Руководство по лабе 3b

В папке **dialog** лежат коды для диалогового меню,
в папке **task** - коды самого класса,
в папке **test** - тесты для класса.

Немного про сам класс:

Класс (**Diagram**) описывает временную диаграмму. Временная диаграмма - это некоторый сигнал (здесь и далее понятия сигнал и временная диаграмма используются как синонимы), состоящий из элементов сигнала (элемент сигнала описывает структура **Element**). В памяти сигнал представляется как статический массив размером 10 элементов, элемент массива имеет тип **Element**.

Подробнее про структуру Element:

Каждый элемент задается одним из трех состояний (**NON, ZERO, ONE**) и длительностью в условных единицах времени. Состояния определены в перечислении **State**. Для вывода последовательности элементов (то бишь сигнала) на экран с каждым из состояний связано его символьное представление:

- NON: x
- ONE: -
- ZERO: _

Например, весь сигнал может выглядеть так: __xx-x__-x.
Данный сигнал состоит из 7 элементов:

1. __ (состояние элемента: ONE, длительность: 2, т.к. две нижние черточки)
2. xx
3. -
4. x
5. __
6. -
7. x

В структуре Element также определены следующие конструкторы:

1. Element() - задает элементу состояние NON и длительность - 1.

2. Element(char *str) - задает элементу состояние, описанное в строке и длительность, равную длине строки. Если в строке представлено несколько состояний - выбрасывает исключение.

Также в структуре Element заданы следующие методы (getters и setters не описываю - их предназначение очевидно):

1. static char stateToChar(State state) - возвращает соответствующее переданному состоянию его символьное представление.

2. static State charToState(char symbol) - возвращает соответствующее переданному символу элемент перечисления State. При недопустимом значении символа выбрасывает исключение.

Подробнее про класс Diagram:

Состояние класса описывается тремя полями:

1. static const int N = 10 - целочисленная константа задает размер статического массива.

2. int size - содержит в себе текущее кол-во элементов массива

3. Element elements[N] - массив элементов. Логически это и называется сигналом.

Конструкторы:

1. Diagram() - задает size = 1 и длительность 0-ого эл-та равной N (состояние 0-ого элемента по-умолчанию равно NON).

2. Diagram(char *str) - по переданной строке строит сигнал. При кол-ве элементов, заданных в строке, больше N - выбрасывает исключение.

3. Diagram(Element e) - задает size = 1 и устанавливает параметры 0-ого элемента равным параметрам переданного элемента.

4. Diagram(State state) - задает size = 1 и устанавливает состояние 0-ого элемента равным state и его длительность - N.

Методы - перегруженные операторы:

1. friend ostream &operator << (ostream &, const Diagram &) - вывод временной диаграммы в поток вывода.

2. friend Diagram & operator << (Diagram &d, char *) - добавляет в конец сигнала диаграммы переданный в виде строки другой сигнал.

3. friend Diagram operator + (const Diagram &, const Diagram &) - объединяет сигналы из переданных диаграмм в один новый и возвращает новую диаграмму, содержащую объединенный сигнал.

4. void operator >>= (int) - циклически сдвигает сигнал на заданное число временных единиц вправо.

5. void operator <<= (int) - циклически сдвигает сигнал на заданное число временных единиц влево.

6. void operator *= (int) - копирует текущий сигнал заданное число раз. Если получившийся в итоге сигнал имеет кол-во элементов больше N, то выбрасывает исключение, а копирование отменяется.

7. void operator () (Diagram &, int) - заменяет текущий сигнал с момента времени, переданного вторым параметром, сигналом, содержащимся в переданной диаграмме.

8. void operator += (const Diagram &) - изменяет текущий сигнал, добавляя в конец сигнал из переданной диаграммы. Если получившийся сигнал имеет кол-во элементов больше N, то выбрасывается исключение, а добавление не производится.

9. char *toStr() - переводит сигнал в строковое представление.

Некоторые функции, не определенные в классе, но используемые для работы его методов:

1. int parse(Element *elements, char *str) - переводит сигнал из строкового представления в набор элементов типа Element. Результат записывается в переданный массив elements. Возвращает кол-во переведенных элементов

2. int countElems(char *str) - считает кол-во элементов в строковом представлении сигнала.

3. int findIndex(Diagram &d, int time) - возвращает индекс элемента сигнала, заданного в диаграмме d, соответствующего времени time.

4. State findState(Diagram &, int time) - возвращает состояние элемента сигнала, заданного в диаграмме d, соответствующего времени time.

5. Element *toBuffer(Diagram &d, int time) - возвращает массив элементов, содержащихся в сигнале диаграммы d с момента времени time.

6. void moveBack(Element *elems, int sizeOfElems, int start) - сдвигает элементы массива elems на один индекс влево, начиная с индекса start.

7. `void correctDuration(Diagram &d, int time)` - используется после применения функции toBuffer() для корректировки длительности граничного сигнала (сигнала, на котором происходило разделение).

8. `Element *toElements(char *str)` - возвращает массив элементов, полученный из строки str.

По вопросам: **tg** - @egor_perevoshchikov, **vk** - https://vk.com/id439012926