{5 \cdot 7} - \frac{(Y - X)^8}{7 \cdot 9} + \ldots$
25	$Z = \frac{Y^2 \cdot \sin(X)}{2} - \frac{\cos(Y^2) \cdot X^4}{4} + \frac{\sin(X^4) \cdot Y^6}{6} - \frac{\cos(Y^6) \cdot X^8}{8} + \ldots$

4. Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы.

2. Цели лабораторной работы.

3. Ответы на контрольные вопросы.

4. Экранные формы и листинг программного кода, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

5. Контрольные вопросы

1. Как оформляется комментарий?

2. Опишите синтаксис условного оператора. Какие части данного оператора являются обязательными?

3. Опишите синтаксис оператора выбора. Поясните почему следует использовать оператор break внутри каждого выбора.

4. Можно ли с помощью for реализовать бесконечный цикл? Поясните ответ на примерах.

5. В разделе «Теоретическое обоснование» приведены примеры 1.1, 1.2 и 1.3. Почему в примерах 1.2 и 1.3 для переменной Sum выбран тип long, хотя в примере 1 для Sum выбран тип int? Обоснуйте ответ.

6. Опешите синтаксис оператора цикла for.

7. Опишите синтаксис оператора while.

8. Поясните назначение операторов break и continue.

9. Опешите синтаксис оператора do … while.

ЛАБОРАТОРНАЯ РАБОТА 9. ДВУМЕРНАЯ ГРАФИКА В ОКОННЫХ ПРИЛОЖЕНИЯХ

1. Цель и содержание

Цель лабораторной работы: научиться использовать инструменты визуализации двумерной графики, предоставляемые пространством имен GDI+.

Задачами лабораторной работы являются:

– научится использовать основные типы, предоставляемые библиотекой GDI+;

– научиться использовать различные режимы вывода графики;

– освоить принципы построения графиков, диаграмм;

– научиться создавать интерактивные приложения;

– научиться обрабатывать события в графических приложениях;

– научиться использовать класс Timer;

– научиься реализовывать механизм анимации в приложениях Windows.

2. Теоретическое обоснование

2.1 Основы GDI+

В таблице 9.1 приводится описание базовых пространств имен GDI+.

Таблица 9.1 – Базовые пространства имен GDI+

Пространство имен	Описание
System.Drawing	Базовое пространство имен GDI+, определяющее множество типов для основных операций визуализации (шрифты, перья, кисти и т.д.), а также тип Graphics.
System.Drawing.Drawing2D	Содержит описание типов, используемых для более сложной двумерной/векторной графики (градиентные кисти, стили окончания линий, геометрические трансформации и т.д.)

Пространство имен	Описание
System.Drawing.Imaging	Содержит описание типов, обеспечивающих обработку графических изображений (изменение палитры, извлечение метаданных изображения, работа с метафайлами и т.д.)
System.Drawing.Printing	Содержит описание типов, обеспечивающих отображение графики на печатной странице, непосредственное взаимодействие с принтером.
System.Drawing.Text	Дает возможность управлять коллекциями шрифтов

При создании приложения Windows с использованием мастера, ссылка на System.Drawing.dll устанавливается автоматически.

Обзор пространства имен System.Drawing.

В таблице 9.2 приводится описание основных типов, необходимых при работе с графикой.

Таблица 9.2 – Базовые типы пространства имен System.Drawing

Тип	Описание
Bitmap	Тип, инкапсулирующий данные изображения
Brush, Brushes, SolidBrush, SystemBrushes, TextureBrush	Объекты Brush используются для заполнения внутренних областей графических форм (прямоугольников, эллипсов и т.д.)
BufferedGraphics	Новый тип .NET 2.0, обеспечивающий графический буфер для двойной буферизации, которая используется для уменьшения или полного исключения эффекта мелькания, возникающего при перерисовке
Color SystemColors	Определяют ряд статических свойств, доступных только для чтения и используемых для получения нужного цвета при использовании перьев и кистей
Font, FontFamily	Тип Font инкапсулирует характеристики данного шрифта (название, плотность, размер и т.д.). FontFamily предлагает абстракцию для группы шрифтов, имеющих аналогичный дизайн, но определенные вариации стиля
Graphics	Представляет реальную поверхность нанесения изображения, а также предлагает ряд методов для визуализации текста, изображений и геометрических шаблонов

Pen Pens SystemPens	Pens – это объекты, используемые для построения линий, кривых. Тип Pen определяет ряд статических свойств, возвращающих новый объект Pen заданного цвета.
Point, PointF	Структуры, представляющие точку
Rectangle RectangleF	Структуры, представляющие прямоугольник
Size SizeF	Структуры, представляющие размер (в виде целого или вещественного числа)

Утилитарные типы System.Drawing.

Многие из методов рисования, определенные объектом System.Drawing.Graphics, требуют указать позицию или область, в которой требуется отобразить данный элемент. Другие методы требуют указания размеров, координат, границ геометрического образа.

Тип Point(F) поддерживает ряд полезных членов:

- перегруженные операции: +, -, ==, != ;

- доступ к значениям (x,y): X, Y;

- поле IsEmpty возвращает значение true, если x и y установлены в 0.

Пример:

Типы **Rectangle(F)**, как и Point, необходимы в большинстве GUI-приложений. Одним из наиболее полезных методов типа Rectangle является Contains(), который позволяет выяснить, находится ли данный тип Point или Rectangle в рамках границ некоторого другого объекта.

Пример:

Класс Region представляет внутреннюю часть геометрической фигуры, поэтому конструкторы класса Region требуют, чтобы им на вход был предоставлен некоторый геометрический объект.

Например, имея внутреннюю часть фигуры, вы можете манипулировать ею с использованием различных членов.

Класс Graphics.

Класс System.Windows.Graphics – это интерфейс для функциональных возможностей GDI+. Этот класс не только предоставляет поверхность, на которой

размещаются изображения (поверхность формы, поверхность элемента управления или область в памяти), но определяет также члены, которые позволяют отображать текст, изображения (пиктограммы, точечные рисунки и т.д.) и самые разные геометрические формы.

Класс Graphics не допускает непосредственного создания своего экземпляра с помощью ключевого слова new, поскольку класс не имеет открытых конструкторов.

Для получения доступа к объекту Graphics необходимо создать обработчик события WM_PAINT, для этого для формы необходимо зарегистрировать обработчик события Paint. В качестве параметра метода-обработчика будет передан объект типа PaintEventArgs. В обработчике события Paint можно будет получить объект Graphics следующим образом:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics new_graf = e.Graphics;

    new_graf.DrawRectangle(10, 10, 100, 100);

}
```

Рисунок 9.1 – Получение объекта Graphics через параметр PaintEventArgs обработчика события Paint

2.2 Обработка события таймера

Для организации анимации в приложениях Windows часто используется класс Timer. Класс достаточно простой и после создания объекта класса программисту требуется только одно событие Tick, которое наступает когда указанный интервал таймера заканчивается. Интрвал устанавливается свойством Interval.

Существую еще два метода, необходимых для работы таймера: Start() – запускает таймер; Stop() – останавливает выполнение таймера.

3. Методика и порядок выполнения работы

3.1. Рисование графика функции

Для овладения практическими навыками визуализации результатов расчетов реализуйте пример: создайте приложение для отображения графика по заданной функции: $F = a \cdot x^{-p} \cdot \sin(k \cdot x + b)$.

Для решения поставленной задачи реализуйте следующие действия:

1. Создайте приложение Windows Forms.

2. Спроектируйте форму в соответствии с представленным шаблоном (рис. 9.2).

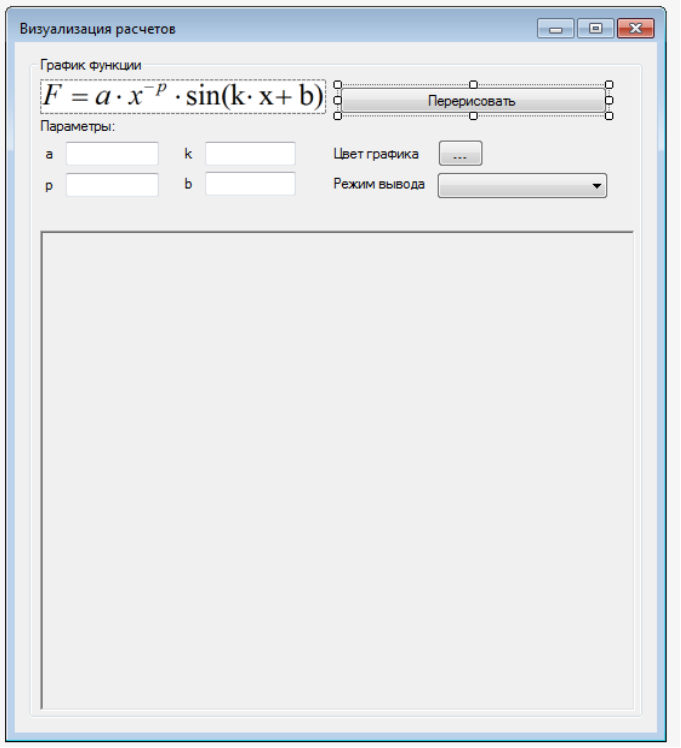


Рисунок 9.2 – Проект формы приложения

В таблице 9.1 приведена дополнительная информация о компонентах, размещенных на форме приложения.

Таблица 9.1 – Элементы управления формы главного окна

Имя элемента управления	Описание
txtA	Элемент управления типа TextBox. Предназначен для ввода коэффициента функции a.

txtP	Элемент управления типа TextBox. Предназначен для ввода коэффициента функции p.
txtK	Элемент управления типа TextBox. Предназначен для ввода коэффициента функции k.
txtB	Элемент управления типа TextBox. Предназначен для ввода коэффициента функции b.
btnChooseColor	Элемент управления типа Button. Предназначен для открытия диалога выбора цвета графика.
comboMode	Элемент управления типа ComboBox. Предназначен для выбора режима отображения графика: точками или в виде линии. Заполните свойство Items и добавьте два элемента: «Точки», «Линия».
panelGraf	Элемент управления типа Panel. Предназначен для отображения графика функции. Свойство Height установлено в 400.
btnRedraw	Элемент управления типа Button. Предназначен для перерисовки графика.

3. В теле класса главной формы приложения определите следующие свойства (рис. 9.3).

```
public partial class Form1 : Form
{
    // Цвет графика
    Color color = Color.Red;
    // Режим отображения
    int mode = 1;
    // Шаг расчета функции
    double dx = 5;
    // Интервал изменения x
    double x0 = 0.0, xn = 400.00;
    // Коэффициенты функции
    double a = 5, b = 100, p = -0.5, k = 2;
```

Рисунок 9.3 – Добавление полей класса

4. В конструктре экранной формы установите значения элементов управления в соответствии со значениями служебных переменных (рис. 10.4).

```
public Form1()
{
    InitializeComponent();

    // Инициализация компонентов формы значениями переменных
    // Выбор режима
    comboMode.SelectedIndex = mode;
    // Установка коэффициентов
    txtA.Text = a.ToString();
    txtB.Text = b.ToString();
    txtK.Text = k.ToString();
    txtP.Text = p.ToString();
}
```

Рисунок 9.4 – Конструктор класса

5. Реализуйте метод вычисления значений функции в соответствии с условием задачи (рис. 9.5).

```
private double MyFunc(double x)
{
    if (x == 0)
        return 0;
    else
        return a * Math.Pow(x, -p) * Math.Sin(k * x + b);
}
```

Рисунок 9.5 – Метод для вычисления значений функции

6. Реализуйте обработчики: нажатия кнопки «Перерисовать» (рис. 9.6), книпки выбора цвета графика (рис. 9.7) и выбора режима рисования графика (рис. 9.8).

```
private void btnRedraw_Click(object sender, EventArgs e)
{
    a = Convert.ToDouble(txtA.Text);
    b = Convert.ToDouble(txtB.Text);
    k = Convert.ToDouble(txtK.Text);
    p = Convert.ToDouble(txtP.Text);
    panelGraf.Invalidate();
}
```

Рисунок 9.6 – Обработчик нажатия кнопки btnRedraw

```
private void btnChooseColor_Click(object sender, EventArgs e)
{
    ColorDialog dlg = new ColorDialog();
    dlg.Color = color;
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        // еустанавливаем новый цвет
        color = dlg.Color;
        // перерисовываем панель с графиком
        panelGraf.Invalidate();
    }
}
```

Рисунок 9.7 – Обработчик нажатия кнопки btnChooseColor

```
private void comboMode_SelectedIndexChanged(object sender, EventArgs e)
{
    // Устанавливаем режим рисования графика
    mode = ((ComboBox)sender).SelectedIndex;
    // Перерисовываем панель
    panelGraf.Invalidate();
}
```

Рисунок 9.8 – Обработчик выбора пункта в comboMode

Внешний вид экранной формы разработанного приложения представлен на рис. 9.9

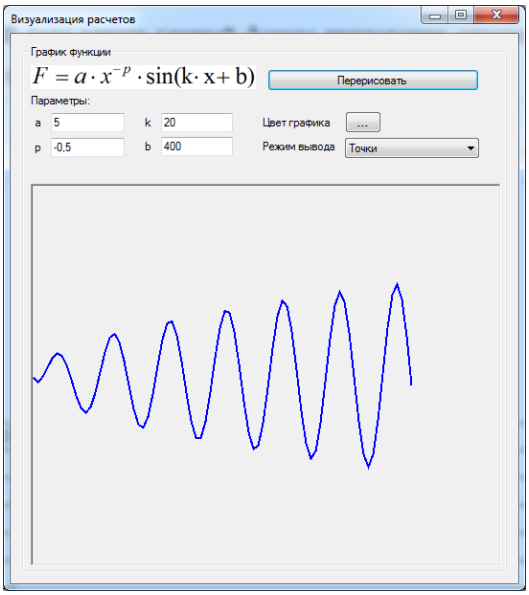


Рисунок 9.9 – Приложение в режиме выполнения

7. Запустите приложение. Протестируйте работу программы при различных значениях параметров.

8. Доработайте приложение: реализуйте механизм вывода графика отдельными точками, так как в представленном варианте график выводится в виде отрезков, соединяющих отдельные точки графика.

9. Разработайте приложение, в котором визуализируются данные сложной функции (учащийся может определить график функции самостоятельно).

3.2. Анимация средствами GDI+

1. Создайте приложение Windows Forms.

2. Дополните класс экранной формы вспомогательными переменными (рис. 9.10).

```
public partial class Form2 : Form
{
    private Point pos = new Point(100, 100);
    private int R = 50;
    double T = 0;
```

Рисунок 9.10 – Переменные класса формы

3. Перетащите из панели инструментов компонент Timer на экранную форму. Присвойте ему имя MainTimer. Установите свойство Interval в значение 500.

4. Модифицируйте обработчик события Paint экранной формы (рис. 9.11).

```
private void Form2_Paint(object sender, PaintEventArgs e)
{
    Graphics gr = e.Graphics;
    gr.DrawEllipse(new Pen(Color.Blue, 10), pos.X - R, pos.Y - R, R, R);
}
```

Рисунок 9.11 – Обработчик события Paint

5. Реализуйте обработчик события Tick таймера MainTimer (рис. 9.12).

```
private void MainTimer_Tick(object sender, EventArgs e)
{
    T += 0.1;
    pos.X = (int)(300 + 150 * Math.Sin(T));
    pos.Y = (int)(300 + 100 * Math.Cos(T));
    Invalidate();

}
```

Рисунок 9.12 – Обработчик события Tick

6. Для запуска и остановки таймера добавьте вызов метода Start() в конструктор класса, и вызов метода Stop() в обработчик события FormClosing главной формы (рис. 9.13).

```
public Form2()
{
    InitializeComponent();
    MainTimer.Start();
}
private void Form2_FormClosing(object sender, FormClosingEventArgs e)
{
    MainTimer.Stop();
}
```

Рисунок 9.13 – Запуск и остановка таймера

3.3. Интерактивная двумерная графика

Выполните и разберите технологию решения учебного задания: разработайте приложение, которое позволяет пользователю взаимодействовать с GDI-объектами. Необходимо вывести на форму простую фигуру и сделать доступным ее перетаскивание.

Для выполнения задания необходимо выполнить следующие действия:

1. Создайте приложение Windows Forms.

2. Добавить в класс формы служебные переменные (рис. 9.14).

3. Модифицируйте метод обработки события перерисовки окна (рис. 9.15).

```
public partial class Form1 : Form
{
    private Point location = new Point(50,50);
    private int R = 50;
    Boolean DropStarted = false;
```

Рисунок 9.14 – Служебные переменные класса

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    // Рисуем круг в точке location
    Graphics gr = e.Graphics;
    gr.DrawEllipse(new Pen(Color.Red, 5), location.X - R, location.Y - R, R, R);
}
```

Рисунок 9.15 – Обработчик Paint

4. Реализуйте обработчик события MouseDown для формы приложения (рис. 9.16).

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    int curX = e.X, curY = e.Y;
    Boolean InX = (curX >= location.X - R && curX <= location.X + R);
    Boolean InY = (curY >= location.Y - R && curY <= location.Y + R);

    if (InX && InY)
        DropStarted = true;
}
```

Рисунок 9.16 – Обработчик MouseDown

5. Реализуйте обработчик события MouseUp для формы приложения (рис. 9.17).

```
private void Form1_MouseUp(object sender, MouseEventArgs e)
{
    if (DropStarted)
    {
        location.X = e.X + R/2;
        location.Y = e.Y + R/2;
        Invalidate();
    }
}
```

Рисунок 9.17 – Обработчик MouseUp

6. Запустите приложение, протестируйте механизм перетаскивания объектов.

Варианты индивидуальных заданий

1. Создайте приложение Windows в среде Visual Studio.

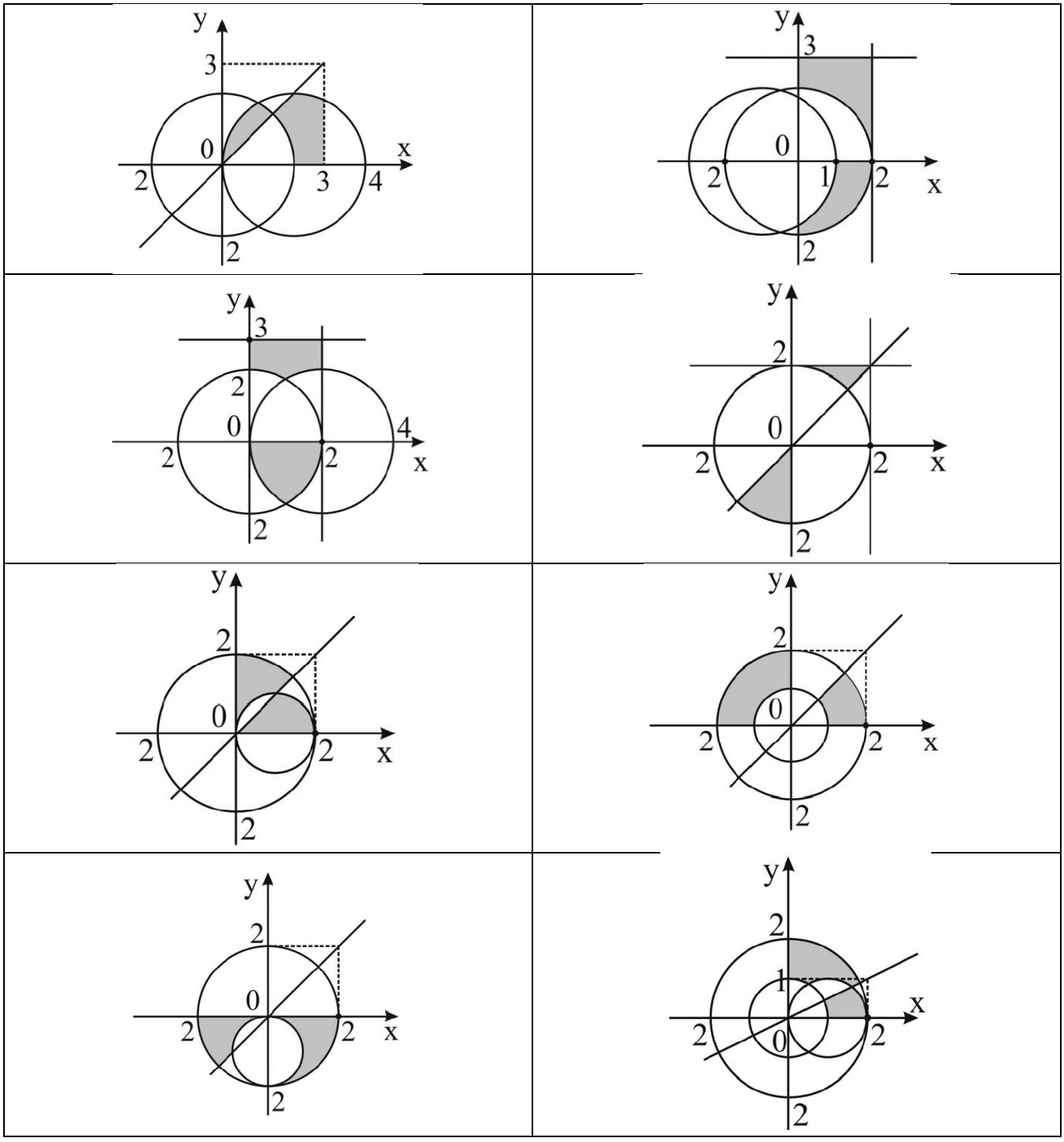
2. Выведите на поверхность формы рисунок, в соответствие с индивидуальным заданием.

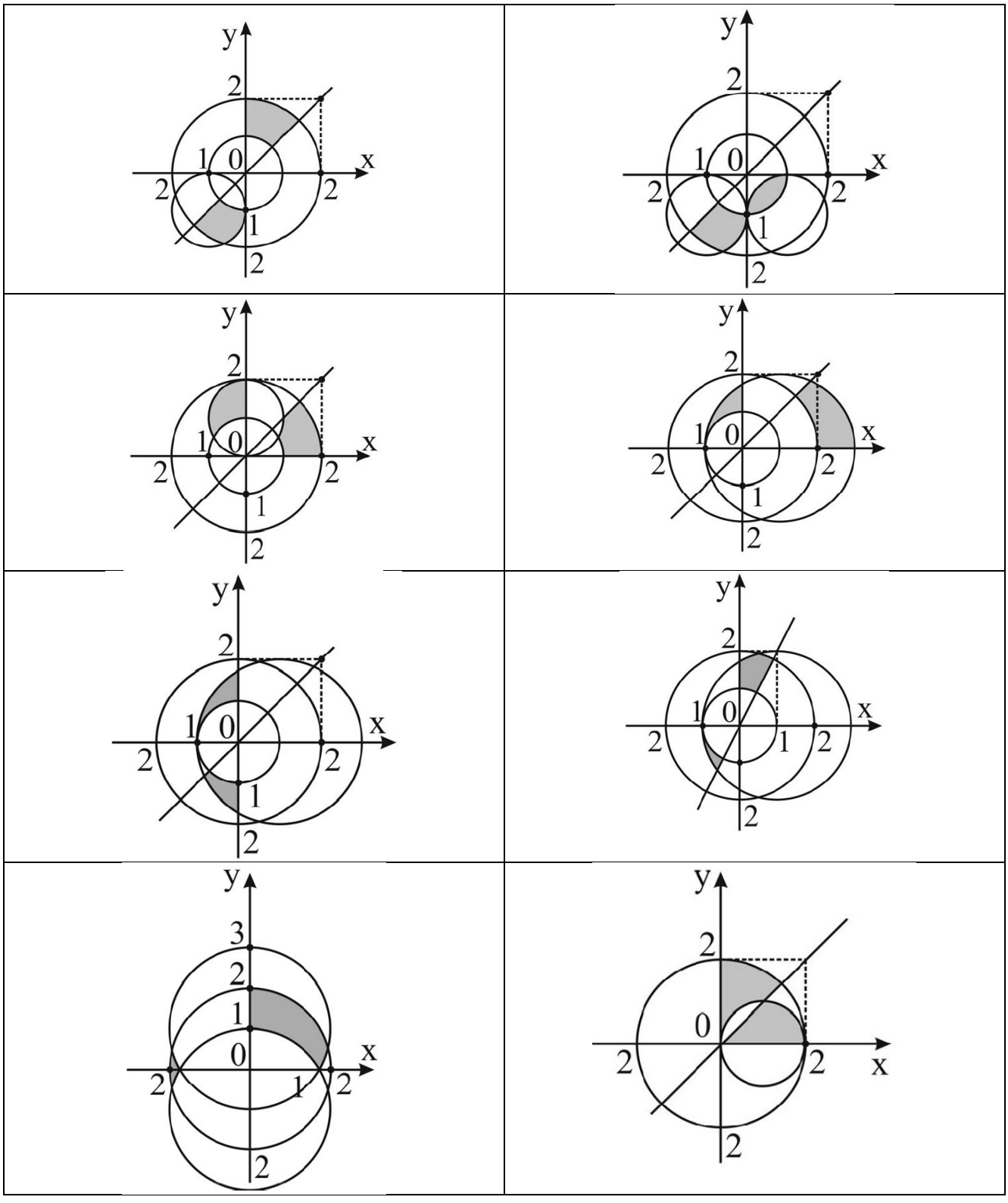
3. Реализуйте обработчик события изменения координат курсора мыши таким образом, чтобы при попадании курсора мыши в закрашенную область, соответствующий закрашенный регион менял свой цвет.

4. Реализуйте возможность выбора цвета цвета закрашенных областей с использованием стандартного диалога операционной системы для выбора цвета.

5. Реализуйте возможность перетаскивания пользователем подписей к рисункам (названия осей, цифры).

6. Реализуйте анимацию: циклическое изменение размеров шрифта подписей, анимацию цвета элементов и т.п. (что анимировать – выбирает учащийся). Необходимо предусмотреть возможность остановки и запуска анимации.





4. Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; цель и задачи лабораторной работы.

3. Ответы на контрольные вопросы.

4. Экранные формы и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения (в данной лабораторной работе необходимо привести листинг кода обработчиков событий, которые были изменены разработчиком).

Отчет о выполнении лабораторной работы с датой, подписанный студентом, сдается преподавателю. Студент защищает лабораторную работу, отвечает на вопросы преподавателя, касающиеся разработанного приложения.

5. Вопросы для защиты работы

1. Назовите основные типы, необходимые для работы с GDI+.

2. Какие пространства имен доступны для работы с графикой?

3. Опишите возможные конструкторы класса Point.

4. В чем состоит особенность конструктора класса Region?

5. Как получить доступ к объекту класса Graphics?

6. Опишите способ, с помощью которого можно реализовать перехват соытия мыши над фигурами GDI+.

7. Для чего предназначе класс GraphicPath?

8. Чем отличаются типы Point и PointF?

9. Какие элементы могут обрабатывать событие Paint?

10. Для чего используется тип PathPointType?

11. Через какой тип происходит рисование в обработчике события Paint?

12. Опишите механизм выбора объекта GDI.

13. Какие события участвуют в процессе интрактивного взаимодействия с GDI объектами графики?

14. Какое событие необходимо задействовать, если добавить в программу отображение контура объекта при перетаскивании?

15. Опишите механизм реализации анимации с использованием GDI+.

16. Опишите назначение основных свойств и методов класса Timer.

17. Для чего предназначено событие Tick класса Timer?

18. Сколько таймеров может присутствовать в рпограмме?

СПИСОК ЛИТЕРАТУРЫ

Книги

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.

2. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.

3. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.

4. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.

5. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.

6. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.

7. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: https://urait.ru/bcode/437489 (дата обращения: 13.02.2020).

Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87530.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей

Перечень Интернет-ресурсов

1. https://github.com/enikolaev/IT-and-Programming - репозиторий с исходными колами для дисциплины.
2. https://www.w3schools.com/python/default.asp - W3C Tutorials.

Учебное пособие (лабораторный практикум) по дисциплине

«Информационные технологии и программирование»

Часть 1 (лабораторные работы 1-11)

для студентов направления

09.03.02 «Информационные системы и технологии»

Составитель: Е.И. Николаев