

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Петербургский государственный университет путей сообщения
Императора Александра I»
(ФГБОУ ВО ПГУПС)

Факультет «Автоматизация и интеллектуальные технологии»
Кафедра «ПИШ ИСКРА»

Курсовая работа
по дисциплине
«Введение в Python»

Выполнил обучающийся
Курс 1
Группа АСУМ-518

Засихин Н.А.

Проверил

Волков Е.А.

Санкт-Петербург
2025

Цель работы: необходимо реализовать консольное приложение, которое осуществляет считывание текстового файла csv, состоящего из строк следующего формата «YEAR; MONTH; DAY; HOUR; MINUTE; TEMPERATURE»

Ход работы:

В архиве файле хранится статистика, собранная датчиком температуры за 1 календарный год. Предполагается, что датчик собирал информацию не чаще чем 1 раз в минуту и сохранял в заданном формате каждое значение в текстовый файл с новой строки.

В какой-то момент времени датчик мог не работать, тогда данные поэтому периоду могут отсутствовать.

Необходимо вывести статистику по каждому месяцу, с учетом исходных данных:

- среднемесячная температура
- минимальная температура в текущем месяце
- максимальная температура в текущем месяце

Также необходимо вывести статистику за год:

- среднегодовая температура
- минимальная температура
- максимальная температура

Вывод статистики по месяцам и за год:

```
PS D:\Программирование\Python_PISH_ZashikhinNA\ZashikhinNA_parser_csv> python temperature_analyzer.py -f temperature_big.csv
```

```
Анализ файла: temperature_big.csv
```

```
-----  
--- Статистика по месяцам ---  
-----
```

```
Месяц: 01  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 02  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 03  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 04  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 05  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 06  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 07  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 08  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 09  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 10  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 11  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
Месяц: 12  
Среднемесечная температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

```
-----  
--- Статистика за год ---  
-----
```

```
Среднегодовая температура: 15.50°C  
Минимальная температура: 1°C  
Максимальная температура: 30°C
```

Вывод статистики за определенный месяц:

```
PS D:\Программирование\Python_PISH_ZashikhinNA\ZashikhinNA_parser_csv> python temperature_analyzer.py -f temperature_small.csv -m 7

Анализ файла: temperature_small.csv

[Ошибка формата] Стока 4: не удалось преобразовать данные в числа. Стока пропущена.
> Содержимое: 2021 | 01 | 16 | 01 | 05 | -xx

-----
--- Статистика за месяц: 07 ---
-----
Среднемесечная температура: 30.00°C
Минимальная температура: 30°C
Максимальная температура: 30°C
```

```
PS D:\Программирование\Python_PISH_ZashikhinNA\ZashikhinNA_parser_csv> python temperature_analyzer.py -h
usage: temperature_analyzer.py [-h] -f FILE [-m <номер месяца>]
```

Консольное приложение для анализа статистики температуры из CSV файла.

```
options:
-h, --help      show this help message and exit
-f, --file FILE    Входной файл CSV для обработки.
-m, --month <номер месяца>
                  Номер месяца для вывода статистики (от 1 до 12). Если не указан, выводится статистика за год.
```

```
Пример использования: python main.py -f temperature_data.csv -m 7
```

Команда help:

```
PS D:\Программирование\Python_PISH_ZashikhinNA\ZashikhinNA_parser_csv> python temperature_analyzer.py -h
usage: temperature_analyzer.py [-h] -f FILE [-m <номер месяца>]
```

Консольное приложение для анализа статистики температуры из CSV файла.

```
options:
-h, --help      show this help message and exit
-f, --file FILE    Входной файл CSV для обработки.
-m, --month <номер месяца>
                  Номер месяца для вывода статистики (от 1 до 12). Если не указан, выводится статистика за год.
```

```
Пример использования: python main.py -f temperature_data.csv -m 7
```

Вывод: было разработано консольное приложение для учета статистики температуры по месяцам

Приложение

```
import csv
import sys
import argparse
from collections import defaultdict

# Загрузка файла
def analyze_temperature_statistics(file_path, target_month=None):

    monthly_stats = defaultdict(list)

    try:
        with open(file_path, mode='r', encoding='utf-8') as file:
            reader = csv.reader(file, delimiter=';')
            print(f"\nАнализ файла: {file_path}\n")

            for line_num, row in enumerate(reader, 1):
                if len(row) != 6:
                    print(f" [Ошибка формата] Стока {line_num}: неверное количество столбцов ({len(row)}) вместо 6). Стока пропущена.")
                    print(f" > Содержимое: {' | '.join(row)}")
                    continue

                try:
                    month = int(row[1])
                    temperature = int(row[5])

                    if not (1 <= month <= 12):
                        print(f" [Ошибка данных] Стока {line_num}: неверный месяц ({month}). Стока пропущена.")
                        continue
                    if not (-99 <= temperature <= 99):
                        print(f" [Ошибка данных] Стока {line_num}: температура ({temperature}) вне диапазона [-99, 99]. Стока пропущена.")
                        continue

                    monthly_stats[month].append(temperature)

                except ValueError:
                    print(f" [Ошибка формата] Стока {line_num}: не удалось преобразовать данные в числа. Стока пропущена.")
                    print(f" > Содержимое: {' | '.join(row)}")
                    continue

            if not monthly_stats:
                print("\nВ файле не найдено валидных данных о температуре для анализа.")
                return

    # Статистика
```

```

if target_month:
    print("\n-----")
    print(f"--- Статистика за месяц: {target_month:02d} ---")
    print("-----")
    if target_month in monthly_stats:
        temps = monthly_stats[target_month]
        avg_temp = sum(temps) / len(temps)
        min_temp = min(temps)
        max_temp = max(temps)

        print(f" Среднемесячная температура: {avg_temp:.2f}°C")
        print(f" Минимальная температура: {min_temp}°C")
        print(f" Максимальная температура: {max_temp}°C")
    else:
        print(" Данные за указанный месяц отсутствуют.")
else:
    # Вывод полной статистики (по всем месяцам и за год)
    print("\n-----")
    print("--- Статистика по месяцам ---")
    print("-----")
    all_temperatures = []

for month in range(1, 13):
    if month in monthly_stats:
        temps = monthly_stats[month]
        all_temperatures.extend(temps)

        avg_temp = sum(temps) / len(temps)
        min_temp = min(temps)
        max_temp = max(temps)

        print(f"\nМесяц: {month:02d}")
        print(f" Среднемесячная температура: {avg_temp:.2f}°C")
        print(f" Минимальная температура: {min_temp}°C")
        print(f" Максимальная температура: {max_temp}°C")
    else:
        print(f"\nМесяц: {month:02d}")
        print(" Данные отсутствуют.")

if all_temperatures:
    print("\n-----")
    print("--- Статистика за год ---")
    print("-----")
    yearly_avg = sum(all_temperatures) / len(all_temperatures)
    yearly_min = min(all_temperatures)
    yearly_max = max(all_temperatures)

    print(f"\n Среднегодовая температура: {yearly_avg:.2f}°C")
    print(f" Минимальная температура: {yearly_min}°C")
    print(f" Максимальная температура: {yearly_max}°C")

```

```
except FileNotFoundError:
    print(f"ERROR: Файл не найден по пути: {file_path}")
    sys.exit(1)
except Exception as e:
    print(f"ERROR: Произошла непредвиденная ошибка: {e}")
    sys.exit(1)

# Основная функция
def main():
    """
    Основная функция для разбора аргументов командной строки и запуска анализа.
    """
    parser = argparse.ArgumentParser(
        description="Консольное приложение для анализа статистики температуры из CSV файла.",
        epilog="Пример использования: python main.py -f temperature_data.csv -m 7"
    )

    parser.add_argument(
        '-f', '--file',
        type=str,
        required=True,
        help="Входной файл CSV для обработки."
    )

    parser.add_argument(
        '-m', '--month',
        type=int,
        choices=range(1, 13),
        metavar='<номер месяца>',
        help="Номер месяца для вывода статистики (от 1 до 12). Если не указан, выводится статистика за год."
    )

    args = parser.parse_args()
    analyze_temperature_statistics(args.file, args.month)

if __name__ == "__main__":
    main()
```