Лабораторная работа №5

«Построение графических изображений»

Цель работы: изучить работу множественного типа. Изучить Базовые принципы объектно-ориентированного программирования и их применение в разработке приложений с графическим пользовательским интерфейсом.

Задание 1.

Реализовать программу клавиатурный тренажёр на языке C++ с использованием графических компонентов. Раскладка клавиатуры и генерируемый текст должен автоматически подстраиваться под один из (на выбор пользователем в выпадающем меню) язык системы. Допустимые языки системы: Немецкий, Французский, Арабский, Китайский, Белорусский, Иврит.

Кроме нажатых клавиш должен отображаться таймер и индикатор количества слов в минуту (слово – набор элементов от пробела до пробела). Слова должны вырисовывать на экране и подсвечиваться разными цветами, в зависимости от того, правильно нажал пользователь на клавишу или нет. Предусмотреть открытия файла с текстом для запуска его в качестве тестового примера на тренажёре. Примеры на Рисунках 1-4.

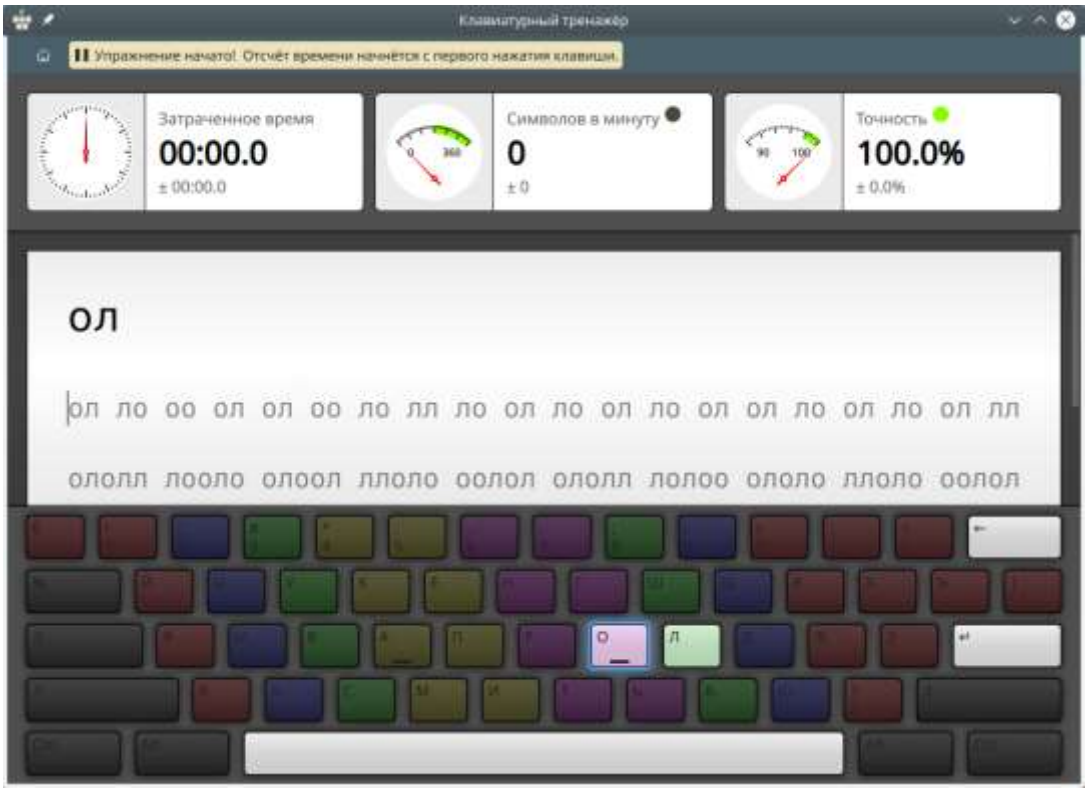

Рисунок 1.

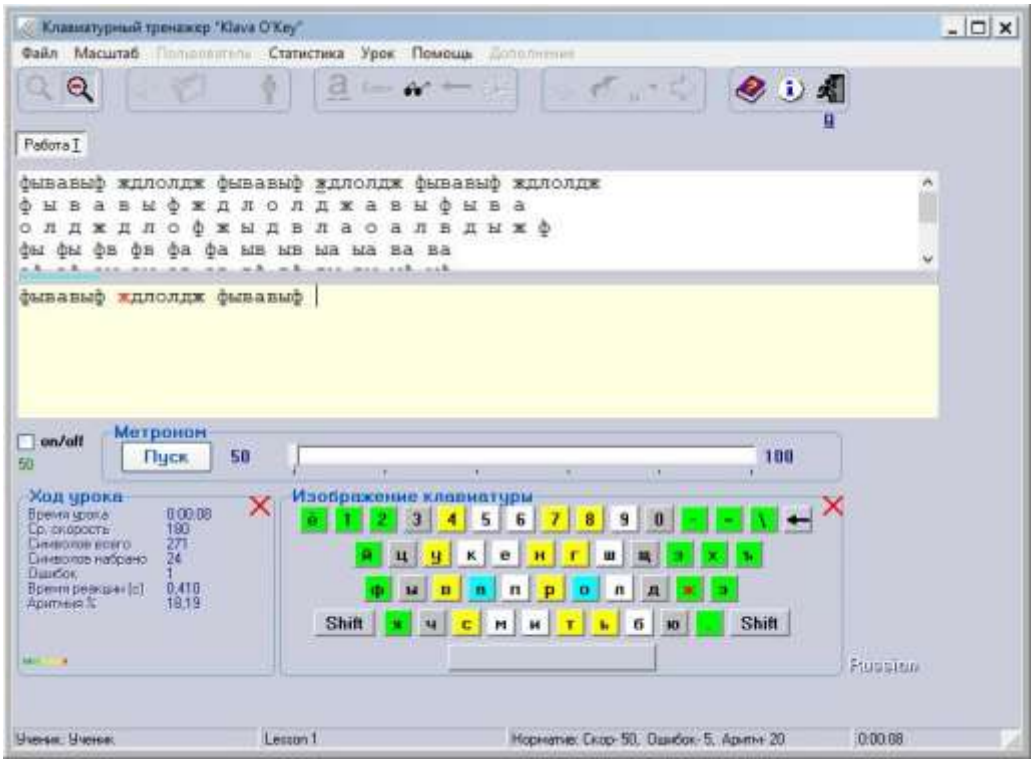

Рисунок 2.

Рисунок 3.

Рисунок 4.

Задание 2.

Реализовать динамическую библиотеку String (аналог std::string) с использованием простейшего итератора и умных указателей из библиотеки STL (т.к. библиотека динамическая, то указатели в данном задании не являются шаблонными).

Реализовать и продемонстрировать с помощью визуальных компонентов следующие функции:

- void* memcpy(void* s1, const void* s2, size_t n);
- void* memmove(void* s1, const void* s2, size_t n);
- char* strcpy(char* s1, const char* s2);
- char* strncpy(char* s1, const char* s2, size_t n);
- char* strcat(char* s1, const char* s2);
- char* strncat(char* s1, const char* s2, size_t n);
- int memcmp(const void* s1, const void* s2, size_t n);
- int strcmp(const char* s1, const char* s2);
- int strcoll(const char* s1, const char* s2);
- int strncmp(const char* s1, const char* s2, size_t n);
- size_t strxfrm(char* s1, const char* s2, size_t n);
- har* strtok(char* s1, const char* s2);
- void* memset(void* s, int c, size_t n);
- char* strerror(int errnum);
- size_t strlen(const char* s);

*не забывайте про реализацию нужных конструкторов и операторов копирования (Правило трех).

Данных задач достаточно, чтобы защитить лабораторную на минимальную оценку.

Задание 3.

Реализовать класс BitSet и продемонстрировать его работу при помощи визуальных компонентов.

Кроме геттеров, сеттеров, &operator[], operator[] (index) const, необходимы следующие функции:

all	Проверяет все биты в этом параметре, bitset чтобы определить, все ли они имеют значение true .
any	Функция-член проверяет, равен ли какой-либо бит в последовательности 1.
count	Эта функция-член возвращает количество бит, заданных в последовательности бит.
flip	Инвертирует все биты в bitset или инвертирует один бит в указанной позиции.
none	Проверяет, присвоено ли хотя бы одному биту в объекте bitset значение 1.
reset	Сбрасывает все биты в bitset в значение 0 или сбрасывает бит в указанной позиции в 0.
set	Присваивает всем битам в bitset значение 1 или присваивает биту в указанной позиции значение 1.
size	Возвращает количество бит в объекте bitset.
test	Проверяет, присвоено ли биту в указанной позиции в bitset значение 1.
to_string	Преобразует объект bitset в строковое представление.
to_ullong	Возвращает сумму значений бит в bitset как unsigned long long.
to_ulong	Преобразует объект bitset в unsigned long, чтобы получить последовательность его битов, если используется для инициализации bitset.

Операции ~,&,| должны работать за O(N/16) (или 32 или 64), а другие за O(1)