Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕНЫННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий

И управления

Кафедра Интеллектуальных информационных технологий

ОТЧЕТ

Лабораторная работа №1

Выполнил:

Кимстач Д.Б.

Проверил:

Никифоров С.А.

Минск 2022

Тема:

Изучение основ объектно-ориентированного программирования на языке С++.

Задание:

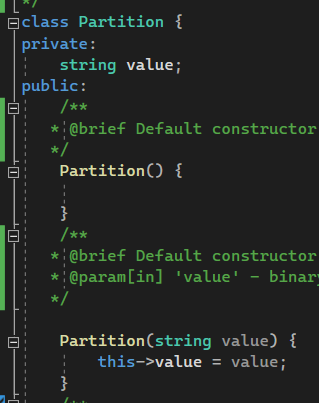
Реализовать на языке С++ один из нижеперечисленных вариантов и написать и сгенерировать документацию при помощи doxygen.

Вариант 1.8.2

Описать классы, реализующие машину Тьюринга.

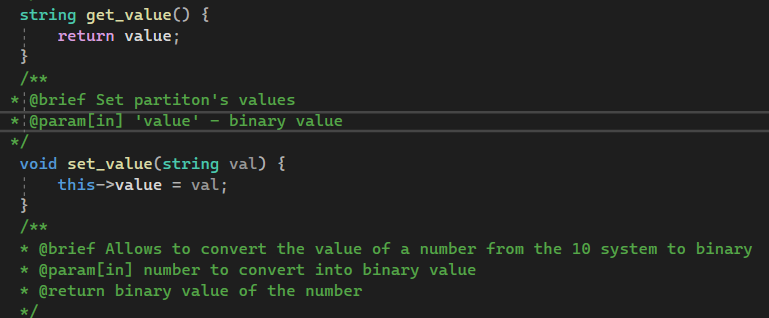
За основу взят материал на YouTube: https://youtu.be/VNDVAXw-Ems

Файл Header.h:

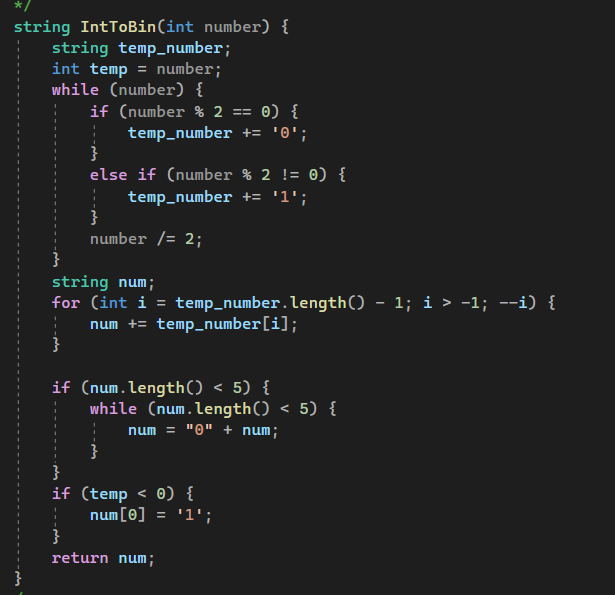


Класс Partition(ячейка в ленте) создан для работы с ячейками в ленте Машины Тьюринга.

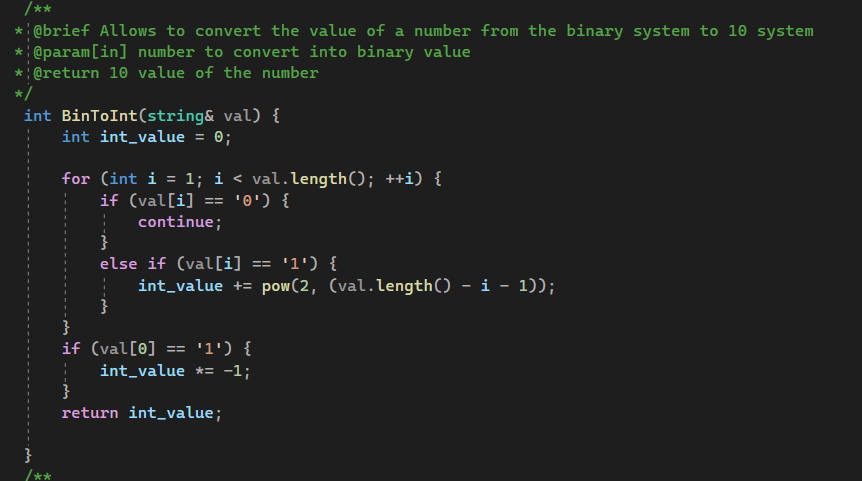
Класс имеет для конструктора: один пустой, другой, присваивающий переменной “value” двоичное представление числа из файла.



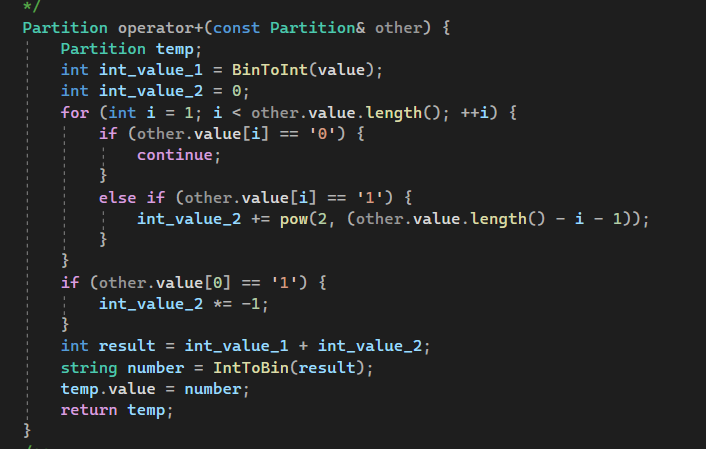
В классе реализованы простейшие геттеры/сеттеры для получения/изменения поля “value” класса “Partition”.



Данный метод создан для перевода числа из десятичной системы в двоичную, учитывая его знак. Само число в двоичной системе имеет разрядность 4, а 5, последний разряд, отведен под знак (если “0” - число положительно, если же “1” - число отрицательное).



Данный метод позволяет перевести число в двоичной системе в десятичную, так же учитывая то, что число в двоичном виде может быть отрицательным.

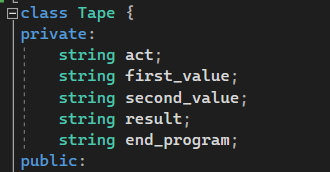


Далее следует перегрузка оператора “+”. Она позволяет беспрепятственно проводить заданную операцию с экземплярами класса “Partition”. Её результат - новый экземпляр класса “Partition”, у которого поле “value” является результатом операции “+” двух других значений поля “value” других экземпляров класса “Partition”. Внутри Метода также идет конвертация в десятичную систему второго слагаемого(его поля “value”).

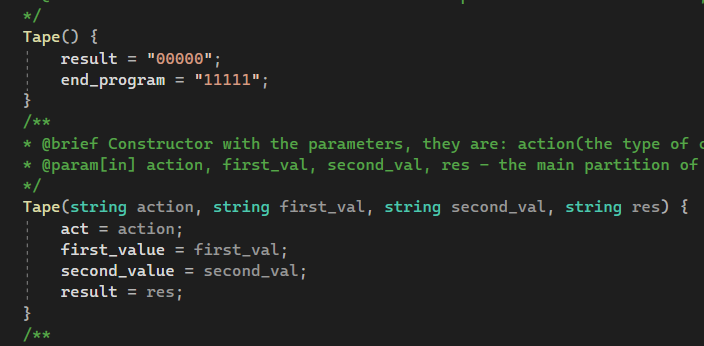
Далее идет перегрузка других арифметических и логических операторов, которые отличаются от перегрузки оператора “+”, по факту, только самим знаком.

В программе реализованы: Сложение, Умножение, Вычитание, Деление, побитовый сдвиг влево, побитовый сдвиг вправо, логическое исключающее “и” (XOR), логическое или (|), логическое “И” (&).

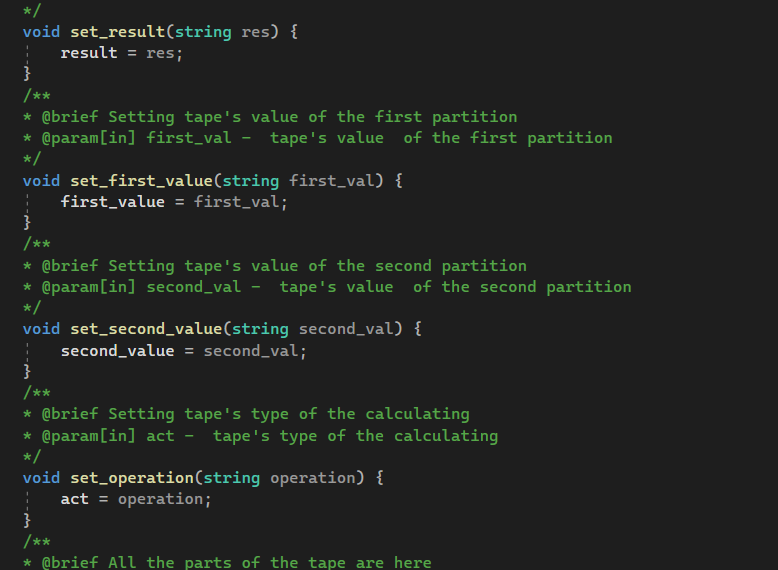
Далее в программе реализован класс “Tape” (лента). Он, по сути, содержит в себе все значения ленты. В ленте идет тип операции в двоичном виде, далее идет первая ячейка для вычисления, далее вторая ячейка для вычисления, далее идет ячейка результата (изначально пустая\*(равна 00000). Далее идет конец операции (11111).



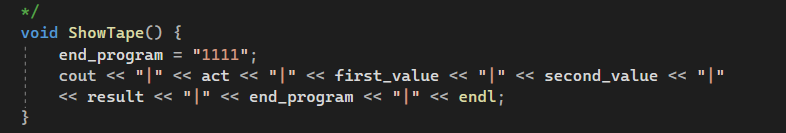
Далее идет реализация конструкторов: в первом из них значения результата и конца программы задаются по умолчанию. Другой конструктор для задания всех “динамических” (изменяются, используются по ходу программы) ячеек.



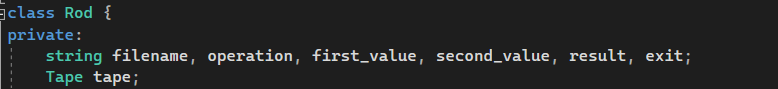
Далее идут классические “Setter(ы)” для изменения значений полей класса “Tape”.



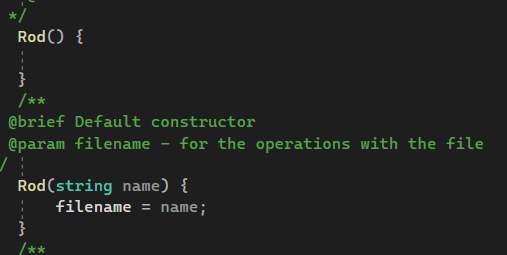
Далее реализован метод, отображающий всю ячейку в консоли



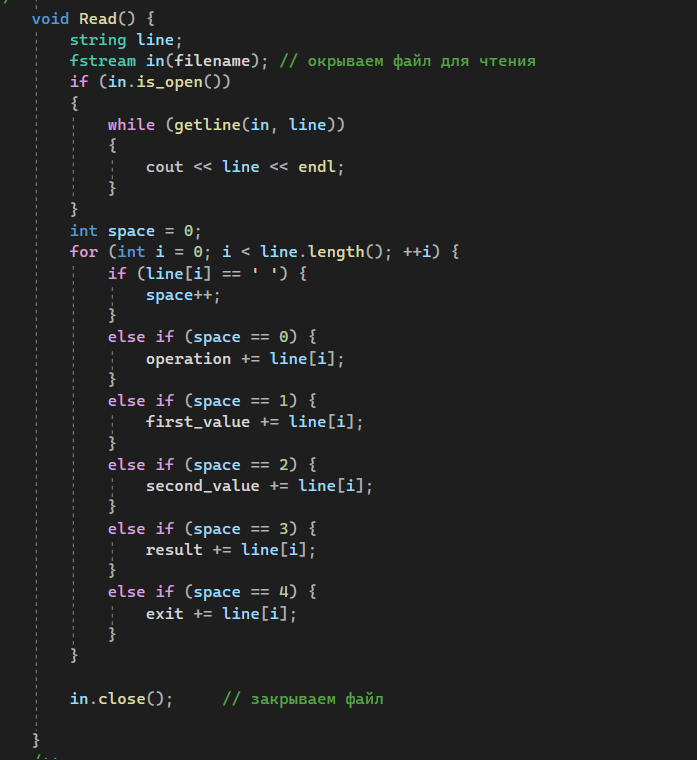
Далее идет реализация класса “Rod”(каретка). Она ответственная за чтение файла, запись в файл.



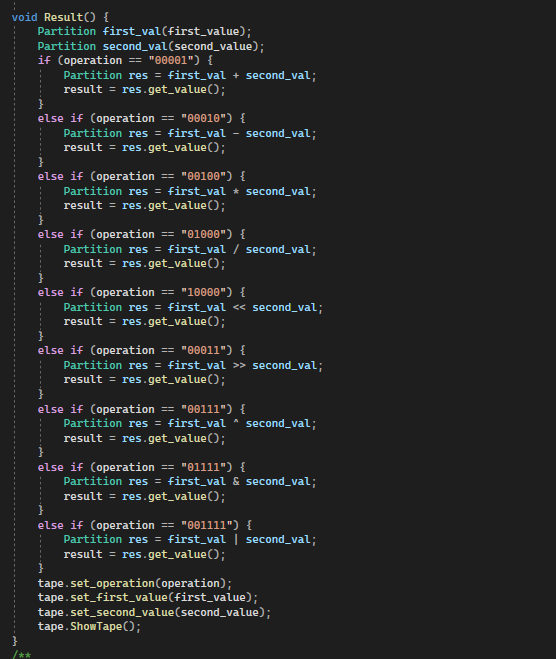
Здесь показана реализация двух конструкторов:



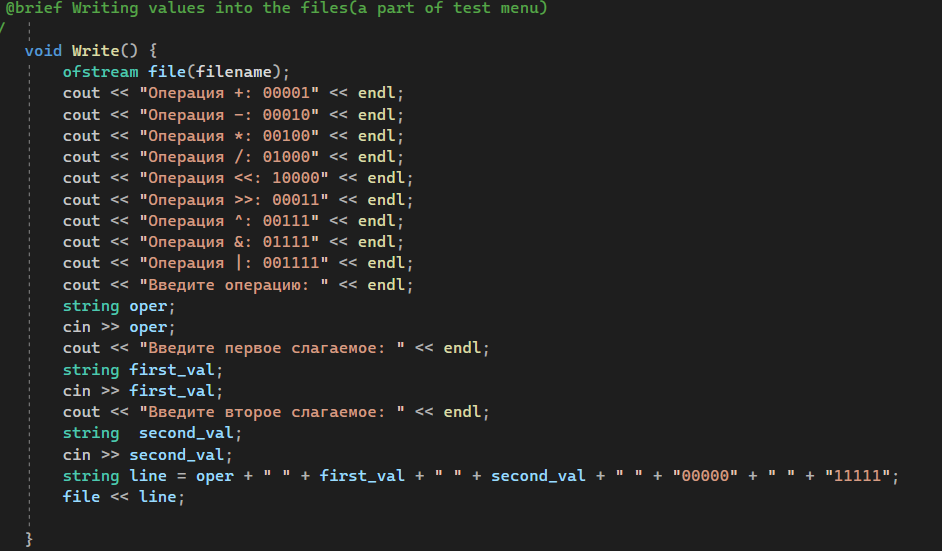
Далее показан метод для чтения из файла. Метод поэлементно считывает значения из файла и “закрепляет” их в коретке.



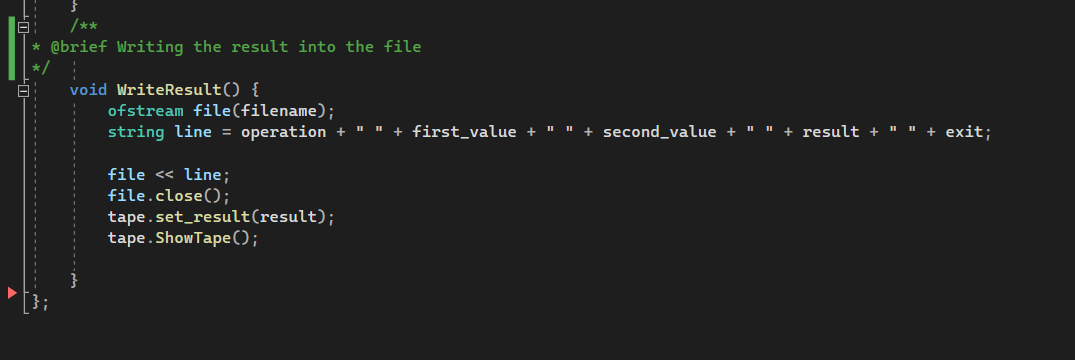
Метод подсчета результата(в зависимости от заданной операции в файле), используется перегрузка арифметических и логических операция класса “Partition”.



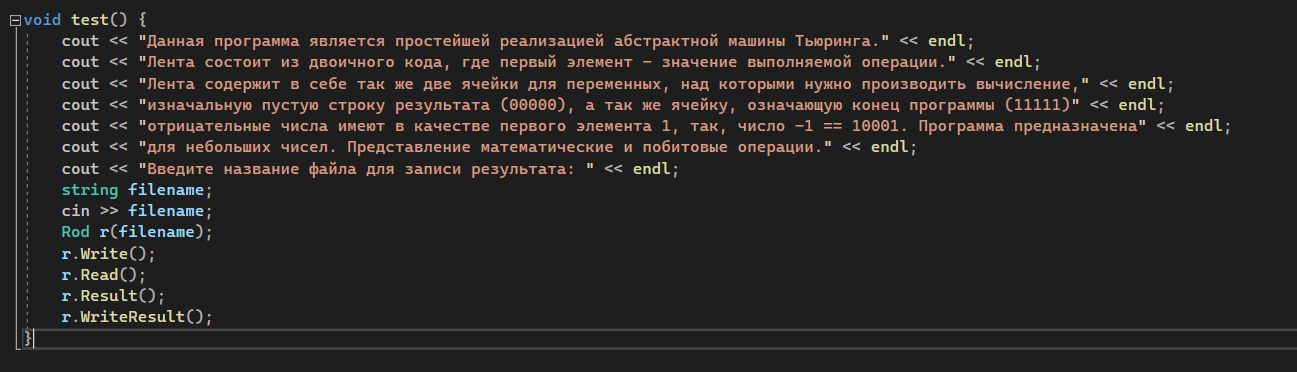
Метод, предназначенный для ручной записи из программы(нужен для тестового менбю).



Метод, перезаписывающий значения результата(используется при готовом файле со значениями)

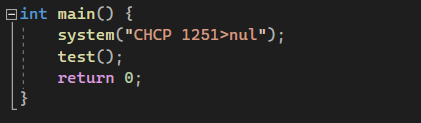


Файл ConsoleApplication1

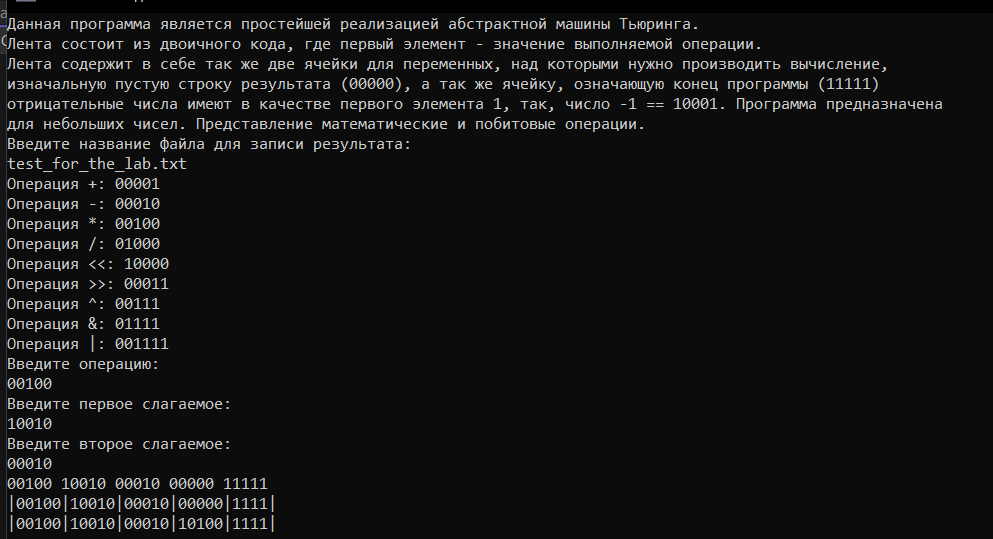


Реализована тестовая функция для работы с файлом “Header1.h”, так же вывод основных особенностей программы.

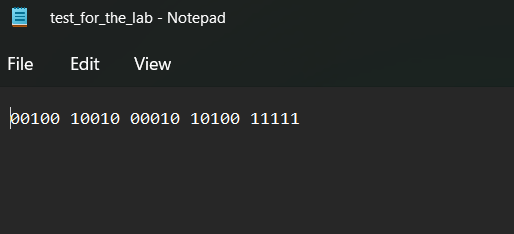
Функция “main” (вызов функции “test”):



Пример работы программы:



Пользователь вводит названия файла для записи туда, выбирает операцию, вводит значения в двоичном виде( в примере указаны значения -2 и 2 и операция умножения). Файл показывает изначальное состояние, где результат ещё не посчитан и конечное состояние, где результат вписан. Конечное состояние файла:



Вывод:

В ходе лабораторной работы была изучена структура машины Тьюринга, ее реализация на языке С++ при помощи Объектно-ориентированного подхода.