# Постановка задачи

## Описание предметной области

В анкетах сотрудников некоторого учреждения имеются следующие сведения: фамилия, год рождения, должность (мастер, инженер, рабочий и т.д.),

стаж, образование. Вывести фамилии мастеров с высшим образованием и вычислить их средний стаж. Выполнить сортировку списка.

Данные анкеты хранятся в таблице Excel.

## Функции программы

Программа предназначена для анализа имеющихся табличных данных о сотрудниках учреждения. Программа должна производить сортировку и вывод данных с заданными параметрами. А также считать среднее значение.

## Входные, выходные данные

В качестве входных данных принимается, заполненная таблица Excel. Выходными данными является отсортированный список фамилий и дробное число.

## Ограничения

1. Ограничения, накладываемые на входные данные
2. Данные таблицы должны иметь следующий формат:

* Фамилия – строка;
* Год рождения – целое положительное число в промежутке от 1900 до 2003;
* Должность – строка;
* Стаж работы – целое положительное число, показывающее количество полных, проработанных лет;
* Образование – строка;

1. Все строковые данные должны быть написаны русскими буквами;
2. Таблица Excel должна находиться в том же каталоге, где находится основная программа;
3. Ограничения, накладываемые на выходные данные
4. Список фамилий должен быть отсортирован по алфавиту и быть пронумерован, начиная с 1;
5. Среднее значение для стажа работы должно быть выведено с точностью 1 знак, после запятой;

## Вид приложения, среда разработки

Приложение будет иметь вид консольного приложения и будет разрабатываться на языке Python, в среде разработки JetBrains PyCharm Community Edition 2019 x64.

1. **Разработка программы**

## Разработка структуры приложения

### Рисунок структуры приложения

### Спецификация

* read\_excel – данная функция предназначена для того, чтобы получить данные из заполненной Excel таблицы (см. Таблица 1).

Таблица 1

Описание функции read\_excel

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Назначение** | **Тип** | **Структура** |
| **Входные данные** | | | |
| path | Название таблицы Excel | String | Строковая переменная |
| **Промежуточные данные** | | | |
| excel\_data | Получение данных из Excel | xlrd.book.Book | Переменная класса |
| sheet | Получение данных из текущей книги Excel | xlrd.sheet.Sheet | Переменная класса |
| **Выходные данные** | | | |
| values | Получение не пустых данных из строк таблицы | List | 2-мерынй массив |

* get\_average – данная функция предназначена для подсчета среднего значения (см. Таблица 2).

Таблица 2

Описание функции get\_average

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Назначение** | **Тип** | **Структура** |
| **Входные данные** | | | |
| values | Хранит данные из Excel | List | 2-мерный массив |
| **Промежуточные данные** | | | |
| sum | Суммирование числе | Целочисленный | Переменная |
| value | Служит для перебора значений из таблицы | List | 1-мерный массив |
| **Выходные данные** | | | |
| sum/len(values) | Подсчет среднего значения | Float | Выражение |

* output – данная функция предназначена для форматированного вывода выходных данных программы (см. Таблица 3)

Таблица 3

Описание функции output

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Назначение** | **Тип** | **Структура** |
| **Входные данные** | | | |
| values | Хранит данные из Excel | List | 2-мерный массив |
| **Промежуточные данные** | | | |
| list\_of\_surnames | Список фамилий сотрудников | List | 1-мерный список |
| i | Счетчик цикла | Целочисленный | Переменная |
| surname | Фамилия конкретного сотрудника | String | Переменная |

* main – основная функция программы, в которой происходит диалог с пользователем и из который происходит вызов нужных функций
* check\_excel\_values – функция, предназначенная для проверки значений таблицы Excel (см. Таблица 4)

Таблица 4

Описание функции check\_excel\_values

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Назначение** | **Тип** | **Структура** |
| **Входные данные** | | | |
| values | Хранит данные из Excel | List | 2-мерный массив |
| **Выходные данные** | | | |
| 1, None | Если все данные соответствуют ограничениям, возвращается 1, если какое-либо из ограничений нарушено, выводится предупреждение | Boolean | Возвращаемое значение |

## Разработка схемы алгоритма

* Блок схема функции main (см. Рисунок 1)



Рисунок 1. Блок схема функции main

* Блок схема функции read\_excel (см. Рисунок 2)



Рисунок 2. Блок-схема функции read\_excel

* Блок схема функции get\_average (см. Рисунок 3).



Рисунок 3. Блок схема функции get\_average

* Блок схема функции output (см. Рисунок 4)



Рисунок 4. Блок-схема функции output

## Разработка пользовательского интерфейса

Программа предназначена для сотрудников учреждения. При запуске программы, будет выведено сообщение с предложением ввести имя файла, который необходимо обработать (см. Рисунок 5).



Рисунок 5. Диалоговое сообщение с предложением ввести имя файла

Если будет обнаружено, что такого файла не существует, программа предложит пользователю повторно ввести название, либо закрыть программу (см. Рисунок 6).

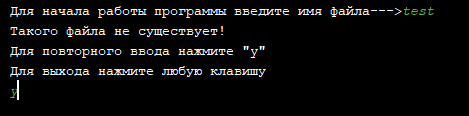


Рисунок 6. Сообщение при ошибке в названии файла

При возникновении ошибок, связанных с данными в таблице программа выведет соответствующее сообщение (см. Рисунок 7).

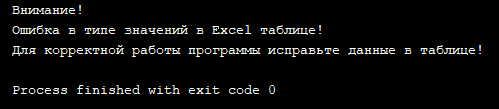


Рисунок 7. Сообщение при ошибке в таблице Excel

После успешного завершения обработки данных, пользователю будет предложено обработать еще один файл или закончить работу программы (см. Рисунок 8).

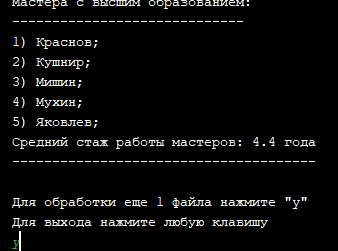


Рисунок 8. Сообщение с предложением продолжить или завершить работу

# Реализация и тестирование приложения

## Описание разработанной программы

Разработанная программа выполняет функцию анализа имеющихся в Excel данных о сотрудниках учреждения. А именно:

1. Выводит отсортированный список мастеров с высшим образованием;
2. Выводит среднее значение стажа работы этих мастеров;

## Тестирование программы

* + 1. Объект испытаний
       1. Назначение: Обработка имеющихся анкетных данных учреждения
       2. Область применения: Некоторое учреждение
       3. Наименование: Обработка
       4. Обозначение: ПО «Обработка»
    2. Цель испытаний
* Проверка программных взаимосвязей;
* Проверка правильности обработки;
* Проверка правильности вычислений;
  + 1. Средства испытания
       1. Технические средства:
          - ПК Lenovo, i7, 16Gb;
          - Монитора 23’’;
          - Мышка Logitech;
          - клавиатура Lenovo;
       2. Программные средства:

Для выполнения задач ПО «Обработка» требуется ОС Windows или Linux, а также наличие в ней Excel и Python.

* + 1. Порядок и методы испытаний (см. Таблица 5)

Таблица

Методы испытаний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Задача** | **Действия пользователя** | **Ожидаемые результаты** | **Контрольный пример** |
| **1.** | Тестирование программы с корректными данными. | Заполнить excel корректными данными (см. Таблица 6). | Программа выведет корректные данные и предложит пользователю продолжить или завершить работу (см. Рисунок 9). | файл “test1.xlsx” |
| **2.** | Тестирование программы с граничными данными. | В столбце год рождения указать граничные значения (1900 или 2003) (см. Таблица 7). | Программа выведет корректные данные и предложит пользователю продолжить или завершить работу (см. Рисунок 10). | файл “test2.xlsx” |
| **3.** | Тестирование программы с выходящими за границы данными. | В столбце год рождения указать значения, выходящие за допустимые границы (см. Таблица 8). | Программа выведет предупреждающее сообщение: «Внимание ошибка в типе значений в Excel таблице! Для корректной работы программы исправьте данные в таблице!» (см. Рисунок 11. Результат работы третьего теста). | файл “test3.xlsx” |
| **4.** | Тестирование программы с несуществующим файлом. | При указании названия файла, выбрать несуществующий файл. | Программа выведет предупреждающее сообщение: «Такого файла не существует». И предложит ввести название файла заново (см. Рисунок 12. Результат работы четвертого теста) | – |

Таблица

Входные значения для первого теста

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Фамилия** | **Год рождения** | **Должность** | **Стаж** | **Образование** |
| Кушнир | 1997 | мастер | 7 | высшее |
| Петров | 1965 | инженер | 4 | среднее |
| Иванов | 1999 | инженер | 6 | среднее |
| Федоров | 1968 | рабочий | 3 | среднее |
| Коробский | 1954 | рабочий | 2 | высшее |
| Яковлев | 1988 | мастер | 5 | высшее |
| Мухин | 1995 | мастер | 4 | высшее |
| Чижов | 1986 | мастер | 5 | среднее |
| Козлов | 1987 | инженер | 9 | среднее |
| Андреев | 1992 | рабочий | 3 | среднее |
| Краснов | 1992 | мастер | 1 | высшее |
| Серов | 1997 | инженер | 4 | среднее |
| Мишин | 1986 | мастер | 5 | высшее |
| Кухаренко | 1967 | рабочий | 6 | высшее |
| Кобец | 1993 | инженер | 7 | среднее |
| Корде | 1969 | рабочий | 8 | высшее |
| Бакунов | 1966 | инженер | 6 | среднее |
| Лапонкин | 1967 | мастер | 9 | среднее |

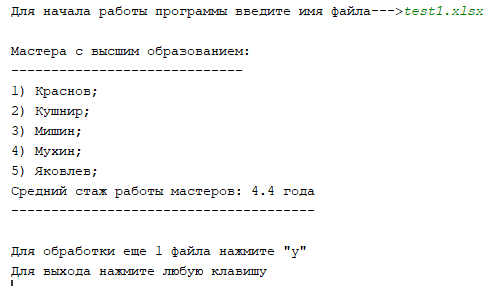


Рисунок . Результат работы первого теста

Таблица

Входные значения для второго теста

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Фамилия** | **Год рождения** | **Должность** | **Стаж** | **Образование** |
| Кушнир | 1997 | мастер | 7 | высшее |
| Петров | 1965 | инженер | 4 | среднее |
| Иванов | 1999 | инженер | 6 | среднее |
| Федоров | 1968 | рабочий | 3 | среднее |
| Коробский | 1954 | рабочий | 2 | высшее |
| Яковлев | 2003 | мастер | 5 | высшее |
| Мухин | 1995 | мастер | 4 | высшее |
| Чижов | 1986 | мастер | 5 | среднее |
| Козлов | 1987 | инженер | 9 | среднее |
| Андреев | 1992 | рабочий | 3 | среднее |
| Краснов | 1900 | мастер | 1 | высшее |
| Серов | 1997 | инженер | 4 | среднее |
| Мишин | 1986 | мастер | 5 | высшее |
| Кухаренко | 1967 | рабочий | 6 | высшее |
| Кобец | 1993 | инженер | 7 | среднее |
| Корде | 1969 | рабочий | 8 | высшее |
| Бакунов | 1966 | инженер | 6 | среднее |
| Лапонкин | 1967 | мастер | 9 | среднее |

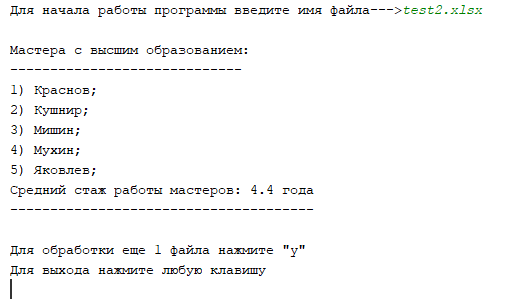


Рисунок . Результаты работы второго теста

Таблица

Входные данные для третьего теста

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Фамилия** | **Год рождения** | **Должность** | **Стаж** | **Образование** |
| Кушнир | 1812 | мастер | 7 | высшее |
| Петров | 1965 | инженер | 4 | среднее |
| Иванов | 1999 | инженер | 6 | среднее |
| Федоров | 1968 | рабочий | 3 | среднее |
| Коробский | 1954 | рабочий | 2 | высшее |
| Яковлев | 1988 | мастер | 5 | высшее |
| Мухин | 1995 | мастер | 4 | высшее |
| Чижов | 2100 | мастер | 5 | высшее |
| Козлов | 1987 | инженер | 9 | среднее |
| Андреев | 1992 | рабочий | 3 | среднее |
| Краснов | 1992 | мастер | 1 | высшее |
| Серов | 1997 | инженер | 4 | среднее |
| Мишин | 1986 | мастер | 5 | высшее |
| Кухаренко | 1967 | рабочий | 6 | высшее |
| Кобец | 1993 | инженер | 7 | среднее |
| Корде | 1969 | рабочий | 8 | высшее |
| Бакунов | 1966 | инженер | 6 | среднее |
| Лапонкин | 1967 | мастер | 9 | среднее |

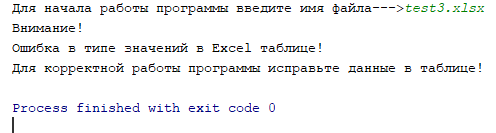


Рисунок . Результат работы третьего теста

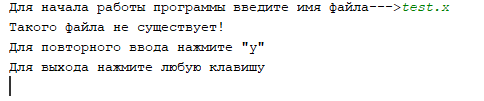


Рисунок . Результат работы четвертого теста

# Выводы

В процессе выполнения работы было проделано следующее:

1. Рассмотрена и изучена предметная область;
2. Определенны и описаны ограничения, накладываемые на входные и выходные данные программы, выделены основные функции, которые требуются для выполнения поставленной задачи;
3. Разработана спецификация и построены блок-схемы основных функций с помощью программы Microsoft Visio;
4. Разработано консольное приложение с помощью языка программирования Python в среде разработке PyCharm;
5. Разработанная программа протестирована по *ГОСТ 19.301-79;*
6. Получены теоретические и практические навыки составления спецификации и стандартизации;

# Список использованных источников

1. Лутц М. Изучаем Python 4-е издание, –Пер. с англ. –СПб.: Символ-Плюс, 2017. – 1280 с., ил.
2. Борис Бейзер Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем, –Пер. с англ. –СПб.: Питер, 2004. – 320 с.
3. Учебн. пос. - Ульяновск: УлГТУ, 2001г. - 64 с. Проектирование алгоритмов при реш. задач на ЭВМ. Технология проектирования и способ проверки несложных визуальных алгоритмов.
4. Стандартизация, сертификация и управление качеством программного обеспечения: учебно-методическое пособие по изучению курса и выполнению лабораторных работ / И.В. Шкуропадский; Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова. – Новочеркасск: ЮРГПУ (НПИ), 2017. – 48 с.
5. ГОСТ 19.301-79
6. Лекции курса «Стандартизация, спецификация и управление качеством программного обеспечения»

# Приложение. Листинг программы

"""  
8.В анкетах сотрудников некоторого учреждения имеются следующие сведения: фамилия, год рождения, должность (мастер, инженер, рабочий и т.д.),  
стаж, образование. Вывести фамилии мастеров с высшим образованием и вычислить их средний стаж. Выполнить сортировку списка.  
"""  
import xlrd  
import os  
  
def check\_excel\_values(values):  
 """  
 check excel  
 :param values: :return:  
 """  
 for value in values:  
 try:  
 if (value[0].isalpha() and  
 (str(int(value[1])).isdigit() and value[1] >= 1900 and value[1] <= 2003) and  
 value[2].isalpha() and  
 (str(int(value[3])).isdigit() and value[3] >=0 and value[3] <= 60) and  
 value[4].isalpha()):  
 return 1  
 else:  
 return  
 except (AttributeError, ValueError):  
 return  
  
  
def read\_excel(path):  
 """  
 read data from excel test.xlsx  
 :param path:  
 :return vals:  
 """  
 try:  
 # open excel  
 excel\_data = xlrd.open\_workbook(**'test.xlsx'**)  
 except FileNotFoundError:  
 print(**"File {} not found!"**.format(path))  
 return  
 # choose active sheet  
 sheet = excel\_data.sheet\_by\_index(0)  
 # get data  
 values = [sheet.row\_values(rownum) for rownum in range(sheet.nrows)]  
 if check\_excel\_values(values[1:]):  
 return values  
 else:  
 print(**"Внимание!**\n**Ошибка в типе значений в Excel таблице!**\n**"  
 "Для корректной работы программы исправьте данные в таблице!"**)  
 return  
  
  
def get\_average(values):  
 """  
 :param values:  
 :return average value:  
 """  
 sum = 0  
 for value in values:  
 # v[3] = experience  
 try:  
 if value[3] < 0:  
 print(**"Значение стажа должно быть больше или равно 0!"**)  
 return None  
 sum += value[3]  
 except (TypeError, ValueError):  
 return None  
 return sum/len(values)  
  
  
def output(values):  
 """  
 formated output  
 :param values:  
 :return None:  
 """  
 if len(values) == 0:  
 print(**'Таких сотрудников не обнаружено'**)  
 return  
  
 print(**'**\n**Мастера с высшим образованием:**\n**'  
 '-----------------------------'**)  
 list\_of\_surnames = [val[0].capitalize() for val in values]  
 # list of surnames  
 list\_of\_surnames.sort()  
 for i, surname in enumerate(list\_of\_surnames):  
 print(**'{}) {};'**.format(i + 1, surname))  
 if not get\_average(values):  
 print(**"Внимание!**\n**Ошибка в типе значений в Excel таблице!**\n**"  
 "Для корректной работы программы исправьте данные в таблице!"**)  
 else:  
 print(**"Средний стаж работы мастеров: {} года**\n**"  
 "--------------------------------------**\n**"**.format(**"%.1f"** % get\_average(values)))  
  
  
def main():  
 """  
 main function  
 :return None:  
 """  
 answer = **'Y'** while answer.lower() == **'y'**:  
 path = input(**"Для начала работы программы введите имя файла--->"**)  
 if not os.path.exists(path):  
 answer = input(**'Такого файла не существует!**\n**'  
 'Для повторного ввода нажмите "y"**\n**'  
 'Для выхода нажмите любую клавишу**\n**'**)  
 continue  
 # data = read\_excel("test.xlsx")  
 data = read\_excel(path)  
 list\_of\_employes = []  
 if not data:  
 exit(0)  
 for val in data:  
 if val[2].lower() == **'мастер'** and val[4].lower() == **'высшее'**:  
 list\_of\_employes.append(val)  
 output(list\_of\_employes)  
 answer = input(**'Для обработки еще 1 файла нажмите "y"**\n**Для выхода нажмите любую клавишу**\n**'**)  
 print(**'Работа программы успешно завершена!'**)  
  
  
if \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()