МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования «Крымский федеральный университет имени В.И. Вернадского»

**ТАВРИЧЕСКИЙ КОЛЛЕДЖ**

(структурное подразделение)

**КУРСОВАЯ РАБОТА**

по **МДК.1.2 Прикладное программирование**

на тему:

**«Реализация бота выполняющего функции нажатия клавиш клавиатуры и мыши по алгоритму заданному пользователем»**

Выполнил: **Халевин Кирилл Александрович** \_\_\_\_\_\_\_\_\_\_\_

Специальность: 09.02.03 Программирование в компьютерных системах

Группа **3ПКС10**

Научный руководитель: **Михерский Ростислав Михайлович** \_\_\_\_\_\_\_\_\_\_\_

Г. Симферополь, 2018

Оглавление

[Введение 3](#_Toc504082777)

[Глава 1 – Автоматизация и области её применения 7](#_Toc504082778)

[1.1 Автоматизация с которой мы сталкиваемся. 7](#_Toc504082779)

[1.2 Виды автоматизации в Windows 8](#_Toc504082780)

[Глава 2 – Составление плана и создание бота 10](#_Toc504082781)

[2.1 Составляем план программы 10](#_Toc504082782)

[2.2 Написание и создание бота 11](#_Toc504082783)

[Глава 3 – Применяем бота на практике 21](#_Toc504082784)

[3.1 Напишем на клавиатуре сообщение Hello world! 21](#_Toc504082785)

[3.2 Выключим компьютер 22](#_Toc504082786)

[3.3 Поменяем время на компьютере и будем повторять это в цикле 23](#_Toc504082787)

[Выводы 24](#_Toc504082788)

[Список литературы 25](#_Toc504082789)

# Введение

Каждый день мы выполняем большое количество действий на компьютере. Эти действия мы выполняем при помощи устройств ввода - клавиатуры и мышки. И бывает, что новые компьютерные операции мы выполняем снова и снова, и нам это надоедает. И после нескольких выполненных повторяющихся действий, разные люди выбирают разные пути решения своих задач. Один человек будет повторять действия снова и снова, другой же увидит в этом задачу. Первый продолжит повторения, пока второй начнёт копания в решении задачи. Он будет искать методы решения и испытывать их на практике, он уделит время на поиск решения, пока первый делает то что ему нужно и получает свой результат решения задачи, который не получает второй пытаясь решить проблему другим путём. Но потом второй может перегнать первого по результатам и сэкономить своё время, конечно в зависимости от твёрдости повторных действий.

Интересно, как же это будет выглядеть в формулах и графиках?

Пусть есть два человека - Пётр и Василий. Обоим нужно делать одни и те же повторяющееся действия. Пётр делает действия снова и снова, а Василий ищет другие решения и пока не получает результат. Внесём обозначения, h1 – число часов Петра, а h2 – число часов Василия, которые они уделили на выполнение повторяющихся действий. Следовательно r1 – результат Петра и r2 – результат Василия. Запишем формулы результатов Петра и Василия.

h1 = r1 – результат Петра

h2 = r2 – результат Василия

Но исходя из примера, Василий не получает результат. Значит r2 = 0 и количество часов работы h2 = 0. Допустим Пётр проработал уже 10 часов, тогда h1 = 10. Отобразим прямолинейные функции на графике (Рисунок 1).

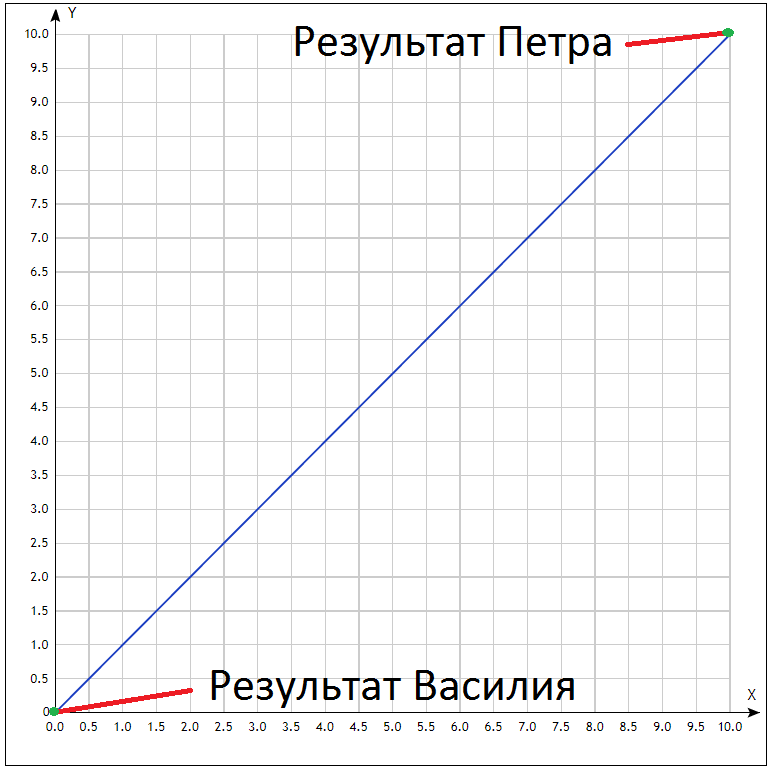


Рисунок 1 – График функции результата \*yotx.ru - Построение графиков функций онлайн

Результат Петра больше результата Василия и получается что Василию не выгодно искать решения по данным правилам. Но однако с другим решением действуют другие законы.

Пусть, Василий, за эти 10 часов сделал программу которая выполняет нужные ему повторяющиеся действия. И пусть задача имеет идеальную твёрдость алгоритма f = 100%. То есть алгоритм, в котором не требуется осознанный выбор человека и который машина сможет спокойно выполнить. Добавим также переменную время машины – m и коэффициент полезного действия η который равен:

η = результат машины / результат Василия

Тогда теперь новая программа Василия делает новый результат по следующей формуле

r2 = η\*f\*m

Например КПД машины будет равно 200%, то наша функция будет выглядеть таким образом(Рисунок 2)

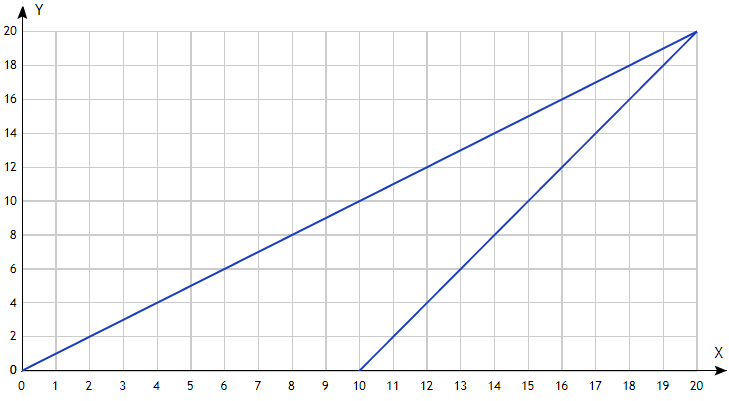


Рисунок 2 – Функция результата Василия за счёт машины \*yotx.ru - Построение графиков функций онлайн

На графике мы видим, что Василий за счёт машины догнал в результате Петра за 10 часов. И в дальнейшем Василий обгонит Петра по результатам. Василий изменивший правила и создавший новые условия быстрее поднял свой результат, но даже если КПД будет 100%, он освобождает своё время, которое в дальнейшем может вложить в решения других задач или просто отдохнуть. Конечно в моём примере очень идеальная модель и в реальной жизни имеется большое множество различных факторов-переменных, каждые из которых оказывают своё влияние на ситуацию которые в разные моменты могут как и сильно помогать, так и сильно ухудшать ситуацию.

Цель - сделать программу, в которой пользователь может задать нужные ему действия из списка для взаимодействия с графическим интерфейсом операционной системы Windows и по нажатию горячих клавиш, запустить заданный алгоритм. Чтобы пользователь мог без знаний языков программирования выполнять повторяющиеся действия и при этом не участвуя самому в процессе. Дать возможность пользователю составлять свой лист действий и функцию сохранения этого листа. Благодаря чему он сможет в дальнейшем использовать свой лист действий после закрытия программы, либо открыть его на другом компьютере.

Правда одним из недостатков программы является таким – программа работает с графическим интерфейсом. А для удобства пользователя, графический интерфейс является гибкой системой. Так например пользователь может указать алгоритм действий, которые включат музыку в браузере:

1. Открыть браузер;
2. Зайти в социальную сеть vk.com;
3. Нажать на плеер музыки – играть.

В этом листе может возникнуть проблема на каждом шаге. Значок браузера не может быть перемещён на другое место рабочего стола или меню пуска, иначе первый блок придётся переделывать. Так же нам важно чтобы никакие программы нам не перекрывали видимость нашего браузера. На втором шаге, когда бот успешно открыл браузер может возникнуть ситуация в которой браузер восстанавливается по прошлой длине и ширине, если пользователь самостоятельно сидел в браузере. В этом случае бот не сможет нажать на нашу вкладку vk.com. Однако если же первые блоки проработали как надо, то на самом сайте могут быть изменения. Изменения могут быть как со стороны сервера или же разработчиков если они изменят расположение плеера, так и со стороны самого пользователя если он в браузере вдруг поменял масштаб страницы.

# Глава 1 – Автоматизация и области её применения

## 1.1 Автоматизация с которой мы сталкиваемся.

Автоматизация – это процесс в котором занятность человека переходит в руки машины или механизма(хотя у большинства машин и механизмов нет рук, но да ладно) и способствует улучшению продуктивности. Автоматизация в наше время это уже обыденность и используется в различных отраслях. Например Вы могли участвовать в процессе автоматизации, если пополняли счёт телефона или продлевали домашний Интернет через терминал. В этом случае терминал узнал у Вас куда Вы хотите пополнить деньги и на какой счёт. Просканировав деньги и убедившись в том, что они действительны он отправляет их через Интернет. В данном случае терминал сам передал информацию о нашем платеже и сам его оплатил.

А в бизнесе автоматизацию может выполнять например веб-сайт. Веб-сайт убирает необходимость делать определённые шаги которые находятся в бизнес модели. Он увеличивает продуктивность, число заказов и доход предпринимателя. Так например клиенты заходят в Интернет, вбивают в поисковик нужный им продукт или услугу, заходят на сайт и после этого сайт позволяет не делать два шага. Первым шагом сайт выполняет функции презентации. На нём клиент узнает характеристики продукта, его стоимость и интересные предложения которые он может получить купив продукт. Если же презентация рассказала о выгодном предложении за минимальное время и привлекла клиента, то следующий шаг который сделает сайт автоматически – это разделение клиентов на группы. Разделение на тех, кого не устраивает предложение и которые уходят с сайта. И на группу людей, которые стали потенциальными клиентами и оставили свои контактные данные для уточнения некоторых вопросов и для дальнейшей покупки. То есть сайт сам прошёл за предпринимателя или отдела продаж этап презентации и выявил группу потенциалов. А дальше остаётся только работа с возражениями которые возникли после презентации.

## 1.2 Виды автоматизации в Windows

Рассмотрим варианты автоматизации в информационной области, в операционной системе Windows. Есть несколько способов как можно автоматизировать работу:

* Один из способов – это написание скриптов. В Windows можно сделать скрипт на VBS и этот скрипт нажмёт нужные клавиши. Или можно скачать бесплатный продукт AutoIt и написать скрипты под него. Недостатком этого способа заключается в том, что пользователю необходимо уметь писать скрипты и знать минимальную документацию языка.
* Скачать программу которая выполняет действия решающие похожие задачи между собой. Например калькулятор решает разные математические примеры требуя лишь входные данные. Или например программа-чистильщик которая по нажатию на кнопку – ‘Очистить’, сама находит временные файлы на компьютере и после, избавляется от них чтобы освободить место.
* А можно скачать готовую программу в которой не нужны знания языков программирования. Внутри которой пользователь сам задаёт действия которые ему нужны и по нажатию горячих клавиш, программа переводит пользовательский список действий в реальные действия выполняющихся в системе или графическом интерфейсе. Такие программы сложно настраивать, но я бы назвал такую программу универсальной, так как ограничения зависят только от фантазии, желания и терпения пользователя. Хотя универсальной в ограниченном смысле.

Если в Интернете не находится уже готовых решений под вашу задачу, то лучше использовать первый вариант и писать скрипты, которые автоматизируют работу. Так как если человек умеет писать скрипты, то он может сделать что только не придумает. И нажать клавишу мыши или клавиатуры. Может также открыть файл, как в графическом интерфейсе так и во внутреннем потоке если это текстовый файл. Скрипт сможет редактировать эти текстовые файлы, а также создавать новые папки и по необходимости их удалять. Информацию в файлах можно сортировать, а любые действия можно загнать в цикл и повторять их снова и снова. Но если человек не знает языков программирования или плохо разбирается в программировании и у него есть задача – сделать клики там-то и там, при том что такие действия являются прямолинейными и они не меняются, то ему подойдёт программа в которой можно задавать свой алгоритм.

# Глава 2 – Составление плана и создание бота

## 2.1 Составляем план программы

Пусть пользователь сможет задавать 4 основных действия с своими поддействиями:

* Виртуальное нажатие клавиши на клавиатуре
  + Нажать и оставить клавишу
  + Отпустить нажатую клавишу
  + Нажать и сразу отпустить нажатую клавишу
* Виртуальное нажатие клавиши на мышке
  + Нажать и оставить нажатой
  + Отпустить нажатую клавишу
  + Нажать и сразу отпустить нажатую клавишу
* Задержка перед новым действием
* Повтор действий, которые уже есть в списке действий.

Сложность заключается в том, что каждое действие имеет разные параметры, а значит интерфейс программы будет содержать разное количество полей. В некоторых случаях пользователю нужно будет заменить одно действие на другое, а в некоторых случаях ему потребуется удалить действие. Сложно сделать гибкость. Чем большую гибкость имеет модель программы, тем сложнее и запутанней становится код. Сложный код повышает шансы на ошибки, а исправление этих ошибок может привести к менее устойчивому коду, что понижает в будущем гибкость кода.

Поля для параметров одного действия будут находится на одной панели. На ней так же будет кнопка удаления действия из списка и надпись с числом действия по последовательности в списке. При наведении на поля ввода, будут подсказки для пользователей. Кроме этого сделаем локализацию на два языка: русский и английский. В программе нужна ещё возможность сохранения и загрузки скрипта(алгоритма действий пользователя).

## 2.2 Написание и создание бота

Писать код и делать проект будем в среде разработки Delphi 7 (Рисунок 4). Delphi использует язык Pascal который знают многие работая с компилятором Turbo Pascal 7.0 (Рисунок 3).

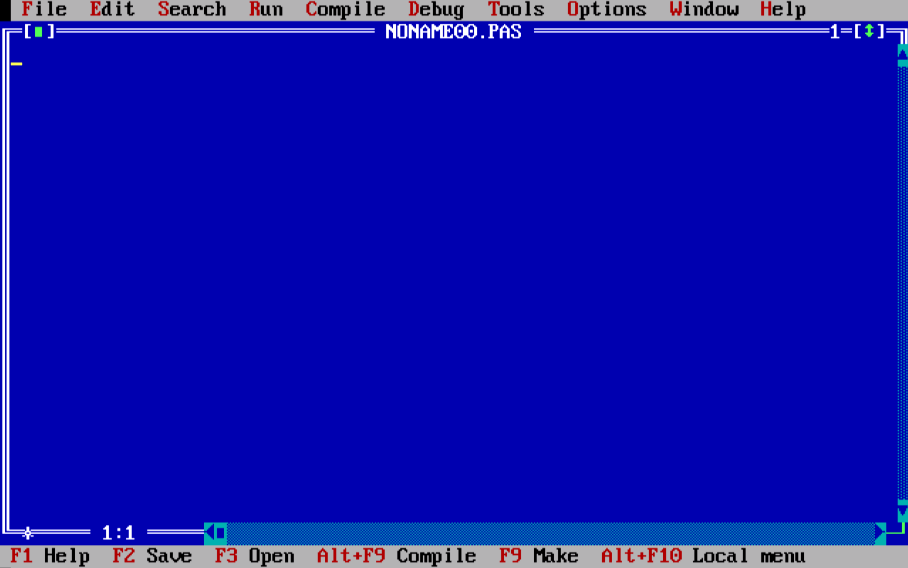


Рисунок 3 – Компилятор Turbo Pascal 7.0

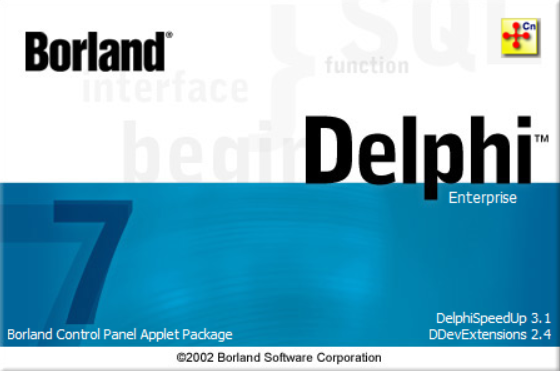


Рисунок 4 – Среда разработки Delphi 7

В новом проекте добавим несколько панелей с 4 действиями: клавиатура(Keyboard); повторение(Repeat); мышь(Mouse); задержка(Sleep) и в каждой панели добавим нужное количество полей и выпадающих списков (Рисунок 5).

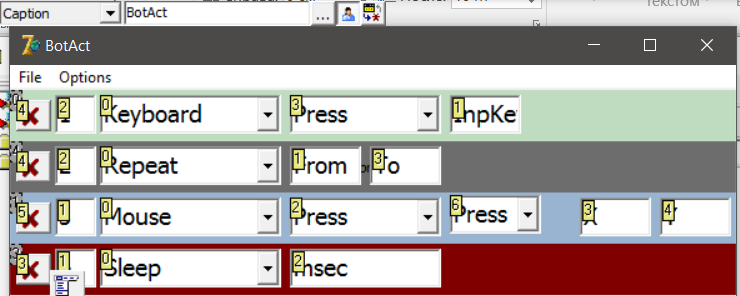


Рисунок 5 – Главная форма с 4 панелями

Добавим переменные и массивы объектов для динамического списка(Рисунок 6).

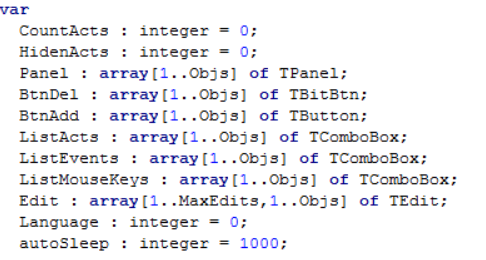


Рисунок 6 – Объявление переменных и массивов объектов

Добавим строковые константы для пользовательского интерфейса и вносим строковые константы для пользовательских подсказок на русском и английском языке почти для каждого объекта пользовательского интерфейса.

Зададим константы для размеров полей ввода и других объектов интерфейса (Рисунок 7)



Рисунок 7 – Константы размеров объектов

Добавим функцию с параметром addAct(ActPos : integer), которая добавляет панель с кнопкой удаления; полем ввода в котором находится номер действия; выпадающий список действий и кнопка для добавления новой панели с возможностью выбора нового действия. Параметр ActPos – позволяет узнать с какой панели пользователь создал новое действия.

Для этого добавим в класс TForm1 - procedure addAct(ActPos: integer);

И добавим в коде

procedure Tform1.addAct(ActPos: integer);

var Act : integer;

i : integer;

begin

end;

Добавим в функцию вот такой код(Рисунок 8)

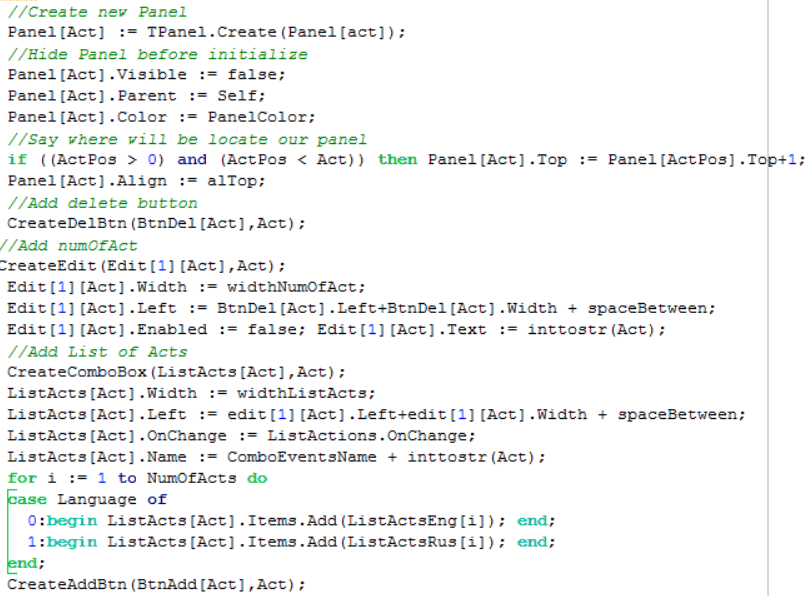


Рисунок 8 – код функции addAct(ActPos: integer)

Теперь запустим функцию addAct(0) и видим результат как на Рисунке 9

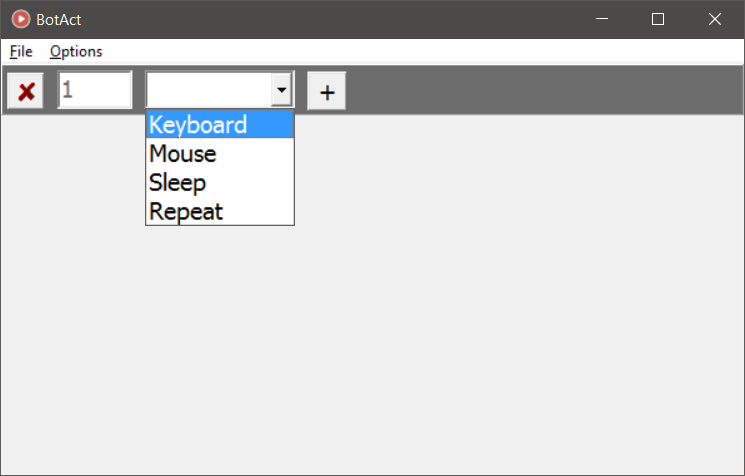


Рисунок 9 – Добавление первой панели при помощи функции addAct(0)

Функция addAct добавляет действие в конец и чтобы можно было добавить действие в середине, добавим ещё функцию сдвига панелей вперёд. Вносим procedure ShiftActsFrom(Var Frm : integer); в класс TForm1 и программируем нашу функцию.

procedure Tform1.ShiftActsFrom(Var Frm : integer);

var i,j: integer;

begin

for i := CountActs-1 downto Frm do

begin

ListActs[i+1].ItemIndex := ListActs[i].ItemIndex;

if (ListActs[i+1].ItemIndex <> -1) then

begin

ListActionsChange(ListActs[i+1]);

If (ListEvents[i] <> nil) and (ListEvents[i+1] <> nil) then begin

ListEvents[i+1].ItemIndex := ListEvents[i].ItemIndex;

end;

try If (ListMouseKeys[i] <> nil) then ListMouseKeys[i+1].ItemIndex := ListMouseKeys[i].ItemIndex; except end;

for j := 2 to MaxEdits do

if (Edit[j][i] <> nil) and (Edit[j][i+1] <> nil) then Edit[j][i+1].Text := Edit[j][i].Text;

ListActs[i].ItemIndex := -1; ListActionsChange(ListActs[i]);

end; end; end;

И сделаем ещё аналогичную функцию которая будет перемещать панели обратно и которая поможет нам после удаления панелей.

procedure Tform1.ShiftActsBack(Var B : integer);

var i,j: integer;

begin

for i := B to CountActs-1 do

begin

ListActs[i].ItemIndex := ListActs[i+1].ItemIndex;

ListActionsChange(ListActs[i]);

If (ListEvents[i] <> nil) and (ListEvents[i+1] <> nil) then begin

ListEvents[i].ItemIndex := ListEvents[i+1].ItemIndex;

end;

try If (ListMouseKeys[i+1] <> nil) then ListMouseKeys[i].ItemIndex := ListMouseKeys[i+1].ItemIndex; except end;

for j := 2 to MaxEdits do

if (Edit[j][i] <> nil) and (Edit[j][i+1] <> nil) then Edit[j][i].Text := Edit[j][i+1].Text;

end;

ListActionsChange(ListActs[CountActs-1]);

end;

Теперь дадим компьютеру инструкции, что делать когда мы выбираем действие из списка. Добавим в код создание и инициализацию новых объектов, занесём в них строковые константы в зависимости от языка и укажем их местоположение. Для новых полей укажем ссылки для новых функций которые допишем.

Для поля ввода параметра в действии keyboard, напишем функцию которая будет записывать в скрытое поле номер клавиши и специальные клавиши без символов записывать текстом(Рисунок 10)

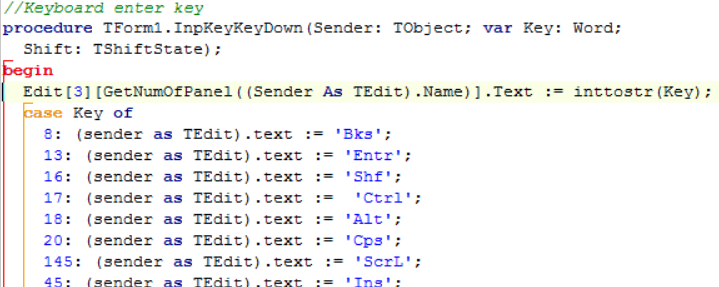


Рисунок 10 – Часть кода для ввода параметра в действии Keyboard

А для параметров в других действиях просто заблокируем ввод символов

procedure TForm1.InpIntKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in ['0'..'9',#8]) then key := #0

else if not (Length((Sender as TEdit).Text) = 0) and not ((Sender as TEdit).Text[1] in ['0'..'9']) then

begin (Sender as TEdit).Text := key; (Sender as TEdit).SelStart := length((Sender as TEdit).Text); key := #0; end;

end;

Добавим горячие клавиши при помощи функции Windows – RegisterHotKey. При создании формы, добавим функцию RegisterHotKey(Handle,1,MOD\_CONTROL,VK\_F1); чтобы назначить горячие клавиши Ctrl+F1. Теперь нажав в любом приложении Ctrl+F1, нажатия передаются в нашу программу. Напишем действия для программы, что делать после нажатия горячих клавиш. Для этого добавим функцию которая принимает наше сообщение.

В класс TForm1 добавляем procedure StartStop(Var Msg : TWMHotKey); и прописываем код в процедуре как на рисунке 11

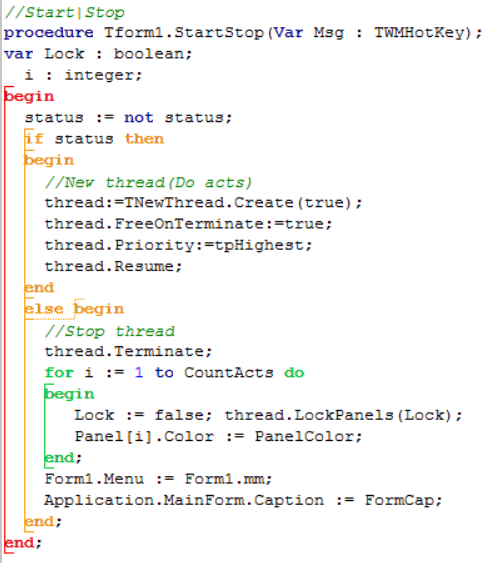


Рисунок 11 – Функция которая выполняется после нажатия горячих клавиш

По нажатию на клавишу у нас будет запускаться новый поток, либо если он уже в работе, то он остановится. Так как ещё потока нет, то добавим его.

var thread : TNewThread; //Поток

status : bool = false; //Статус потока

И добавим класс потока

type

TNewThread = class(TThread)

private

public

procedure Execute; override;

procedure DoAct(Var PanelN:integer);

procedure LockPanels(Var Lock:Boolean);

end;

Процедура Execute выполняется по созданию потока, но для упрощения кода добавим ещё две функции DoAct и LockPanels. Функция LockPanels в зависимости от параметра Lock блокирует или разблокирует панели пользовательского интерфейса. А функция DoAct вызывается в цикле и по параметру PanelN считывает какое действие нужно сделать и с какими параметрами его запускать.

Теперь сделаем возможность сохранения и загрузки. Переводим данные в числовые значения и получается что действия может принимать значения от 0 до 3, а дальше берём числа из параметров. Это делает наша процедура SaveScript(Var F : String). А процедура OpenScript(Var F : String) читает файл и вносит алгоритм пользователя заново в пользовательский интерфейс и в его список действий.

Создадим так же новую форму с настройками и сделаем возможность менять язык, горячие клавиши и добавим авто задержку(Рисунок 12).

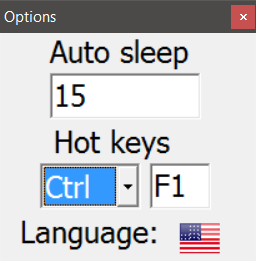


Рисунок 12 – Настройки

По нажатию на флаг меняется язык, а при изменении любых настроек изменения записываются в C:\Users\User\AppData\Local\BotAct\Options.bao

# Глава 3 – Применяем бота на практике

## 3.1 Напишем на клавиатуре сообщение Hello world!

Первый алгоритм который очень популярен у программистов в новой среде, это написание сообщения – Hello world! Создадим в боте алгоритм который сделает множество нажатий на клавиатуре(Рисунок 13)

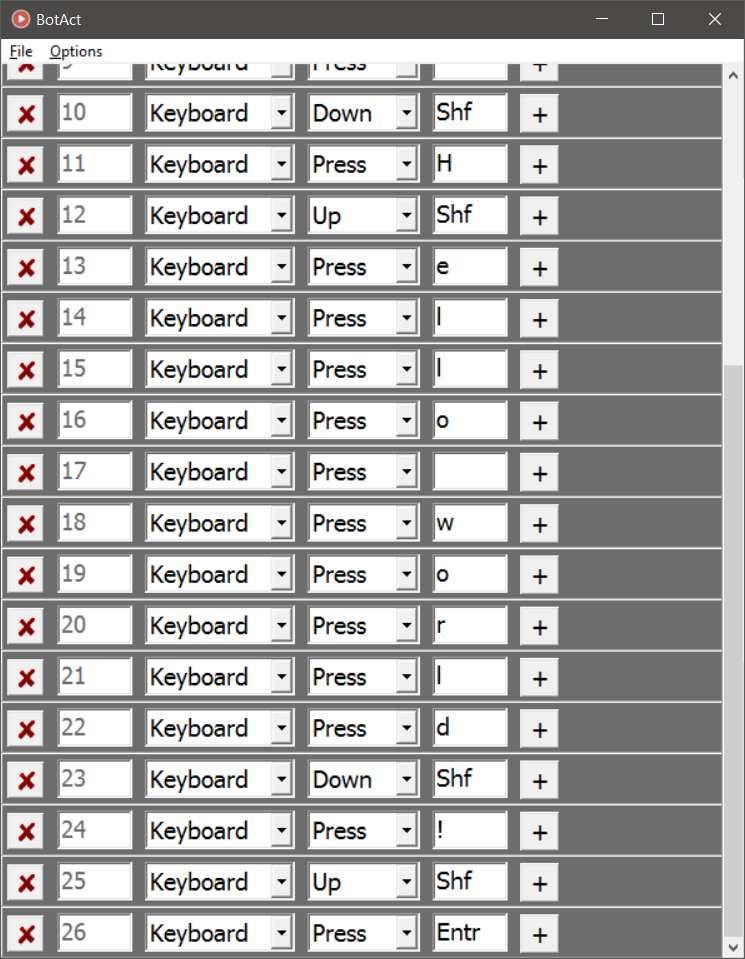


Рисунок 13 – алгоритм набора сообщения Hello world!

Теперь по нажатию на горячии клавиши у нас вводится сообщение Hello world!

## 3.2 Выключим компьютер

Теперь зададим алгоритм который выключит компьютер. Это можно сделать нажав на клавишу Windows и после выбрать выключение компьютера в меню пуск(Рисунок 14) или можно прописать команду(Рисунок 15)

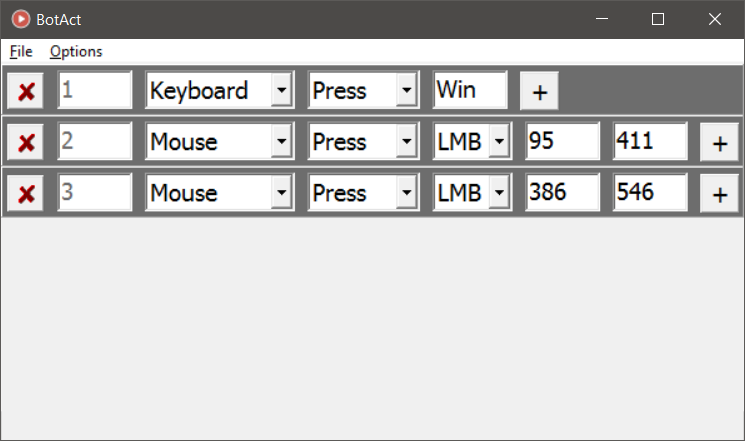


Рисунок 14 – алгоритм выключения компьютера первый способ

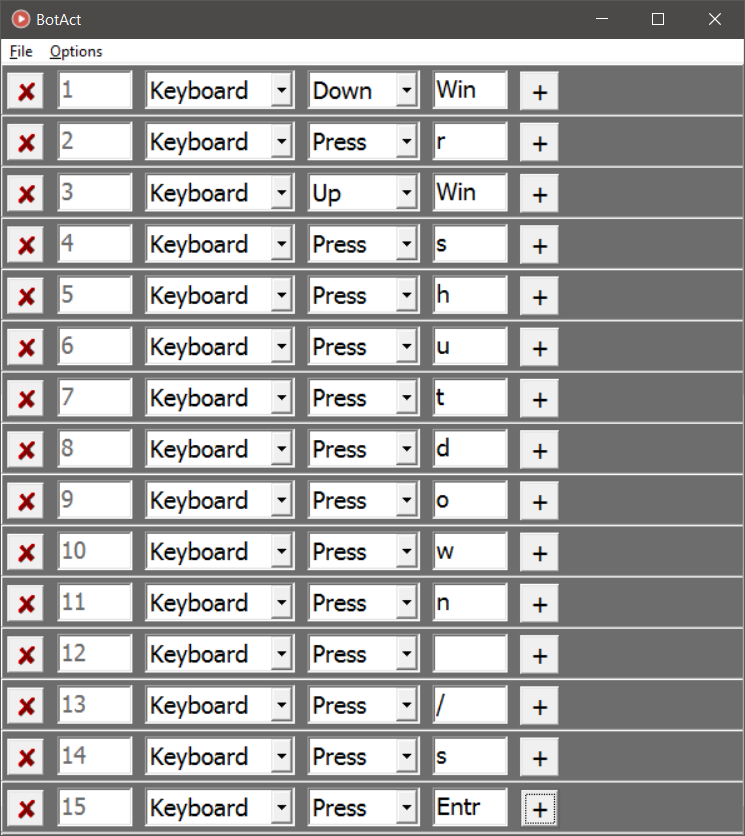


Рисунок 15 – алгоритм выключения компьютера второй способ

## 3.3 Поменяем время на компьютере и будем повторять это в цикле

Откроем командную строку. Для этого откроем меню пуск нажав клавишу Windows, введём в поиске cmd и откроем командную строку от имени администратора(В данном алгоритме требуется чтобы был выключен VAC), вводим time и вводим время на которое хотим изменить(Рисунок 16).

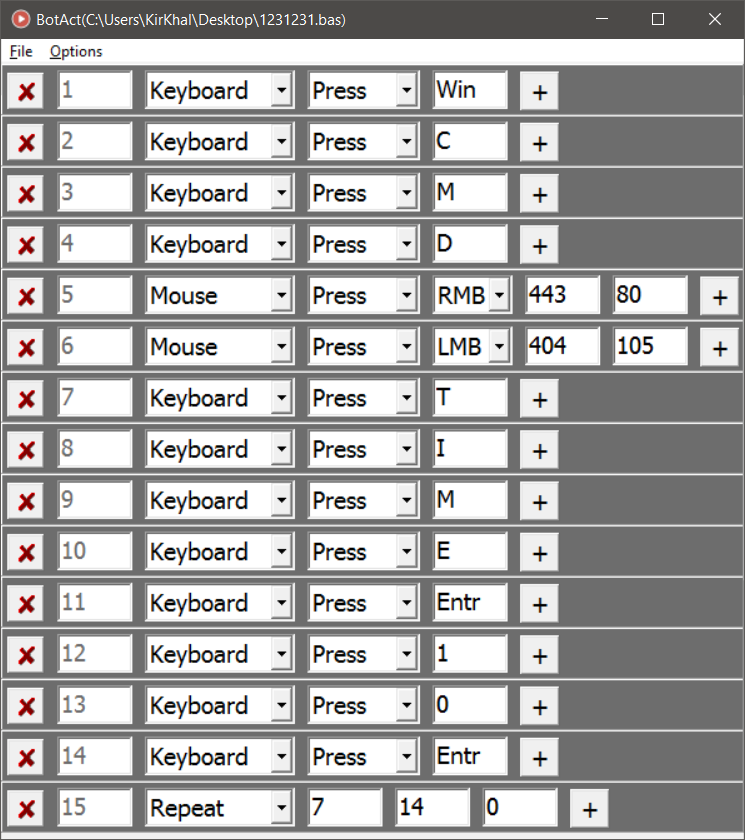


Рисунок 16 – Алгоритм изменения времени на 10 часов через командную строку с бесконечными числом повторов.

# Выводы

Цель достигнута! Бот-программа создана. В ней можно выбирать действия из списка и после чего программа будет нажимать клавиши клавиатуры и клавиши мыши. Между действиями пользователь может сделать авто задержку или добавить одну задержку после какого-то действия в своём алгоритме. И так же есть возможность повторять алгоритм любое количество раз. Алгоритм может быть запущен даже когда программа свёрнута при помощи горячих клавиш Ctrl+F1. А для написания алгоритмов в программе, пользователю не нужны знания программирования, бот сам программирует действия за него.

В программе реализована функция сохранения скриптов(алгоритмов созданных в программе) и их загрузка. Благодаря чему пользователь может составить несколько алгоритмов и хранить их на компьютере, а потом открыть один из нужных и запустить.

# Список литературы

1. Фленов М. Е. - Библия Delphi – 3-издание. Издательство “БХВ-Петербург” город Санкт-Петербург 2015г. 686 стр.
2. Фленов М. Е. - Программирование в Delphi глазами хакера – 2-издание. Издательство “БХВ-Петербург” город Санкт-Петербург 2007г. 480 стр.
3. Шапоров В. Я. и Кузан Д. Н. - Программирование Win32 API в Delphi. Издательство “БХВ-Петербург” город Санкт-Петербург 2005г. 368 стр.
4. Фленов М. Е. - Компьютер глазами хакера – 3-издание. Издательство “БХВ-Петербург” город Санкт-Петербург 20012г. 272 стр.
5. Книттель Брайан - Windows 7.Скрипты, автоматизация и командная строка. Издательство “Питер” город Санкт-Петербург 2012г. 784 стр.
6. Попов А. В. - Введение в Windows PowerShell. Издательство “БХВ-Петербург” город Санкт-Петербург 2012г. 464 стр.
7. Чекмарев А. Н. - Microsoft Windows 7 для пользователей. Издательство “БХВ-Петербург” город Санкт-Петербург 2010г. 560 стр.
8. Гукин Дэн - Компьютер для чайников. Издание для Windows 7. Издательство “Вильямс” город Москва 2011г. 336 стр.
9. Престон Гралла - Windows 8. Трюки. Издательство “Питер” город Санкт-Петербург 2013г. 416 стр.