**1. Состав участников.**

1. Бухтияров Виталий;
2. Вишняк Илья;
3. Легкая Мария;
4. Павленко Александр.

**2. Постановка задачи.**

Необходимо написать программу «Вечный календарь» для определения дня недели по дате. Дополнительно будут напечатаны название сезона и номера века, записанного римскими цифрами.

**3. Словесно-формульное описание алгоритма решения.**

Для вычисления дня недели используется формула:

где  — число месяца, номер месяца, номер столетия, номер года в столетии, соответственно. [·] — здесь означает, что берется целая часть числа.

Проект содержит следующие файлы:

* century.py – содержит функцию century, которая принимает номер года, выводит номер столетия.
* monthToNum.py – содержит Словарь, где индекс – строка, название месяца, а значение – его номер. Важно: нумерация месяцев начинается с марта.
* numOfYear.py – содержит функцию numOfYear, которая принимает номер года, выводит номер года в столетии.
* rome\_century – содержит функцию age, которая функция age(ndata\_age) принимает год и выводит век в римских цифрах.
* season.py – содержит функцию season, которая принимает номер месяца, выводит строку, название сезона в году.
* week.py – содрежит список в котором индексы сопоставляются строке, дням недели. Важно: нумерация дней недели начинается с воскресенья.
* test.py – содержит список тестов
* general.py – главный файл, в котором реализован проект.

В главном файле general.py с помощью вышеуказанных функций считаются переменные: — число месяца, номер месяца, номер столетия, номер года в столетии, соответственно. По указанной в условии формуле считается день недели и сохраняется в переменной day\_of\_week. Для извлечения целой части достаточно использовать целочисленное деление.

После обработки вышеуказанных переменных, выводится результат в форме, указанной в описании задачи.

Однако, в ходе работы было замечено, что формула, описанная в условии задачи, работает некорректно. Для некоторых входных данных программа выводит неверный ответ. Например, при вводе "23 сентября 2019" программа выводит не "понедельник", верный ответ, а "суббота", неверный ответ.

Мы создали файл [secondary.py](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fsecondary.py), в котором реализован алгоритм правильного подсчета дня недели с помощью вышеуказанных функций и новой формулы в функции weekday, которая описана в файле [weekday.py](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fweekday.py):

где  — число месяца, номер месяца (нумерация с января), номер года в столетии, соответственно. [·] — здесь означает, что берется целая часть числа. Результат — номер дня недели (нумерация с понедельника).

В конце программы файла secondary.py реализован алгоритм сохранения результата в текстовый файл Vima\_secondary.tex.

**4. Пошаговая инструкция по работе с программой для пользователя.**

Пользователь вводит дату в формате *дд˽мм˽гггг*, где *мм* – полное название месяца на русском языке, а символ «˽» – пробел.

Пример:

|  |
| --- |
| Введите дату >>> 30 октября 2019 |

Программа выводит день недели, название сезона и номер века, записанный римскими цифрами, в виде:

|  |
| --- |
| День недели: *\*день недели\**  Век: *\*номер века, записанный римскими цифрами\**  Сезон: *\*название сезона\** |

**5. Обоснование каждого теста, выбранного в п. 1 (в виде таблицы).**

|  |  |  |
| --- | --- | --- |
| Тест 1 | 29 февраля 2020 | День високосного года |
| Тест 2 | 1 января 2000 | Граница после перехода столетия |
| Тест 3 | 31 декабря 1999 | Граница до перехода столетия |
| Тест 4 | 26 апреля 1143 | Случайная дата |
| Тест 5 | 13 февраля 2020 | Сегодняшняя дата |
| Тест 6 | 25 июля 749 | Год первого тысячелетия |

**6. Результат работы программы для тестов из п. 1.**

Введите дату >>> 29 февраля 2020

----------------

День недели: суббота

Век: XXI

Сезон: зима

Введите дату >>> 1 января 2000

----------------

День недели: суббота

Век: XXI

Сезон: зима

Введите дату >>> 31 декабря 1999

----------------

День недели: пятница

Век: XX

Сезон: зима

Введите дату >>> 26 апреля 1143

----------------

День недели: вторник

Век: XII

Сезон: весна

Введите дату >>> 13 февраля 2020

----------------

День недели: четверг

Век: XXI

Сезон: зима

Введите дату >>> 25 июля 749

----------------

День недели: суббота

Век: VIII

Сезон: лето