

АДМИНИСТРАЦИЯ ЗАВОДОУКОВСКОГО ГОРОДСКОГО ОКРУГА
ДЕПАРТАМЕНТ ПО СОЦИАЛЬНЫМ ВОПРОСАМ

МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ЗАВОДОУКОВСКОГО ГОРОДСКОГО ОКРУГА
«ЗАВОДОУКОВСКАЯ СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 2»

Разработка программ по решению задач статистического анализа

Автор: Некрасов Игорь Сергеевич

муниципальное автономное
общеобразовательное учреждение
Заводоуковского городского
округа «Заводоуковская средняя
общеобразовательная школа № 2»,
6 «А» класс

Научный руководитель:

Уразаева Диана Дамировна,

учитель математики и
информатики,

муниципальное автономное
общеобразовательное учреждение
Заводоуковского городского
округа «Заводоуковская средняя
общеобразовательная школа № 2»

Разработка программ по решению задач статистического анализа

Некрасов Игорь Сергеевич

Россия, Тюменская область, Заводоуковский городской округ, город Заводоуковск,
Муниципальное автономное общеобразовательное учреждение Заводоуковского
городского округа

«Заводоуковская средняя общеобразовательная школа № 2», 5 «А» класс.

Аннотация

Я всегда интересовался точными науками, такими как информатика и математика. Зная это, мама обратилась ко мне за помощью в проведении анализа при работе с электронными таблицами. К сожалению, я не мог выполнить все необходимые расчеты, и каждый раз обращался к ресурсам сети Интернет, чтобы изучить работу с той или иной функцией. И я задался вопросом: «А можно ли автоматизировать данный процесс, чтобы не приходилось каждый раз составлять новые функции?».

Цель исследования:

Разработка программ по выполнению расчетов над статистическими данными результатов проверочных и экзаменационных работ учащихся для использования учителем в проведении анализа.

Задачи исследования:

1. Исследовать функционал языка программирования Python в области применения к задачам статистического анализа.
2. Разработать программы по выполнению статистических расчётов.
3. Составить инструкции по использованию программ в помощь учителю.
4. Оказать помощь учителям в выполнении статистических расчетов на базе МАОУ «СОШ № 2» г. Заводоуковск.

Методы исследования:

1. *Описательный метод.* Включает в себя сбор информации, анализ возможностей языка программирования Python, беседа с учителями.
2. *Метод проектирования.* Способствует проведению подбора статистических задач к набору программ и функций по их выполнению.
3. *Экспериментальный метод.* Позволит проверить эффективность разработанного материала при его применении учителями в проведении анализа.

Гипотеза: мы предположили, что программы по выполнению статистических расчетов поспособствуют эффективному комплексному анализу результатов проверочных и экзаменационных работ, а также сократит время, уделяемое учителем, на данную часть работы.

В ходе проведённой работы разработали набор программ и инструкцию по их использованию.

Разработка программ по решению задач статистического анализа

Некрасов Игорь Сергеевич

Россия, Тюменская область, Заводоуковский городской округ, город Заводоуковск,
Муниципальное автономное общеобразовательное учреждение Заводоуковского
городского округа

«Заводоуковская средняя общеобразовательная школа № 2», 5 «А» класс.

Научная статья.

Глава 1. Общая характеристика языка программирования Python.

Python является высокоуровневым объектно-ориентированным языком программирования, использование которого делает возможным выполнение различных задач. Данный язык ориентирован на повышение производительности разработчика и читаемости кода.

Стандартная библиотека Python содержит большой объём функций, а его синтаксис достаточно прост в изучении и использовании. Именно поэтому данный язык является наиболее удобным для тех, кто только начинает изучать программирование.

Python отличается рядом характеристик: 1) код написанный на данном языке программ легко читаем благодаря чёткому и последовательному синтаксису; 2) есть возможность объединить язык с файлами различных типов; 3) массив может включать объекты разных типов; 4) значение любого типа может быть назначено переменной. [1]

В данном языке программирования используется 6 видов оператора:

1. Условные операторы: if – если; else – иначе, elif (else if) используется, когда условий и альтернатив несколько.
2. Операторы цикла: while (пока), for (для).
3. Оператор определения класса class
4. Оператор определения функции, метода или генератора def.

2. Оператор обработки исключений try – except – else или try-finally. [2]

Помимо стандартной библиотеки, Python содержит множество других библиотек, каждая из которых имеет своё предназначение. Рассмотрим некоторые из них:

1. Matplotlib – визуализация данных, которая позволяет предоставить информацию в наглядном виде и изучить более подробно, а также доступно изложить другим людям.
2. Dora – библиотека, которая предназначена для анализа основных свойств данных и нахождения в них общих закономерностей.

3. `Openpyxl` – это библиотека Python для чтения и записи файлов Excel (с расширением `xlsx` / `xlsm` / `xltx` / `xltm`). [3]

В настоящее время Python используется большим количеством программистов во многих странах мира. К одним из основных причин такой распространенности языка является то, что работа осуществляется одинаково эффективно на различных операционных системах. Функционал данного языка позволяет разрабатывать приложения любого объёма.

Таким образом, Python является наиболее оптимальным вариантом при решении задач по статистике, так как данный язык более прост в самостоятельном изучении, а также отвечает всем необходимым требованиям.

Глава 2. Чтение и запись файлов с использованием библиотеки `Openpyxl`

В сфере образования посредством статистических вычислений проводится анализ деятельности учебных учреждений, например, подготовки учащихся к сдаче Единого государственного экзамена. Результаты подобных исследований используют для практических выводов. Часто перед учителями стоят задачи по извлечению каких-то данных из файлов формата XLS без предварительной технической подготовки.

В соответствии с формой представления данных и статистическими величинами (максимальный балл, среднее значение баллов по списку учащихся; количество учащихся, выполнивших работу на ту или иную оценку и т.д.), которые необходимо извлечь и провести вычисления, для разработки программ выбрана библиотека `Openpyxl`.

`Openpyxl` – это библиотека Python для чтения и записи файлов Excel. Данная библиотека позволяет считывать данные и выполнять статистические вычисления на основе данных из электронной таблицы [4]. Рассмотрим основные возможности библиотеки `Openpyxl`:

1. Определение файла, с которого будут считываться значения. Для выполнения данного действия необходимо воспользоваться функцией `load_workbook`.

Например: `wb = load_workbook('ege_result.xlsx')`

2. Определение листа, с которого будут считываться значения. Для реализации данного действия необходимо добавить строку следующего вида:

`ws = wb['Индивидуальные результаты']`, где в квадратных скобках содержится название листа.

3. Определение ячейки, с которой необходимо вывести числовое значение. Чтобы обозначить данную ячейку в кодовой программе, после того, как определили

файл и лист, необходимо ввести следующую строку: `ws['E' + str(i)].value`, где E – столбец, i – переменная, от которой зависит номер строки.

4. Функции `statistic.mode()`, `statistic.median()`. Данные функции позволяют вычислить значения моды и медианы из числового ряда [5].

Владея базовыми знаниями и умениями по программированию на языке Python, хорошей математической подготовкой, можно использовать вышеуказанные функции для разработки эффективных программ по выполнению статистических расчетов. Для вычисления таких величин как значение максимального количества баллов за проверочную работу, общего количества учащихся и т.д., дополнительно составляются формулы на основе встроенных функций Python.

Таким образом, библиотека `Openpyxl` содержит все необходимые функции для выполнения необходимых статистических расчетов, а также позволяет структурировать данные в целях облегчения анализа.

Практическая часть.

В 5 классе по предмету «информатика» мы начали изучать работу с электронными таблицами, составляли формулы и функции по проведению расчетов, строили графики и диаграммы. Данные знания дали мне возможность помочь маме в проведении анализа при работе с электронными таблицами. Я начал искать на ресурсах сети Интернет программы, о которых говорил учитель информатики. Но, в основном, я находил информацию либо только об отдельных функциях, либо программы, которые позволяют выполнить только часть статистических расчетов. Изучив данные программы, понял, что не смогу с их помощью провести все вычисления, которые необходимы при анализе данных по результатам работы. Я обратился к учителю информатики. Учитель рассказал мне о возможных способах: создание электронной таблицы со встроенными функциями и разработка специальных программ, позволяющих провести статистические расчеты посредством чтения данных из электронной таблицы. Меня заинтересовал второй вариант, так как я уже на тот момент изучал язык программирования Python и хотел рассмотреть его дополнительные возможности.

Мы провели беседу с учителями, чтобы узнать, сколько времени занимает у них работа с электронными таблицами и проведение расчетов. И выяснили, что не все учителя могут самостоятельно выполнить все необходимые вычисления, часть из них обращаются за помощью друг к другу, либо проводят вычисления вручную. Также я узнал и о статистических величинах, которые необходимо вычислить для проведения эффективного анализа результатов работы учителем.

Таким образом, мы решили разработать набор программ, которые позволят учителю любого предмета провести комплексный анализ результатов их работы, затрачивая при этом минимум времени.

Мы разработали 10 программ для статистических расчётов (Приложение I).

Пример реализации программы.

Формулировка задачи: «Определить, какую часть от общего числа, составляют учащиеся, получившие менее минимального балла. Полученное значение округлить до 2 знаков после запятой (минимум: 10 баллов)».

Этапы решения задачи:

1. Импорт функции из библиотеки, обеспечивающей возможности работы с файлами формата *xlsx*, для прочтения данных:

```
from openpyxl import load_workbook
```

2. Определение файла и листа, с которых будут считываться данные:

```
wb = load_workbook('Контрольная работа.xlsx')
```

```
ws = wb['Индивидуальные результаты']
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1				Максимальный балл	1	1	2	2	2	2	3	3	3	1	2	3
2	Результат	Баллы	Учащийся	Оценка	1. Типы д	2. Создан	3. Услови	4. Резуль	5. Сортир	6. Цена т	7. Как на	8. Как на	9. Как на	10. Основ	11. Класс	12. Ключевое
3	44%	11	Аванян Арина		1	0	2	2	0	0	0	0	0	1	2	3
4	72%	18	Басаков Дмитрий		1	0	2	2	0	2	3	0	3	0	2	3
5	100%	25	Борзых Олеся		1	1	2	2	2	2	3	3	3	1	2	3
6	54%	13,6	Голенков Илья		1	0	2	2	0	2	0	0	3	0	0,6	3
7	84%	21	Желнин Никита		1	0	2	2	2	2	3	0	3	1	2	3
8	80%	20	Исаев Максим		1	1	2	2	2	2	3	0	3	1	0	3
9	37%	9,3	Кайгородов Евгений		1	0	0	2	0	2	3	0	0	1	0,3	0
10	25%	6,3	Колобнев Никита		0	0	2	0	0	0	0	0	3	1	0,3	0
11	85%	21,2	Костюк Полина		1	1	2	2	2	2	3	0	3	1	1,2	3
12	25%	6,3	Кунгурцев Александр		0	0	0	0	0	2	3	0	0	1	0,3	0
13	70%	17,6	Медведев Владимир		1	0	2	0	2	2	3	0	3	1	0,6	3
14	25%	6,2	Шаварёв Саша		0	0	0	0	0	2	3	0	0	0	1,2	0
15				Максимальный балл	12	12	24	24	24	24	36	36	36	12	24	36
16				Набранные баллы	9	3	18	16	10	20	27	3	24	9	12,5	24
17				Процент	75%	25%	75%	66,67%	41,67%	83,33%	75%	8,33%	66,67%	75%	52,08%	66,67%

Рис 1. Определение листа и файла

3. Подсчет количества учащихся:

```
rows = len(list(ws.rows))-2
```

Исходя из данных файла, вычитаем 2, так как в первых двух строках не содержатся данные по учащимся:

```
rows = 17 - 2 = 15
```

	A	B	C	D
1				Максимальная
2	Результат	Баллы	Учащийся	Оценка
3	44%	11	Аванян Арина	
4	72%	18	Баскаков Дмитрий	
5	100%	25	Борзых Олеся	
6	54%	13,6	Голенков Илья	
7	84%	21	Желнин Никита	
8	80%	20	Исаев Максим	
9	37%	9,3	Кайгородов Евгений	
10	25%	6,3	Колобнев Никита	
11	85%	21,2	Костюк Полина	
12	25%	6,3	Кунгурцев Александр	
13	70%	17,6	Медведев Владимир	
14	25%	6,2	Шаварёв Саша	
15				Максимальная
16				Набранных
17				Процент

Рис. 2 Подсчет количества учащихся

4. Заводим счетчик, с помощью которого вычисляется количество учащихся, получивших менее 10 баллов:

$$a = 0$$

5. Включаем цикл для перебора всех строк, начиная с третьей (первые две строки не включают данные по итоговым баллам) и определяем столбец, содержащий данные по количеству баллов:

for i in range (3, rows):

$$k = \text{int}(\text{ws}['B' + \text{str}(i)].\text{value})$$

	A	B	C	D
1				Максимальная
2	Результат	Баллы	Учащийся	Оценка
3	44%	11	Аванян Арина	
4	72%	18	Баскаков Дмитрий	
5	100%	25	Борзых Олеся	
6	54%	13,6	Голенков Илья	
7	84%	21	Желнин Никита	
8	80%	20	Исаев Максим	
9	37%	9,3	Кайгородов Евгений	
10	25%	6,3	Колобнев Никита	
11	85%	21,2	Костюк Полина	
12	25%	6,3	Кунгурцев Александр	
13	70%	17,6	Медведев Владимир	
14	25%	6,2	Шаварёв Саша	
15				Максимальная
16				Набранных
17				Процент

Рис.3 Перебор строк и определение столбца

6. В цикл включаем условие (если количество баллов меньше, то к счетчику прибавляется единица):

if k < 10:

a=a+1;

	Максимальная	Учащийся	Оценка
Баллы			
a=0	11	Аванян Арина	
a=0	18	Баскаков Дмитрий	
a=0	25	Борзых Олеся	
a=0	13,6	Голенков Илья	
a=0	21	Желнин Никита	
a=0	20	Исаев Максим	
a=1	9,3	Кайгородов Евгений	
a=2	6,3	Колобнев Никита	
a=2	21,2	Костюк Полина	
a=3	6,3	Кунгурцев Александр	
a=3	17,6	Медведев Владимир	
a=4	6,2	Шаварёв Саша	

Рис. 4 Работа с условием

7. Вычисляем часть учащихся (в процентах), получивших менее минимального балла:

*s=a*100/rows*

По данным файла получаем:

*s = 4*100 / 15 = 26,6666666667*

8. Округляем полученные данные до сотых посредством функции *round*:

l= round(s,2)

Получаем: 26,6666666667≈26,67

9. Выводим полученный результат:

print ('Количество:', s)

Также мы разработали инструкцию по работе с программами (Приложение III).

Разработанные программы позволяют педагогу любого предмета провести ряд статистических вычислений. Программы содержат подробные комментарии к каждому этапу, что делает удобным их применение в работе с файлом формата XLS, содержащего итоговые данные по выполнению той или иной проверочной работы.

Также провели апробацию программ на основе статистических данных результатов контрольной работы по алгебре 9 «В» класса и **рпав**. Апробация прошла успешно, учителя ознакомились с инструкцией и самостоятельно провели ряд статистических расчётов.

Выводы.

1. Python является наиболее оптимальным вариантом при решении задач по статистике, так как данный язык более прост в самостоятельном изучении, а также

отвечает всем необходимым требованиям. Библиотека `Openpyxl` содержит все функции для выполнения необходимых статистических расчетов, а также позволяет структурировать данные в целях облегчения анализа.

2. Дополнительные возможности языка программирования Python в области решения задач статистического анализа позволили разработать набор программ, выполняющих большой объем статистических расчетов. Апробация с учителями разных предметов прошла успешно. Мы высчитали значения различных статистических величин, что позволило провести комплексный анализ результатов работ учащихся, затратив при этом гораздо меньше времени.

3. Разработанные инструкции по использованию программ в помощь учителю, позволяют применять наши программы в собственной работе.

4. На практике выяснили, что разработанные нами программы успешно используются учителями МАОУ «СОШ № 2».

При разработке программ выявлены статистические данные (средний уровень выполнения заданий по определенному разделу, данные по нескольким лучшим работам и т.д.), для вычисления которых необходимо дальнейшее изучение синтаксиса, циклов и функций языка программирования Python. В ближайшем будущем я собираюсь работать над разработкой новых программ по вычислению этих данных.

Библиографический список.

1. Язык программирования Python [Электронный ресурс]. Режим доступа: http://inf-w.ru/?page_id=4293. Дата обращения: 03.11.2021.
2. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2017. – 992 с.
3. Стандартные библиотеки Python [Электронный ресурс]. Режим доступа: <https://losst.ru/standartnye-biblioteki-python>. Дата обращения: 11.01.2022.
4. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 118 с.
5. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2017. – 92 с.

ПРИЛОЖЕНИЕ I

Набор программ для реализации статистических расчетов.

Программа № 1

1. Формулировка задачи: «Посчитать количество учащихся, принимающих участие в проверочной работе».

2. Код программы:

```
from openpyxl import load_workbook #подключение хранилища данных,
содержащего инструменты для считывания данных

wb = load_workbook('Анализ ВПР.xlsx') # считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows))-1 #подсчет количества учащихся.
print ('Количество экзаменуемых:', rows) #вывод данных
```

Программа № 2

1. Формулировка задачи: «Определить количество учащихся, выполнивших работу на 100 баллов».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

wb = load_workbook('Анализ работы.xlsx') # считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows))-1 #подсчет количества учащихся.
a=0; # заводим счетчик (переменную, которая поможет подсчитать количество
стобалльников)

for i in range(2,rows): # перебор всех строк, содержащих данные по количеству
баллов

    k = int(ws['B' + str(i)].value); # определение столбца, содержащего данные по
количеству баллов (если данные по итоговым баллам содержатся в другом столбце,
необходимо изменить номер столбца в кодовой строке)

    if k == 100: # если количество баллов равно 100, то к счетчику прибавляется
единица
```

```
a=a+1;
print ('Количество:', a) # вывод данных
```

Программа № 3

1. Формулировка задачи: «Определить количество учащихся, не получивших ни одного балла за задания из второй части».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

wb = load_workbook('Анализ работы.xlsx') # считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows)) -1 # подсчет строк

for i in range(2, rows): # цикл для перебора всех строк, содержащих данные по
количеству баллов

    k = 0 # заводим счетчик

    if str(ws['F' + str(i)].value) == '0(3)0(2)0(3)0(4)': # постановка условия для подсчета
количества учащихся, выполнивших работу на 100 баллов (в зависимости от критериев
оценивания работы необходимо изменить количество заданий и соответствующих баллов
в кодовой строке).

        k+=1

print(k)
```

Программа № 4

1. Формулировка задачи: «Определить, какую часть от общего числа, составляют учащиеся, получившие менее минимального балла. Полученное значение округлить до 2 знаков после запятой (минимум: 40 баллов)».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

wb = load_workbook('Анализ работы.xlsx') # считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows))-1 # подсчет строк

a=0 # заводим счетчик
```

for i in range(2, rows): # цикл для перебора всех строк, содержащих данные по количеству баллов

k = int(ws['B' + str(i)].value); # определение столбца, содержащего данные по количеству баллов (если данные по итоговым баллам содержатся в другом столбце, необходимо изменить номер столбца в кодовой строке)

if k < 40: # если количество баллов меньше, то к счетчику прибавляется единица (при необходимости количество минимальных баллов можно скорректировать в кодовой строке)

a=a+1;

s=a*100/ (rows-1); #формула для вычисления части экзаменуемых, получивших менее минимального балла

l= round(s,2) #округление полученных данных

print ('Количество:', s) #вывод результата

Программа № 5

1. Формулировка задачи: «Найти среднее значение баллов по заданию №7».

2. Код программы:

from openpyxl import load_workbook # подключение хранилища данных, содержащего инструменты для считывания данных

import statistics # подключение функции, содержащей встроенные формулы по выполнению статистических вычислений

wb = load_workbook('Анализ работы.xlsx') #считывание данных с файла (в скобках необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows))-1 # подсчет строк

for i in range(2, rows): # цикл для перебора всех строк, содержащих данные по количеству баллов

marks2 = str(ws['D' + str(i)].value) # определение столбца, содержащего данные по количеству баллов (если данные по заданию содержатся в другом столбце, необходимо изменить номер столбца в кодовой строке)

m = sum(marks2)/len(marks2) # вычисление среднего значения (сумма баллов по заданию делится на количество учащихся)

print (m) #вывод данных

Программа № 6

1. Формулировка задачи: «Определить максимальное количество набранных баллов по проверочной работе».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

import statistics # подключение функции, содержащей встроенные формулы по
выполнению статистических вычислений

wb = load_workbook('Анализ работы.xlsx') #считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows))-1 # подсчет строк

for i in range(2, rows): # цикл для перебора всех строк, содержащих данные по
количеству баллов

    marks2 = str(ws['J' + str(i)].value) # определение столбца, содержащего данные
по количеству итоговых баллов (если данные по итоговым баллам содержатся в другом
столбце, необходимо изменить номер столбца в кодовой строке)

    m = max (marks2) # определение максимального значения из столбца, содержащего
данные по итоговым баллам

print (m) #вывод данных
```

Программа № 7

1. Формулировка задачи: «Найти размах между наибольшим и наименьшим количеством итоговых баллов по проверочной работе».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

import statistics

wb = load_workbook('Анализ работы.xlsx') #считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

rows = len(list(ws.rows))-1 # подсчет строк

for i in range(2, rows): # цикл для перебора всех строк, содержащих данные по
количеству баллов
```

```
marks2 = str(ws['B' + str(i)].value) # определение столбца, содержащего данные
по количеству итоговых баллов (если данные по итоговым баллам содержатся в
другом столбце, необходимо изменить номер столбца в кодовой строке)

m = max (marks2) # определение максимального значения из столбца, содержащего
данные по итоговым баллам

l = min (marks2) # определение минимального значения из столбца, содержащего
данные по итоговым баллам

s = m-l # нахождение размаха

print (s) # вывод данных
```

Программа № 8

1. Формулировка задачи: «Необходимо по всем заданиям из первой части определить, какие результаты встречались чаще всего».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

import statistics # подключение функции, содержащей встроенные формулы по
выполнению статистических вычислений

wb = load_workbook('Анализ работы.xlsx') #считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

len(list(ws.rows))-1 # подсчет строк

for i in range (2,rows): # цикл для перебора всех строк, содержащих данные по
количеству баллов

    mark = str(ws['E' + str(i)].value) # определение столбцов, содержащих данные по
баллам из первой части

    k = list(mark) # представление баллов из первой части в виде списка

    m = statistics.mode(k) # вычисление моды (результата, который встречался чаще
всего)

print (k) # вывод данных
```

Программа № 9

1. Формулировка задачи: «Найти сумму баллов заданий из первой части».

2. Код программы:

```
from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных
```

```

def(a, b): # функция для подсчета суммы баллов за задания из первой части
    r = 0 # счетчик (переменная, которая помогает просуммировать баллы)
    for j in range(len(a)): #цикл для перебора всех строк, содержащих данные по
количеству баллов
        r += 1 if a[j] >0 else 0
    return r

wb = load_workbook('Анализ работы.xlsx') #считывание данных с файла (в скобках
необходимо указать название вашего файла)

ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)

len(list(ws.rows))-1 # подсчет строк

result = 0 # счетчик, который позволяет просуммировать данные

for i in range(2, rows): # цикл для перебора всех строк, содержащих данные по
количеству баллов

    marks = str(ws['E' + str(i)].value) # определение столбцов, содержащих данные по
фамилиям учащихся

    marks2 = str(ws['F' + str(i)].value) # определение столбцов, содержащих данные по
сумме баллов из первой части

    result += f(marks, marks2) # суммирование данных

print(result) #вывод данных

```

Программа № 10

Формулировка задачи: «Найти количество учащихся, выполнивших более половины заданий».

Код программы:

```

from openpyxl import load_workbook # подключение хранилища данных,
содержащего инструменты для считывания данных

def (a, b): # функция для подсчета суммы баллов за задания из первой части
    r = 0 # счетчик для заданий из первой части (переменная, которая помогает
просуммировать баллы)
    k = 0 # счетчик для заданий из второй части (переменная, которая помогает
просуммировать баллы)
    for j in range(len(a)): # функция для подсчета суммы баллов за задания из первой
части
        r += 1 if a[j] == '+' else 0
        k += float(b[0])

```

```

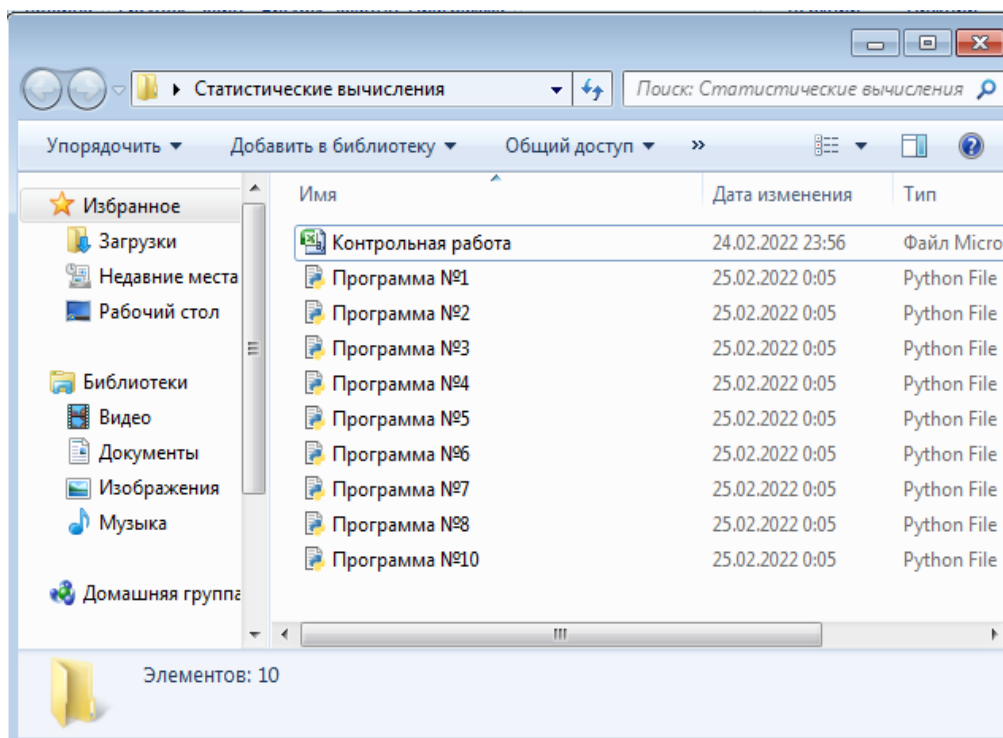
        k += float(b[4])
        k += float(b[8])
        k += float(b[12])
    return (r, k)

def (a, b): # функция для подсчета выполненных заданий
    r = 0
    k = 0
    for j in range(len(a)):
        r += 1 if a[j] == '+' else 0
        k += 1 if float(b[0]) == 3 else 0
        k += 1 if float(b[4]) == 2 else 0
        k += 1 if float(b[8]) == 3 else 0
        k += 1 if float(b[12]) == 4 else 0
    return r + k

wb = load_workbook('Анализ работы.xlsx') # считывание данных с файла (в скобках
необходимо указать название вашего файла)
ws = wb['Индивидуальные результаты'] # определение листа из файла для
считывания данных (в скобках необходимо указать название вашего листа)
len(list(ws.rows))-1 # подсчет строк
count = 0 # счетчик для количества выполненных заданий (
for i in range(2, rows): # проход по всем строкам, начиная со второй
    marks = str(ws['E' + str(i)].value) # оценки за первую часть
    marks2 = str(ws['F' + str(i)].value) # оценки за вторую часть
    res = f(marks, marks2) # отбор данных по выполненным данным за обе части
    if g(marks, marks2) > 13: # если количество выполненных заданий больше 13,
то к счетчику прибавляем 1
        count += 1
print(count)

```

ПРИЛОЖЕНИЕ II



Расположение файлов

ПРИЛОЖЕНИЕ III

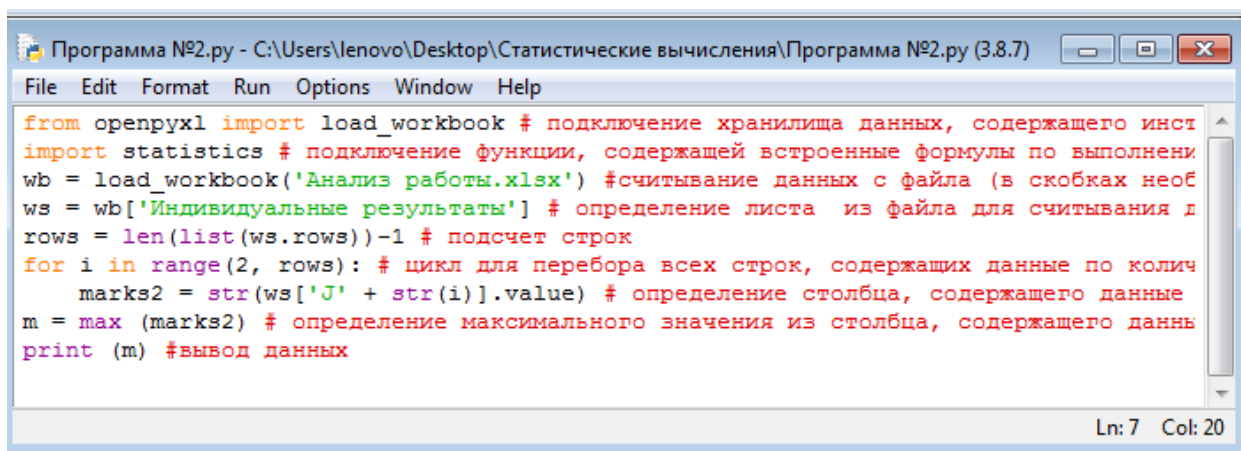
Инструкция по работе с программами

Для корректной работы программ и получения достоверных вычислений необходимо расположить их в одной папке. В данную папку также необходимо переместить файл, содержащий результаты проверочной работы (Приложение II).

При работе педагогу необходимо корректировать данные по названию файла, соответствующего листа, номерам столбцов. Каждая программа позволяет выполнить определенные статистические расчёты: №1 – количество учащихся; №2 – количество учащихся, выполнивших работу на максимальное количество баллов; №3 – количество учащихся, не получивших ни одного балла за ту или иную часть работы; №4 – часть учащихся, получивших менее минимального балла; №5 – среднее значение баллов по конкретному заданию; №6 – максимальное количество набранных баллов; №7 – размах между наибольшим и наименьшим количеством набранных баллов; №8 – результат в баллах, который встречался чаще всего; №9 – сумма баллов за задания определенной части; №10 – количество учащихся, выполнивших более половины заданий.

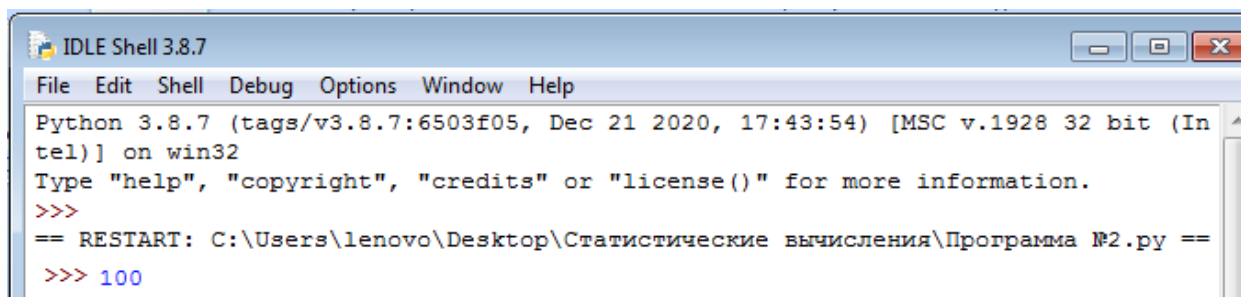
Рассмотрим пример работы с программой №2.

1. В папке, содержащей файлы, необходимо открыть программу №2 двумя щелчками мыши. Открывается окно приложения, содержащее код программы и указатели по выполнению тех или иных действий.



Окно приложения

2. Далее переходим по указателю «Run» и выбираем опцию «Run Module», после чего открывается окно с результатом выполнения программы.



Результат выполнения программы