Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Южно-Уральский государственный университет»

Факультет «Компьютерных технологий, управления и радиоэлектроники»

Кафедра «Электронные вычислительные машины»

Графический интерфейс для машины Тьюринга

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ

по дисциплине: Программная инженерия

ЮУрГУ -230100.2015.886 ПЗ КП

Нормоконтролёр Руководитель

С.В. Сяськов А.А. Кирсанова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г. «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2015 г.

Авторы проекта

студенты группы

КТУР-318

И.Е. Ботов

«\_\_\_\_\_\_» \_\_\_ \_\_\_\_\_\_\_ 2015 г.

М.А. Черкасов

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 г.

Проект защищен с оценкой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 г

Челябинск, 2015

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Южно-Уральский государственный университет»

Факультет «Компьютерных технологий, управления и радиоэлектроники»

Кафедра «Электронные вычислительные машины»

Специальность «Информатика и вычислительная техника»

ЗАДАНИЕ

на курсовой проект студентов

Ботова Ильи Евгеньевича

Черкасова Максима Александровича

Группа: КТУР-318

1 Дисциплина: программная инженерия (методологии проектирования программного обеспечения, паттерны)

2 Тема проекта

Графический интерфейс для машины Тьюринга

3 Срок сдачи законченного проекта \_\_\_\_\_\_\_\_\_\_\_\_\_2015г.

4 Перечень вопросов, подлежащих разработке

Целью курсового проекта является написание графического интерфейса для абстрактного исполнителя – машины Тьюринга.

Реализовать следующие функции:

* написание, выполнение и отладка программ для машины Тьюринга с помощью графического интерфейса пользователя;
* возможность выполнения программы по шагам, с точками остановки, с задаваемыми паузами между шагами;
* поддержка импорта и экспорта программы в файл, а также временное сохранение текущей ленты
* файл справки
* добавление условия решаемой задачи и комментариев к ней.

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.А. Кирсанова/

Студенты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/И.Е. Ботов/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Черкасов/

АННОТАЦИЯ

Ботов И.Е., Черкасов М.А. Разработка графического

интерфейса машины Тьюринга –

Челябинск: ФГБОУ ВПО ЮУРГУ (НИУ), КТУР-318, 19 с., 3 ил.,

библиогр. список – 1 наим., 1 прил.

В рамках курсового проекта проведен анализ проблем разработчика и составлен список вопросов, подлежащих разработке.

Была спроектирована и реализована модель машины Тьюринга включающая графический интерфейс пользователя.

Пояснительная записка к курсовому проекту оформлена в текстовом редакторе MS Word 2013.

Диаграммы выполнены при помощи редактора диаграмм StarUml.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ1

1 ОПИСАНИЕ ПРОЦЕССА РЕШЕНИЯ ЗАДАЧИ1

ЗАКЛЮЧЕНИЕ2

БИБЛИОГРАФИЧЕСКИЙ СПИСОК2

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. СПЕЦИФИКАЦИЯ ПРОЕКТНОГО РЕШЕНИЯ3

ВВЕДЕНИЕ

Необходимо разработать и написать интерфейс для машины Тьюринга и реализовать заданный перечень функций.

1 ОПИСАНИЕ ПРОЦЕССА РЕШЕНИЯ ЗАДАЧИ

Результаты проектирования и реализации представлены в приложении А.

В ходе проектирования был выявлен необходимый функционал графического интерфейса. Он представлен на рисунке А.2.

Полученная модель интерфейса была смоделирована с помощью модулей, представленных на рисунке А.1.

Полное описание объектно-ориентированной модели представлено на рисунке А.3.

В ходе разработки модели были задействованы паттерны, перечисленные таблице А.1

Получившиеся классы:

1. Machine.
2. Table.
3. Tape.
4. TapeMemento.
5. rule.

Общие сведения о методах, атрибутах класса Machine представлены в таблицах А.2 – А.22.

Общие сведения о методах, атрибутах класса Table представлены в таблицах А.23 – А.32.

Общие сведения о методах, атрибутах класса Tape представлены в таблицах А.33-А.39.

Общие сведения о методах, атрибутах класса TapeMemento представлены в таблицах А.40-А.44.

Общие сведения о методах, атрибутах класса rule представлены в таблицах А.45 и А.46.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была спроектирована и написана программа интерфейса для машины Тьюринга с заданным перечнем функций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм и др. – СПб: Питер, 2001. – 368 с.

**СПЕЦИФИКАЦИЯ ПРОЕКТНОГО РЕШЕНИЯ**

.

|  |  |
| --- | --- |
| Авторы проекта  Ботов И.Е.  Черкасов М.А. |  |
| Группа  КТУР-318 |  |
| Дата создания документа 19.05.2015 |  |

Версия: 1.0

**История изменений**

| Дата | Автор | Версия документа | Комментарии (Что изменено, добавлено, удалено и т.д.) |
| --- | --- | --- | --- |
| 19.05.2015 | Ботов И.Е. | 1.0 | Описание проектного решения, спецификация программного решения |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Рецензии**

| ФИО рецензента | Версия документа | Должность рецензента | Дата |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Описательные свойства документа**

| Имя | Значение |
| --- | --- |
| Название документа | Спецификация проекта |
| Автор | Ботов И.Е. |
| Дата создания | 19.05.2015 |
| Дата последнего обновления | 25.05.2015 |
| Дисциплина | Программная инженерия |
| **ФИО и подпись лиц утвердивших документ** | А.А Кирсанова |
| **Дата утверждения документа** |  |

Оглавление

[**1 ОПИСАНИЕ ПРОЕКТНОГО РЕШЕНИЯ** 6](#_Toc420401311)

[**1.1 Архитектура программной системы** 6](#_Toc420401312)

[**1.2 Физический дизайн** 8](#_Toc420401313)

[**1.3 Используемые типовые решения (паттерны)** 9](#_Toc420401314)

[**2 СПЕЦИФИКАЦИЯ ПРОГРАММНОГО РЕШЕНИЯ** 10](#_Toc420401315)

[**2.1 Подсистема Машина Тьюринга** 10](#_Toc420401316)

[**2.1.1 Класс Machine** 10](#_Toc420401317)

[**2.1.2 Класс Table** 14](#_Toc420401318)

[**2.1.3 Класс Tape** 17](#_Toc420401319)

[**2.1.4 Класс TapeMemento** 18](#_Toc420401320)

[**2.1.5 Класс rule** 19](#_Toc420401321)

**1 ОПИСАНИЕ ПРОЕКТНОГО РЕШЕНИЯ**

Данное проектное решение эмулирует работу машины Тьюринга и предоставляет графический интерфейс для написания, выполнения и отладки программ для нее.

**1.1 Архитектура программной системы**

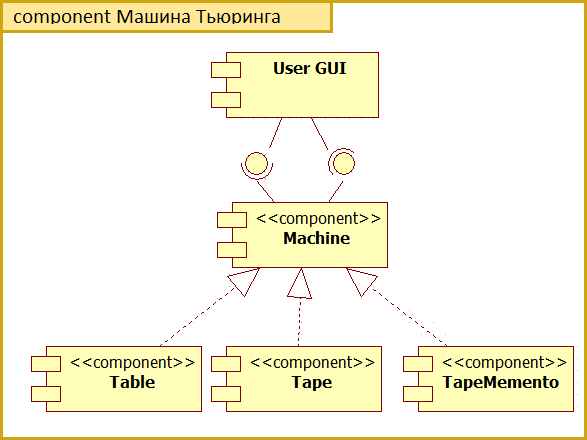
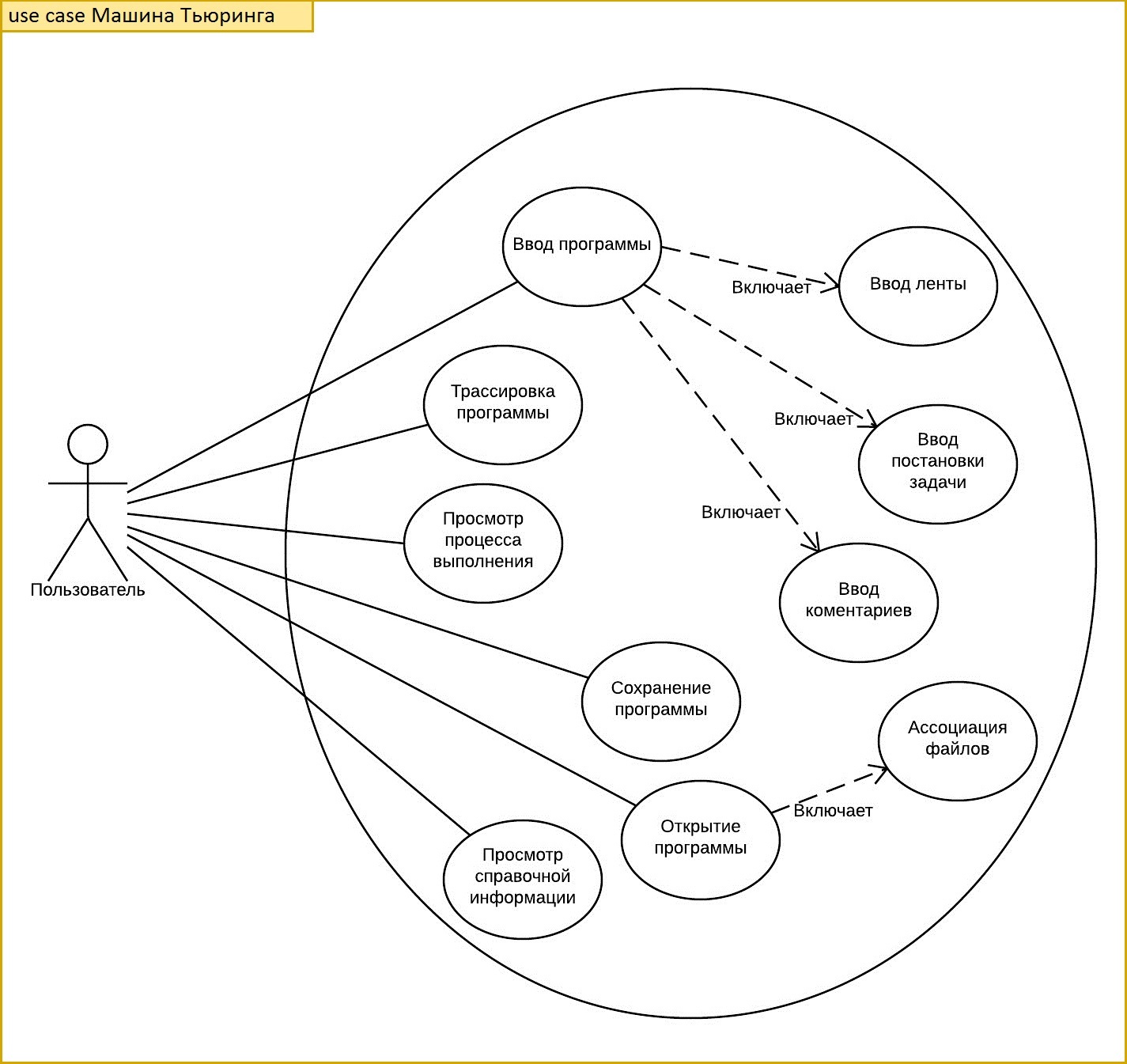


Рисунок А.1 – Диаграмма компонентов

Рисунок А.2 – Диаграмма использования

**1.2 Физический дизайн**

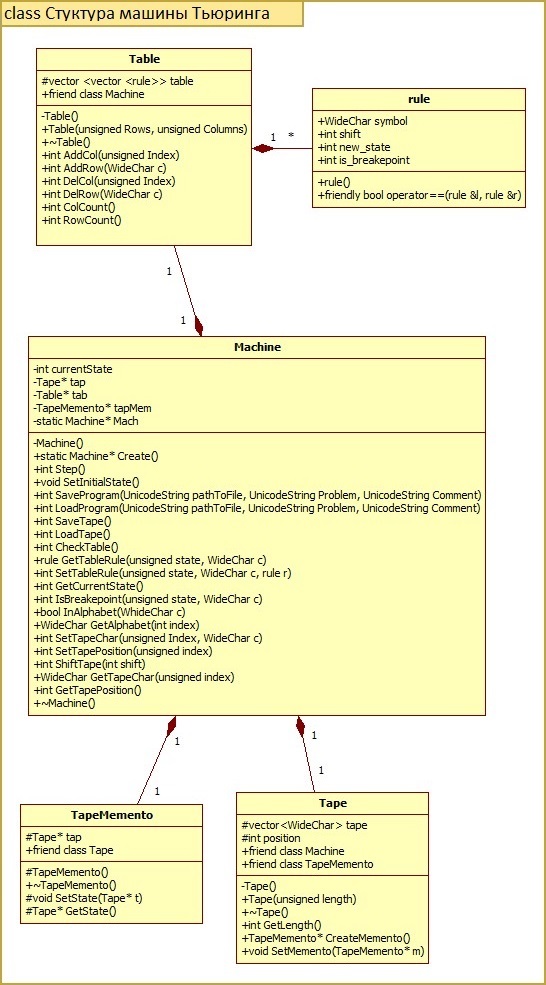


Рисунок А.3 – Диаграмма классов

**1.3 Используемые типовые решения (паттерны)**

Таблица А.1 – Используемые паттерны

|  |  |  |
| --- | --- | --- |
| Имя паттерна | Обоснование использования в проектном решении | Классы проектного решения, которые задействованы в паттерне |
| Хранитель | Необходимо временно сохранять состояние ленты для последующей после изменений загрузки. | Machine, TapeMemento, Tape |
| Одиночка | Машина должна существовать в единственном экземпляре. | Machine |

**2 СПЕЦИФИКАЦИЯ ПРОГРАММНОГО РЕШЕНИЯ**

**2.1 Подсистема Машина Тьюринга**

**2.1.1 Класс Machine**

Таблица А.2 – Общие сведения о классе Machine

|  |  |
| --- | --- |
| Назначение класса в проектном решении | Эмуляция работы машины Тьюринга, предоставление интерфейса работы с лентой и таблицей. |
| Соответствие логическому дизайну | Machine |
| Размещение исходного кода | <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/Machine.cpp>  <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/Machine.h> |

Таблица А.3 – Атрибуты класса Machine

|  |  |  |
| --- | --- | --- |
| Имя атрибута в программном коде | Тип атрибута | Описание |
| currentState | Int | Хранение текущего состояния машины |
| tapMem | TapeMemento\* | Хранение ссылки на класс сохраняющий состояние ленты |
| tap | Tape\* | Ссылка на класс реализующий ленту |
| tab | Table\* | Ссылка на класс реализующий таблицу переходов |

Таблица А.4 – Метод Create

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | Create | | | |
| Назначение | Создание и инициализация машины Тьюринга | | | |
| Возвращаемое значение | Machine\* | Ссылка на созданную\полученную машину | | |
| Список параметров | Имя | | Тип | Описание |
| tapLen | | Unsigned | Начальная длина ленты |
| tabRows | | Unsigned | Начальное количество строк таблицы |
| tabCol | | Unsigned | Начальное количество столбцов таблицы |

Таблица А.5 – Метод Step

|  |  |  |
| --- | --- | --- |
| Имя метода | Step | |
| Назначение | Выполнение одного шага | |
| Возвращаемое значение | Int | Результат после совершения шага |

Таблица А.6 – Метод SetInitialState

|  |  |  |
| --- | --- | --- |
| Имя метода | SetInitialState | |
| Назначение | Установка текущего состояния машины в 1 | |
| Возвращаемое значение | void |  |

Таблица А.7 – Метод SaveProgram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | SaveProgram | | | |
| Назначение | Сохранение ленты, таблицы, постановки задачи, комментариев и их параметров в файл | | | |
| Возвращаемое значение | Int | Результат записи в файл | | |
| Список параметров | Имя | | Тип | Описание |
| pathToFile | | UnicodeString | Полный\относительный путь до файла |

Таблица А.8 – Метод LoadProgram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | LoadProgram | | | |
| Назначение | Загрузка ленты, таблицы, постановки задачи, комментариев и их параметров в файл | | | |
| Возвращаемое значение | Int | Результат загрузки из файла | | |
| Список параметров | Имя | | Тип | Описание |
| pathToFile | | UnicodeString | Полный\относительный путь до файла |

Таблица А.9 – Метод SaveTape

|  |  |  |
| --- | --- | --- |
| Имя метода | SaveTape | |
| Назначение | Временное сохранение состояния ленты | |
| Возвращаемое значение | void |  |

Таблица А.10 – Метод LoadTape

|  |  |  |
| --- | --- | --- |
| Имя метода | LoadTape | |
| Назначение | Загрузка состояния ленты из временного хранилища | |
| Возвращаемое значение | Int | Результат загрузки |
| Имя метода | GetTableRule | |
| Назначение | Проверка таблицы на корректность (наличие символов правил в алфавите) | |
| Возвращаемое значение | Int | Результат проверки |

Таблица А.11– Метод GetTableRule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | GetTableRule | | | |
| Назначение | Получение правила из таблицы | | | |
| Возвращаемое значение | rule | Найденное правило | | |
| Список параметров | Имя | | Тип | Описание |
| state | | Unsigned | Состояние в таблице (столбец) |
| c | | WideChar | Символ в таблице (строка) |

Таблица А.12 – Метод SetTableRule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | SetTableRule | | | |
| Назначение | Установка правила в таблицу | | | |
| Возвращаемое значение | Int | Результат установки | | |
| Список параметров | Имя | | Тип | Описание |
| state | | Unsigned | Состояние в таблице (столбец) |
| c | | WideChar | Символ в таблице (строка) |

Таблица А.13 – Метод GetCurrentState

|  |  |  |
| --- | --- | --- |
| Имя метода | GetCurrentState | |
| Назначение | Получение текущего состояния машины | |
| Возвращаемое значение | Int | Текущее состояние |

Таблица А.14 – Метод IsBreakepoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | IsBreakepoint | | | |
| Назначение | Проверка правила из таблицы на наличие точки останова | | | |
| Возвращаемое значение | Int | Результат проверки | | |
| Список параметров | Имя | | Тип | Описание |
| state | | Unsigned | Состояние в таблице (столбец) |
| c | | WideChar | Символ в таблице (строка) |

Таблица А.15 – Метод InAlphabet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | InAlphabet | | | |
| Назначение | Проверка наличия символа в алфавите машины | | | |
| Возвращаемое значение | Bool | Результат проверки | | |
| Список параметров | Имя | | Тип | Описание |
| с | | WideChar | Проверяемый символ |

Таблица А.16 – Метод GetAlphabet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | GetAlphabet | | | |
| Назначение | Получение символа алфавита по индексу строки | | | |
| Возвращаемое значение | WideChar | Символ строки | | |
| Список параметров | Имя | | Тип | Описание |
| index | | Int | Номер строки |

Таблица А.17 – Метод SetTapeChar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | SetTapeChar | | | |
| Назначение | Установка символа в ленту | | | |
| Возвращаемое значение | Int | Результат установки | | |
| Список параметров | Имя | | Тип | Описание |
| index | | Unsigned | Положение в ленте |
| c | | WideChar | Устанавливаемый символ |

Таблица А.18 – Метод SetTapePosition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | SetTapePosition | | | |
| Назначение | Установка текущего положения в ленте | | | |
| Возвращаемое значение | Int | Результат установки | | |
| Список параметров | Имя | | Тип | Описание |
| index | | Unsigned | Положение в ленте |

Таблица А.19 – Метод ShiftTape

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | ShiftTape | | | |
| Назначение | Сдвиг текущего положения в ленте | | | |
| Возвращаемое значение | Int | Результат сдвига | | |
| Список параметров | Имя | | Тип | Описание |
| shift | | int | Направление сдвига |

Таблица А.20 – Метод GetTapeChar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | GetTapeChar | | | |
| Назначение | Получение символа из ленты | | | |
| Возвращаемое значение | WideChar | Полученный символ | | |
| Список параметров | Имя | | Тип | Описание |
| index | | Unsigned | Положение в ленте |

Таблица А.21 – Метод GetTapePosition

|  |  |  |
| --- | --- | --- |
| Имя метода | GetTapePosition | |
| Назначение | Получение текущего положения ленты | |
| Возвращаемое значение | Int | Положение в ленте |

Таблица А.22 – Метод Machine

|  |  |
| --- | --- |
| Имя метода | ~Machine |
| Назначение | Деструктор машины |

**2.1.2 Класс Table**

Таблица А.23 – Общие сведения о классе Table

|  |  |
| --- | --- |
| Назначение класса в проектном решении | Реализация таблицы переходов |
| Соответствие логическому дизайну | Table |
| Размещение исходного кода | <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/Table.cpp>  <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/Table.h> |

Таблица А.24 – Атрибуты класса Table

|  |  |  |
| --- | --- | --- |
| Имя атрибута в программном коде | Тип атрибута | Описание |
| Table | vector<vector<rule> > | Хранение данных таблицы переходов |

Таблица А.25 – Метод Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | Table | | | |
| Назначение | Создание и инициализация таблицы переходов | | | |
| Возвращаемое значение | - |  | | |
| Список параметров | Имя | | Тип | Описание |
| Rows | | Unsigned | Начальное количество строк таблицы |
| Columns | | Unsigned | Начальное количество столбцов таблицы |

Таблица А.26 – Метод AddCol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | AddCol | | | |
| Назначение | Добавление нового столбца | | | |
| Возвращаемое значение | Int | Результат добавления столбца | | |
| Список параметров | Имя | | Тип | Описание |
| Index | | Unsigned | Индекс до которого добавится столбец |

Таблица А.27 – Метод AddRow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | AddRow | | | |
| Назначение | Добавление новой строки | | | |
| Возвращаемое значение | Int | Результат добавления строки | | |
| Список параметров | Имя | | Тип | Описание |
| с | | WideChar | Символ добавляемой строки |

Таблица А.28 – Метод DelCol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | DelCol | | | |
| Назначение | Удаление столбца | | | |
| Возвращаемое значение | Int | Результат удаления столбца | | |
| Список параметров | Имя | | Тип | Описание |
| Index | | Unsigned | Индекс удаляемого столбца |

Таблица А.29 – Метод DelRow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | DelRow | | | |
| Назначение | Удаление строки | | | |
| Возвращаемое значение | Int | Результат удаления строки | | |
| Список параметров | Имя | | Тип | Описание |
| с | | WideCharс | Символ удаляемой строки |

Таблица А.30 – Метод DelRow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | DelRow | | | |
| Назначение | Удаление строки | | | |
| Возвращаемое значение | Int | Результат удаления строки | | |
| Список параметров | Имя | | Тип | Описание |
| с | | WideCharс | Символ удаляемой строки |

Таблица А.31 – Метод ColCount

|  |  |  |
| --- | --- | --- |
| Имя метода | ColCount | |
| Назначение | Получение количества столбцов в таблице | |
| Возвращаемое значение | Int | Количество столбцов |

Таблица А.32 – Метод RowCount

|  |  |  |
| --- | --- | --- |
| Имя метода | RowCount | |
| Назначение | Получение количества строк в таблице | |
| Возвращаемое значение | Int | Количество строк |

**2.1.3 Класс Tape**

Таблица А.33 – Общие сведения о классе Tape

|  |  |
| --- | --- |
| Назначение класса в проектном решении | Реализация ленты машины |
| Соответствие логическому дизайну | Tape |
| Размещение исходного кода | <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/Tape.cpp>  <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/Tape.h> |

Таблица А.34 – Атрибуты класса Tape

|  |  |  |
| --- | --- | --- |
| Имя атрибута в программном коде | Тип атрибута | Описание |
| tape | vector<WideChar> | Хранение ленты |
| position | int | Хранение текущего положения в ленте |

Таблица А.35 – Метод Tape

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | Tape | | | |
| Назначение | Создание и инициализация ленты | | | |
| Возвращаемое значение | - |  | | |
| Список параметров | Имя | | Тип | Описание |
| length | | Unsigned | Начальная длина ленты |

Таблица А.36 – Метод GetLenght

|  |  |  |
| --- | --- | --- |
| Имя метода | GetLenght | |
| Назначение | Получение длины ленты | |
| Возвращаемое значение | Int | Длина ленты |

Таблица А.37 – Метод Resize

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | Resize | | | |
| Назначение | Изменение длины ленты | | | |
| Возвращаемое значение | void |  | | |
| Список параметров | Имя | | Тип | Описание |
| length | | Unsigned | Длина ленты |

Таблица А.38 – Метод CreateMemento

|  |  |  |
| --- | --- | --- |
| Имя метода | CreateMemento | |
| Назначение | Сохранение ленты и ее параметров во временное хранилище | |
| Возвращаемое значение | TapeMemento\* | Ссылка на временное хранилище |

Таблица А.39 – Метод SetMemento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | SetMemento | | | |
| Назначение | Загрузка ленты и ее параметров из временного хранилища | | | |
| Возвращаемое значение | void |  | | |
| Список параметров | Имя | | Тип | Описание |
| m | | TapeMemento\* | Ссылка на временное хранилище |

**2.1.4 Класс TapeMemento**

Таблица А.40 – Общие сведения о классе TapeMemento

|  |  |
| --- | --- |
| Назначение класса в проектном решении | Реализация временного хранителя для ленты |
| Соответствие логическому дизайну | TapeMemento |
| Размещение исходного кода | <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/TapeMemento.cpp>  <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/TapeMemento.h> |

Таблица А.41 – Атрибуты класса TapeMemento

|  |  |  |
| --- | --- | --- |
| Имя атрибута в программном коде | Тип атрибута | Описание |
| tap | Tape\* | Ссылка на временно сохраненную ленту |

Таблица А.42 – Метод TapeMemento

|  |  |  |
| --- | --- | --- |
| Имя метода | TapeMemento | |
| Назначение | Создание и инициализация мементо | |
| Возвращаемое значение | - |  |

Таблица А.43 – Метод SetState

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя метода | SetState | | | |
| Назначение | Сохранение ленты в класс | | | |
| Возвращаемое значение | void |  | | |
| Список параметров | Имя | | Тип | Описание |
| t | | Tape\* | Ссылка на объект, из которого будет сохранена лента |

Таблица А.44 – Метод GetState

|  |  |  |
| --- | --- | --- |
| Имя метода | GetState | |
| Назначение | Выдача сохраненной ленты | |
| Возвращаемое значение | Tape\* | Ссылка на сохраненную ленту |

**2.1.5 Класс rule**

Общие сведения о классе

Таблица А.45 – Общие сведения о классе rule

|  |  |
| --- | --- |
| Назначение класса в проектном решении | Хранение правила таблицы переходов |
| Соответствие логическому дизайну | rule |
| Размещение исходного кода | <https://github.com/NaturalSpanking/Turing/blob/master/Alpha/rule.h> |

Таблица А.46 – Атрибуты класса rule

|  |  |  |
| --- | --- | --- |
| Имя атрибута в программном коде | Тип атрибута | Описание |
| symbol | WideChar | Символ выставляемый в ленту при использовании правила |
| shift | Int | Сдвиг ленты при использовании правила |
| new\_state | Int | Новой состояние в которое должна перейти машина после использования правила |
| is\_breakepoint | Int | Маркер того, что данное правило помечено как точка останова при трассировке программы |