Модули

Console/

BDigits.h/BDigits.cpp

void outDigits(const string &s) – получает на вход строку и выводит ее в виде больших символов.

Console.h/Console.cpp

void showConsoleCursor(bool flag) – показывает/скрывает курсор в зависимости от flag.

void ignorePreviousInput() – игнорирования предыдущего ввода, нужно после получения нажатий с клавишь.

void setCursor(short x, short y) – устанавливает курсор в указанную позицию.

void setColor(int color) – устанавливает цвет текста(color 0-15).

CorrectInput.h/CorrectInput.cpp

bool fileExists(const string &name) – проверка файла на существование.

string getCorrectInput(ifstream &in, ofstream &out, const function<bool(string)>& predicate) – функция для получения корректных данных, на вход принимает поток ввода, вывода, функцию валидации ввода, возвращает строку, которая прошла проверку.

bool checkClockFormat(const string &s) – проверка ClockFormat, true если HH_MM_SS, MM_SS, SS, иначе false.

bool checkClockType(const string &s) - проверка ClockType, true если TIMER/STOPWATCH, иначе false.

bool checkTime(const string &s) – true если строка содержит только цифры, иначе false.

bool checkAudioPath(const string &s) – true если файл существует, и имеет расширение .wav, иначе false.

bool checkSeparator(const string &s) – true если ‘-’ ‘:’ ‘.’, иначе false.

Menu/

MenuItem.h/MenuItem.cpp

Поля(private):

- string name – названия элемента меню.
- function<void()> action – функция которая выполнится по нажатию.

Методы(public):

- runAction() – запускает функцию.
- display(short x, short y) – выводит элемент в указанную позицию.

Конструкторы:

- MenuItem(string name, function<void()> action).

Menu.h/Menu.cpp

Поля(private):

- vector<MenuItem> items – элемент меню.
- int selected – выбранный элемент.

Методы(public):

- void display(short x, short y) – выводит меню начиная с указанной позиции
- void moveUp() – переместиться выше.
- void moveDown() – переместиться ниже.
- void executeSelected() – запуск функции на текущем элементе.
- int getSelected() – геттер для selected.

Конструкторы:

- Menu(vector<MenuItem> items).

KeyTracker/

KeyTracker.h/KeyTracker.cpp

Поля(private):

- bool isRunning – состояние запуска.
- bool isPaused – состояние паузы.
- thread keyTrackerThread – поток трекера.

Методы(private):

- virtual void onSpacePressed() = 0.
- virtual void onEscapePressed() = 0.
- virtual void onBackspacePressed() = 0.
- virtual void onEnterPressed() = 0.
- virtual void onArrowUpPressed() = 0.
- virtual void onArrowDownPressed() = 0.
- virtual void onArrowLeftPressed() = 0.
- virtual void onArrowRightPressed() = 0.

Методы(public):

- void start() – запуск потока.
- void stop() – завершение потока.
- void pause() – пауза потока.
- void resume() – возобновление потока.
- void wait() – неблокирующие ожидание через join().
- void waitBlock() – блокирующие ожидание через whie.

Конструкторы:

- KeyTracker().

ClockKeyTracker.h/ClockKeyTracker.cpp

Наследован от KeyTracker

Дополнительные поля(private):

- Clock *clock – указатель на часы, которыми нужно управлять.

Методы нажатия на SPACE, ESCAPE, BACKSPACE перегружены, для управления часами.

Конструкторы:

- ClockKeyTracker(Clock *clock).

MenuKeyTracker.h/MenuKeyTracker.cpp

Наследован от KeyTracker

Дополнительные поля(private):

- Menu *menu – указатель на меню, которым нужно управлять.
- short x, y – координату, куда нужно выводить меню.

Методы нажатия на ENTER, ARROW_UP, ARROW_DOWN перегружены, для управления меню.

Конструкторы:

- MenuKeyTracker(Menu *menu, short x, short y);

Config/

ConfigParser.h/ConfigParser.cpp

Поля(private):

- map<string, string> config – мапка для хранения параметров.
- vector<int> loops – массив для хранения кругов.

Методы(private):

- bool optionExist(const string &key) – проверят существует ли ключ в config.
- static pair<string, string> parseOption(const string &s) – получает на вход строку из конфиг-файла и возвращает пару ключ значения.

Методы(public):

- bool loadConfig(const string &path) – загрузить конфиг из файла.
- bool saveConfig(const string &path) – выгрузить конфиг в файл.
- void setOption(const string &key, string value) – установить параметр key, значение value.
- ClockFormat getClockFormat() – получить ClockFormat из конфига.

- ClockType getClockType() – получить ClockType из конфига.
- int getTime() – получить время(для таймера) из конфига.
- string getAudioPath() – получить путь к звуку из конфига.
- string getSeparator() – получить разделитель из конфига.
- string getMessage() – получить сообщение(для таймера) из конфига.
- bool validate() – проверить конифг на валидность.
- void addLoop(int time) – добавить круг.

Конструкторы:

- ConfigParser().

Clock/

Clock.h/Clock.cpp

Поля(private):

- int seconds – текущее время.
- int startSeconds – стартовое время(нужно, чтобы выполнить сброс).
- bool isRunning – состояние запуска.
- bool isPaused – состояние паузы.
- string audioPath – путь к звуку.
- string message – сообщение по окончанию.
- thread clockThread – поток для счета секунд.
- thread delayedFunctionsThread – поток для отложенных функций.
- thread repeatedFunctionsThread – поток для повторяющихся функций
- map<int, vector<function<void()>>> delayedFunctions – храниние отложенных функций
- map<int, vector<function<void()>>> repeatedFunctions - хранение
- ClockType clockType – тип часов
- ClockFormat format - формат
- string separator – разделитель

Методы(private):

- void runClockThread() – запуск потока для счета секунд
- void runDelayedFunctionsThread() – запуск потока для отложенных функций.
- void runRepeatedFunctionsThread() – запуск потока для повторяющихся функций.
- string formatTime() – форматирует текущее время в строку, согласно формату

Методы(public):

- void start() – запуск.
- void stop() – завершение.
- void pause() – пауза.
- void resume() – возобновление.
- void reset() – сброс(секундомер к 0, таймер к стартовому времени)
- void wait() – неблокирующие ожидание join().
- void waitBlock() – блокирующие ожидание while.
- bool getIsRunning() – геттер для isRunning
- bool getIsPaused() – геттер для isPaused.
- int getSeconds() – геттер для секунд.
- ClockType getClockType() – геттер для clockType.
- void addDelayedFunction(int time, const function<void()>& function) – добавление отложенной функции.
- void addRepeatedFunction(int delay, const function<void()>& function) – добавление повторяющийся функции.

Конструкторы:

- Clock(ClockFormat format, string separator).
- Clock(ClockFormat format, string separator, int time, string message, string audioPath).

ClockUI.h/ClockUI.cpp

void runTimer() – запуск таймера.

void runStopwatch() – запуск секундомера.

void createConfig() – запуск создания конфига.

void runClockUI() – запустить полноценный интерфейс.

ClockUI

При запуске вас встречает меню, в котором вы можете перещаться при помощи стрелок на клавиатуре.

Запуск производится нажатием клавиши ENTER.

В меню есть 4 базовые функции:

- Запустить таймер
- Запустить секундомер
- Создать конфиг
- Выход

Помимо этих пунктов можно будет увидеть и другие пункту, с припиской S или T в конце – это ранее созданные конфиги.

S – конфиг секундомера.

T – конфиг таймера.

При запуске секундомера/таймера начинается отсчет времени. Для управления можно использовать следующие клавиши:

- SPACE – пауза/продолжить
- BACKSPACE – ресет
- ESCAPTE – выход

По окончанию таймера, чтобы остановить звук, нажмите SPACE.

При создании конфигов можно некоторые параметры задавать как null, тогда при запуске конфига, будут спрашиваться параметры которые помечены как null.