Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

кафедра Информатики

Дисциплина: Программирование

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему

ВИРТУАЛЬНАЯ КЛАВИАТУРА

Студент: гр. 753505 Снегур А.Н.

Руководитель: ассистент кафедры информатики Богушевский Е.Л.

Минск 2018

Содержание

1. **Введение**

1.1 История клавиатуры.....................................................3

1.2 О Microsoft Visual Studio.............................................13

1. **О программе**

2.1 Внешний дизайн.........................................................20

2.2 О том, что внутри........................................................26

1. **Заключение**

3.1 Преимущества программы.............................................36

3.2 Список литературы..........................................................36

Введение

**Клавиатура компьютера** — устройство для ввода информации в компьютер и подачи управляющих сигналов. Содержит стандартный набор клавиш печатной машинки и некоторые дополнительные клавиши — управляющие и функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

**Рождение клавиатуры**

Корни современной компьютерной клавиатуры уходят далеко в 19 век. Все началось с появления простой пишущей машинки. В 1868 году Кристофер Латам Шольз (Christopher Sholes) запатентовал свою дела как совокупность символов, располагавшихся в алфавитном порядке. Как оказалось позже, это, мягко говоря, неудобно, так как редко используемые символы находились на самых видных местах и наоборот. В 1890 году придумали раскладку «QWERTY», которую мы используем и до сих пор при наборе текста латинскими буквами. А русскую раскладку клавиш, как ни парадоксально, придумали в Америке в конце 19 века. С тех пор она не претерпела сильных изменений.

Название раскладки «QWERTY» происходит от первых шести латинских букв на клавиатуре, начиная от левого верхнего угла слева направо.

Ключевым моментом превращения печатной машинки в компьютерную клавиатуру стало изобретение в конце 19 века телепечатной машины Бодо. Этот метод заменил телеграф, в котором информация кодировалась двухбитным методом («точка-тире», а позднее «наличие сигнала – отсутствие сигнала»). В связи Бодо для кодирования букв алфавита использовался пятибитовый код, при помощи которого сложные электромеханические устройства печатали принимаемый текст на бумагу. Связь была синхронной, и телеграфист должен был нажимать на кнопку только при получении специального звукового сигнала. Позднее передача данных стала асинхронной, и такой способ связи получил название «телетайп» (дословно – «печать на расстоянии»). В 1920-х годах телетайп уже широко использовался для передачи финансовой и политической информации. Позднее принимающие устройства стали электронными, и в целях экономии бумаги текст выводился на экран и распечатывался только при необходимости.

**Первые компьютерные клавиатуры**

1943 год ознаменовался появлением компьютера ENIAC, который произвел фурор в мире науки. Этот компьютер использовался военными для баллистических расчетов. Исходные данные он получал посредством перфокарт и телетайпных лент. Программное управление операциями осуществлялось при помощи переключения штекеров и наборных панелей.

В 1948 году начинается разработка компьютеров UNIVAC и BINAC, предназначенных не для единичного, а для относительно более массового производства. Отдельное внимание в этих машинах было уделено устройствам ввода-вывода. Средствами ввода-вывода для них служили телетайпы или табуляторы-перфораторы. BINAC мог записывать информацию на магнитную ленту.

**Емкостная клавиатура**

1960 год является переломным моментом в истории развития компьютерных клавиатур – на рынок выходит электрическая печатная машинка. Она имела емкостную клавиатуру.

Емкостная клавиатура производилась на печатных текстолитовых платах. Название технологии говорит само за себя – этот тип работает за счет конденсаторов, расположенных в устройстве. Две площадки из олова и никелированной меди, которые, кстати, никак не соединены друг с другом, ни механически, ни электрически, формируют каждую клавишу. Рассмотрим, как это работает. Мы нажимаем клавишу – расстояние между площадками увеличивается, при этом изменяется электрическая емкость, которая при нажатом состоянии равна приблизительно 2 пикофарадам, а при не нажатом – 20 пикофарадам. Уменьшение емкости создает поток заряженных частиц, который обрабатывается контроллером клавиатуры, который, в свою очередь, генерирует код нажатой клавиши. Кажется долго, но при этом такая клавиатура позволяет вводить текст со скоростью до 300 символов в секунду.

Главным плюсом печатной машинки стала легкость ввода текста – теперь, чтобы печатать не нужно было прикладывать столько усилий, как например, на классической машинке Шольза.

**Новая эра**

Следующий этап связан с развитием многопользовательских систем, разделением машинного времени и появлением терминалов. ЭВМ оснащались графическими дисплеями и телетайпами.

В 1965 году лаборатории компаний Bell и General Electric объединились, чтобы создать принципиально новый вид многопользовательской операционной системы MULTICS, которая впоследствии привела к появлению ОС UNIX. Главной чертой проекта стало создание нового пользовательского интерфейса – видеотерминального показа. Теперь пользователи могли видеть, какой текст они набирают, и при этом имели возможность его сходу редактировать.

Примерно в конце 1970-ых и в начале 1980-ых годов производители стали выпускать компьютеры, которые могли себе позволить не только офисы крупных компаний, но и простые смертные для домашнего использования. Разновидностей и различных линеек было разработано предостаточно. Может, и в твоем доме когда-то стоял, например, Amiga или Spectrum. Эти компьютеры представляли собой обычную клавиатуру, в которую был встроен сам компьютер (процессор, ПЗУ, ОЗУ и несколько модулей для дополнительных устройств). Все это было выполнено компактно и не занимало много места. Безусловно, клавиатуры этих машин по функциям и по количеству клавиш превосходили электрические печатные машинки.

Так, например, был добавлен ряд клавиш типа Control, Alt. Клавиша с изогнутой стрелкой стала называться Enter и выполнять функцию не только возврата каретки, но и завершения ввода данных. Для работы с электронными документами были добавлены стрелки управления курсором, но пока они находились на цифрах. Эти элементы нашли свое активное применение в графических интерфейсах и, конечно же, компьютерных играх, которые активно начали развиваться, как только персональные компьютеры получили широкое распространение.

Следующий этап связан с развитием модульных ПК, которые имели возможность апгрейда. Самые яркие их представители – компьютеры компании Apple ( Apple PC ), Commandore , IBM PC . Все главные компоненты, такие как процессор, ОЗУ, были спрятаны в отдельный корпус, и, соответственно, клавиатура была сделана отдельно от них, то есть стала полноценным устройством. К компьютеру она подключалась с помощью шнура через коннектор Din-5 и выглядела следующим образом: всего 83 клавиши, разделенные на два блока. Первый блок – алфавитно-цифровой, на нем также располагались стрелки управления, и второй блок – служебный (для системных клавиш). Эта клавиатура не имела функции индикации положения клавиш заглавного регистра Caps Lock, а также блокировки служебно-цифровой клавиатуры Num Lock и блокировки просмотра Scroll Lock. Основным недостатком было размещение элементов управления. До функциональных клавиш приходилось тянуться, и печатать становилось просто неудобно, а кнопка Enter болталась где-то в уголке и не выделялась своими размерами среди остальных. Этот период тоже можно отнести к ключевому в истории клавиатур – ведь клавиатура стала полноценным устройством.

Компания Apple специализировалась, в основном, на офисных компьютерах, и именно ей принадлежит изобретение манипулятора «мышь».

**Недалекое прошлое**

В начале 1980-х был разработан стандарт архитектуры IBM PC, компьютеры потеряли определенную марку, и уже полностью устоялся магистрально-модульный принцип построения. Теперь производители занимались не выпуском конкретных компьютеров, а изобретали и усовершенствовали все новые и новые устройства. Разумеется, это касалось и клавиатур. Такие компании, как Cherry, Focus Electronic, KeyTronic и другие занимались и сейчас занимаются выпуском этих устройств. Прогресс не стоит на месте – новые компьютеры, соответственно, и новые клавиши.

В период, когда для АТ были выпущены только 286 процессоры, производители подстраивались под стандарты. Но для того чтобы хоть как-то выделиться, а может и опасаясь критики (оказывается, были еще и противники раскладки «QWERTY», которые хотели расположения в алфавитном порядке), производители начали вносить некоторые коррективы: например, была реализована смена раскладки прямо на ходу, для этого даже придумали отдельную клавишу. Некоторые фирмы разрабатывали универсальные клавиатуры, например, двухстандартная клавиатура AT простым переключением тумблера превращалась в PC/XT-совместимый девайс. Но с приходом новых технологий о PC/XT забыли.

Прошло несколько лет, и производители сделали очередной шаг в сторону стандартизации клавиатуры AT. Добавилось множество новых функций, в том числе и новая клавиша Sys Req. Но главным отличием новой клавиатуры от предшественниц являлся блок управления. Клавиатуру можно было программировать на собственные команды, что сделало ее полностью несовместимой с PC/XT, хотя разъем у них был один. Да, кстати, клавиша Enter стала значительно больше, и вообще устройство стало намного удобней для использования.

**Расширенная клавиатура**

Прошло не так много времени, и хотя выпуск AT-клавиатур продолжался, на конвейер была поставлена новая разработка. Ее стали называть расширенной клавиатурой. Несмотря на то, что в плане электроники это устройство было копией своего родителя AT, на нем поменяли расположение клавиш. Их число тоже увеличили, и новая клавиатура приобрела уже привычный для наших дней вид – 101 клавиша. Добавили «F11» и «F12» (кстати, все «функционалы» вынесли в отдельный верхний ряд), клавиши управления курсора были выделены в отдельный блок, клавиши Ctrl и Alt были продублированы и разнесены по обе стороны основного блока.

**Жесткоконтактная клавиатура**

Производители всегда стараются удешевить и упростить производство. Вот и пришла смена технологии. Новинка получила название «жесткоконтактная клавиатура». Производство емкостной клавиатуры являлось сложным процессом и, соответственно, достаточно дорогим. Новая технология пришла на помощь потребителю. В жесткоконтактной клавиатуре каждая клавиша работает как маленький выключатель. При нажатии клавиши в замкнутых проводниках начинает проходить электрический ток, и специальная цепь фиксирует его наличие. Далее сигнал передается микропроцессору самой клавиатуры и генерируется код, который отправляется в центральный процессор компьютера. Простота исполнения делает такой вид клавиатур очень дешевым. Например, клавиатура PCjr стала эталоном простоты. В ней даже использовались резиновые прокладки (вместо пружинок) для возвращения клавиш в исходное положение. Главным минусом такого решения является недолговечность самого устройства, зато дешево и сердито. Данная технология применяется в большинстве современных клавиатур.

Для PCjr компания IBM сделала попытку создать беспроводную клавиатуру: она разместила в ней инфракрасный датчик: излучатель на задней панели клавиатуры подавал сигнал, а приемник, подключенный к порту, его принимал. Для работы устройству требовалось 4 батарейки АА. Клавиатуру можно было использовать как со шнуром, так и без. При подключении провода запитка от батареек прекращалась, что позволяло экономить заряд.

**Защита от пользователя**

Не секрет, что лучше работается с чашкой кофе и свежим пончиком, поэтому пользователи часто закусывают перед компьютером, и, естественно, проливают сладкую жидкость и роняют жирную еду на несчастную клаву – она же ближе всего. От этого контактная пленка слипается, и клавиши перестают нажиматься. Производители начали делать между пленкой и клавишами еще один слой пластмассы, клавиши приобрели длинные штырьки, а отверстия для них были приподняты над поверхностью. Таким образом, залить современную клавиатуру киселем – непросто, хотя, если очень постараться, все же можно.

**ATX - клавиатура**

Новые компьютеры принесли нам, в первую очередь, улучшенный интерфейс управления питанием. Компьютер стало можно выключать и включать программно. Как известно, лень – двигатель прогресса, и чтобы не тянуться к кнопке на корпусе или не проделывать несколько операций мышкой, на клавиатуре добавили клавишу Power, по нажатию которой можно было выключить питание. Кнопка Sleep вводит компьютер в спящий режим, а также выводит из него. В свое время эти кнопки попортили немало нервов пользователям, привыкшим к 101-кнопочной клаве, пока их не догадались сделать меньше, скруглить и больше утопить в корпус.

Также изменился разъем подключения клавиатуры. Второе «рождение» получил PS/2, разработанный в первой половине 1980-х годов сразу вслед за стандартом IBM PC. Он намного меньше по сравнению со своим собратом Din-5 от АТ-клавиатуры.

**Мультимедийные клавиатуры**

Не только с развитием железа, но и с улучшением программного обеспечения в клавиатуру старались запихнуть все больше клавиш. Компания Microsoft, которая всю жизнь занималась производством программного обеспечения, стала выпускать клавиатуры и также приложила руку к появлению новых клавиш. Расположила она их между клавишами Control и Alt. Одна клавиша открывает меню «Пуск», а другая эмулирует нажатие правой кнопки мыши. Пришлось «откусить» кусок у клавиши пробела. На некоторых моделях клавиатур встраивают сенсорную панель для управления мышиным курсором.

Также появились кнопки управления мультимедиа, например, с помощью мультимедийной клавиатуры можно одним нажатием отрегулировать звук или получить полное управление над Windows Media Player. Примерно в то же время на клавиатуру были вынесены основные функции управления браузером.

**Эргономика клавиатуры**

Для тех, кто не знает, поясним: эргономика – это приспособленность устройства под биологические особенности человека. Это то, что мы ощущаем, насколько нам комфортно, как эффективно мы можем использовать устройство, и насколько все это красиво выглядит. Первыми эргономичными клавиатурами стала заниматься компания Cherry, и это несмотря на то, что она как никто другой всегда соблюдала классический стиль, вплоть до цвета девайса. Ее устройство выглядело следующим образом: главный буквенный блок разделен пополам, эти половинки чуть разведены под небольшим углом (этот угол фиксации можно было регулировать), в центре имеется горб. С этого момента пошла новая мода на создание эргономичных клавиатур. Эстафету подхватили компании Microsoft , Logitech , BTC ( Behavior Tech Computer ) и до сегодняшнего времени они являются лидерами в производстве подобных устройств.

**Взгляд в будущее**

В некоторые клавиатуры сейчас встраивают считыватели смарт-карт. Они служат, якобы, для безопасности, выполняя функцию ключа: вставил – вошел в операционную систему, не вставил – не вошел.

Также появились User-to-interface устройства, в частности, DataHand System не совсем клавиатура – больше напоминает терминал управления космическим кораблем. Кнопок как таковых нет, зато есть десять дырок, куда надо просовывать пальцы. Пальцами можно двигать в пяти направлениях, таким образом и надо печатать. Пока выучишь такой метод, да еще и с новой раскладкой... Разработчик ставил себе целью сократить до минимума количество выполняемых пальцами движений, но при этом сохранить возможность работать как с клавиатурой, так и с мышкой, причем одновременно.

Сенсорная клавиатура нового поколения – это уже вовсе не клавиатура. Есть только два сенсора, которые надо одевать на обе руки и печатать по воздуху. Если привыкнуть, то очень удобно будет использовать девайс для мобильных решений. Работает эта футуристическая разработка следующим образом: устройство объединяет сенсорную технологию с искусственной нейронной сетью, с помощью чего приемник точно отслеживает движения пальцев печатающего человека. Датчики реагируют на движения пальцев и преобразуют их в буквы. Новинка поддерживает раскладку «QWERTY». [1]

О Microsoft Visual Studio

**Microsoft Visual Studio** — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.  
  
**Интегрированная среда разработки** — система программных средств, используемая программистами для разработки программного обеспечения (ПО).  
  
Обычно среда разработки включает в себя:

* текстовый редактор
* компилятор и/или интерпретатор
* средства автоматизации сборки
* отладчик.

Иногда содержит также средства для интеграции с системами управления версиями и разнообразные инструменты для упрощения конструирования графического интерфейса пользователя. Многие современные среды разработки также включают *браузер классов*, *инспектор объектов* и *диаграмму иерархии классов* — для использования при объектно-ориентированной разработке ПО. Хотя, и существуют среды разработки, предназначенные для нескольких языков программирования — такие, как Eclipse, NetBeans, Embarcadero RAD Studio, Qt Creator или Microsoft Visual Studio, обычно среда разработки предназначается для одного определённого языка программирования - как, например, Visual Basic, Delphi, Dev-C++.  
  
**ИСТОРИЯ**  
  
История создания Visual Studio берет свое начало с 1997 года. **Visual Studio 97** — первая выпущенная версия Visual Studio, в ней впервые были собраны вместе различные средства разработки ПО. Она была выпущена в двух версиях Professional и Enterprise. Visual Studio 97 — была первой попыткой Microsoft создать единую среду для разработки на разных языках программирования: Visual C++, Visual J++, Visual InterDev, и MSDN использовали одну среду, называемую Developer Studio.   
  
Более поздние версии:

* **Visual Studio 6.0** — выпущена в июне 1998 — последняя версия Visual Studio, работающая на платформе Win9x. Данная версия являлась основной средой разработки приложений под Windows от Microsoft, до появления платформы .NET.
* **Visual Studio .NET** (кодовое имя *Rainier*; внутренняя версия 7.0) — выпущена в феврале 2002 года (включает .NET Framework 1.0). Service Pack 1 для Visual Studio .NET (2002) выпущен в марте 2005.
* **Visual Studio .NET 2003** (кодовое имя *Everett*; внутренняя версия 7.1) — выпущена в апреле 2003 года (включает .NET Framework 1.1). Service Pack 1 для Visual Studio .NET 2003 выпущен 13 сентября <2006.
* **Visual Studio 2005** (кодовое имя *Whidbey*; внутренняя версия 8.0) — выпущена в конце октября 2005 года, последняя официально работающая на Windows 2000, (включает .NET Framework 2.0). В начале ноября 2005 также вышла серия продуктов в редакции Express: **Visual C++ 2005 Express**, **Visual Basic 2005 Express**, **Visual C# 2005 Express** и др.
* **Visual Studio 2008** (кодовое имя *Orcas*; внутренняя версия 9.0) — выпущена 19 ноября 2007, одновременно с .NET Framework 3.5. Нацелена на создание приложений для ОС Windows Vista (но поддерживает и XP), Office 2007 и веб-приложений. Включает в себя LINQ, новые версии языков C# и Visual Basic. В студию не вошёл Visual J#. С 28 октября 2008 года впервые доступна версия на русском языке.
* **Visual Studio 2010** (кодовое имя *Hawaii*, для Ultimate — *Rosario*; внутренняя версия 10.0) — выпущена 12 апреля 2010 года вместе с .NET Framework 4.0. Visual Studio включает поддержку языков C# 4.0 и Visual Basic .NET 10.0, а также языка F#, отсутствовавшего в предыдущих версиях.

Рассмотрим наиболее позднюю версию - **Visual Studio 2010**. Она состоит из нескольких пакетов, которые, соответственно, различаются по возможностям и ценам:



Ниже указана таблица  различных пакетов Visual Studio, взятая с официального сайта.



На следующей таблице указаны цены и преимущества подписок:

Сноски:  
  
1. Может требовать одного или более Microsoft Visual Studio Load Test Virtual User Pack 2010  
  
2. Team Foundation Server и клиентская лицензия на Team Foundation Server включены в качестве льготы в некоторые подписки MSDN, но могут быть приобретены отдельно  
  
3. UML является зарегистрированной торговой маркой Object Management Group, Inc.  
  
4. Лицензия на пользователя позволяет неограниченно устанавливать и использовать для дизайна, разработки, тестирования и демонстрации возможностей приложений  
  
5. Требует Team Foundation Server и клиентскую лицензию Team Foundation Server, которые не включены в подписку MSDN Essentials  
  
6. Доступность зависит от страны и уровня подписки;  детали доступны на сайте MSDN; доступны владельцам подписок от ноября 2010 и более поздних  
  
7. Одна клиентская лицензия Team Foundation Server на подписку  
  
8. Доступность Feature Pack зависит от уровня подписки  
  
9. Соответствует Microsoft Expression Studio 4 Ultimate, но не включает SketchFlow  
  
10. Цены указаны в долларах США, без учета налогов и других сборов. Не является публичной офертой.

\* MSDN - Microsoft Developer Network (англ. *MSDN*) — подразделение компании Майкрософт, ответственное за взаимодействие фирмы с разработчиками. В данном случае, под разработчиками понимаются разработчики аппаратного обеспечения, интересующиеся операционной системой, а также разработчики, использующие программные интерфейсы операционной системы и скриптовые языки различных приложений, разработанных Microsoft. Такое взаимодействие с разработчиками имеет несколько форм: веб-сайты, новостные рассылки, конференции разработчиков, блоги, рассылка CD/DVD. Жизненный цикл взаимодействия с разработчиками варьируется от поддержки уже устаревших продуктов до распространения информации о новых возможностях.  
  
Подразделение работает как информационный сервис для разработчиков программного обеспечения. Основное внимание (в последнее время) уделяется платформе Microsoft .NET, но присутствуют и статьи, охватывающие такие области как практика программирования и шаблоны проектирования. Многие ресурсы доступны в онлайн, другие — только по почте на основе платной подписки. В зависимости от уровня подписки, подписчики могут получать ранние редакции новых версий операционных систем или других продуктов Майкрософт (приложения Microsoft Office, Visual Studio и др.). Некоторые университеты включены в программу MSDN Academic Alliance (MSDNAA), что позволяет студентам получать преимущества подписки MSDN.

Одним из главных недостатков Visual Studio является отсутствие кросс-платформенности продукта, а также абсолютная коммерциализованность продуктов (эти недостатки присущи практически всем продуктам Майкрософт). Курсовая работа была написана с помощью данной среды разработки. [2]

О программе

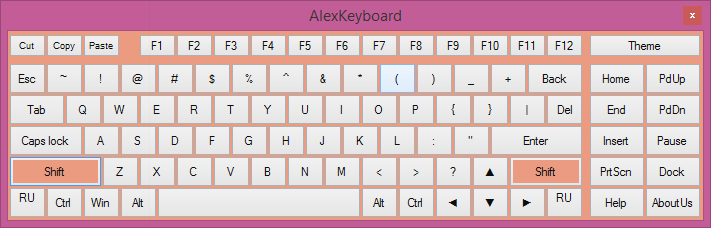
В этом пункте речь пойдет непосредственно о программе, написанной автором. На написания данной работы потребовалось достаточно много времени с учетом всех корректировок и изменений в самих модулях для увеличения функциональности и работоспособности программы.

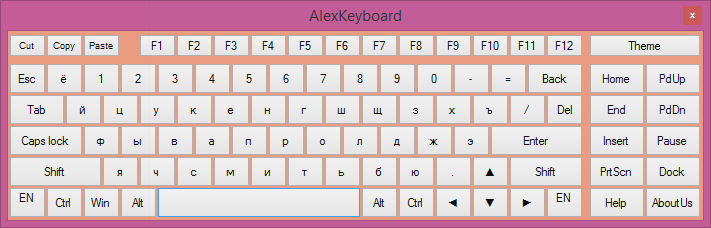
Проект состоит из двух основных модулей, речь о которых пойдет позже. В одном модуле происходит работа непосредственно с событиями через формы, в другом модуле установлен неуправляемый код, взятый мной из стандартной библиотеки “system32” для создания некоторых условий оптимальной работы программы. Ниже представлена релиз-версия проекта:

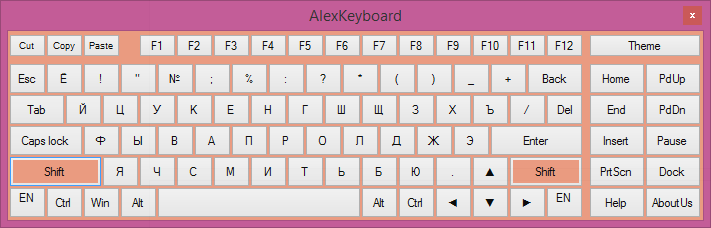


**Внешний дизайн**

Проект имеет возможность работы с основными командами системы. В следующих примерах продемонстрирована работа с разными языковыми пакетами:

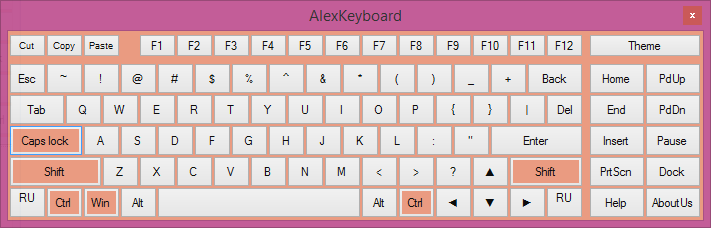






Кроме работы с языками и основными командами операционной системы Windows, приложение отслеживает нажатия клавиш “Shift” и “Caps Lock” для предотвращения непредвиденных ситуаций работы с основной клавиатурой компьютера. Программа была хорошо протестирована и отлично подойдет для среднестатистического пользователя (проверена на определенных личностях), но так же придется по вкусу и профессиональным пользователям, которые, к несчастью, не ознакомлены со стандартной экранной клавиатурой Windows.

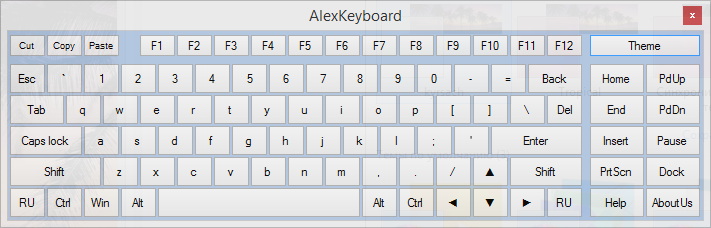
Помимо банальных функций типа “Ctrl+C”, “Ctrl+V” проект оснащен мощным инструментом для работы с комбинациями **сложностью** **до 4 клавиш**. В следующем скриншоте наглядно можно увидеть принцип работы этого метода:



Помимо всех стандартных функций, проект оснащен авторскими кнопками, выполняющими определенное действие для удобства, дизайна и простоты работы (чего, прошу заметить, не удосужились сделать создатели стандартной клавиатуры Windows). Примером являются бросающееся в глаза три кнопки в левом верхнем углу: “cut”, “copy”, “paste”. Название каждой клавиши говорит само за себя: это ничто иное, как упрощенный вариант стандартных комбинаций клавиш для быстроты и удобства работы в случае неисправности встроенной клавиатуры (лично мне не особо нравится тыкать мышкой) либо полном ее отсутствии.

Две кнопки “RUS” по нижним углам позволяют переключать язык, так же не задумываюсь о нажатии определенных комбинаций, но это не значит, что язык нельзя переключить другими методами, напротив: насчет этого я постарался.

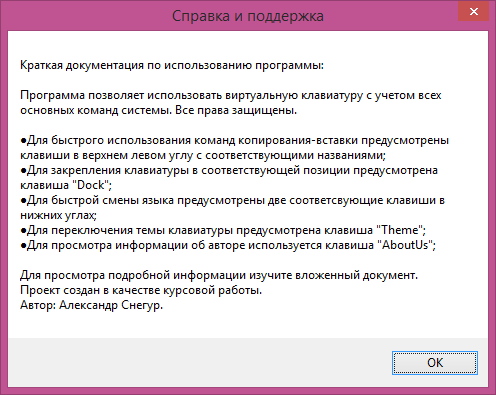
Кнопка “Theme” служит для изменения темы оформления приложения, позволяя проникнуться теплыми тропическими оттенками. Ниже приведен пример использования некоторых тем:



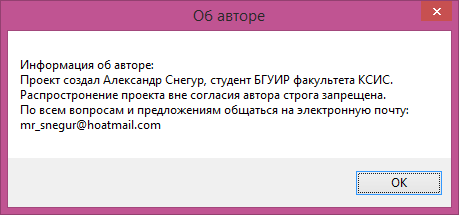




При создании приложения возникла необходимость создать возможность обратной связи и просмотр информации об авторе, а так же короткую справку об возможностях программы. Если приложение является активным окном и была нажата клавиша F1, либо пользователь нажал клавишу “Help”, появляется следующее окно:



При нажатии на клавишу “AboutUs” пользователь может ознакомиться с информацией, касающейся автора:



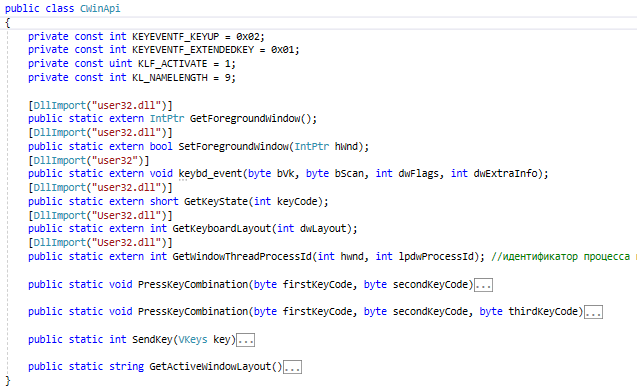
Так же стоит упомянуть о клавише “Dock”. Она служит для прикрепления клавиатуры в удобную для большинства пользователей позицию:



Графический интерфейс – очень важная вещь, которая является главным фактором пользовательского спроса на ту или иную программу, но не стоит забывать также, что то, что исходит извне программы – это и есть корень и причина разработки ПО, и, конечно же, его актуальность и спрос на продукт.

**О том, что внутри**

Как и было упомянуто ранее, проект состоит из двух основных модулей: модуль с формой и основными методами и модуль с библиотеками. Следующий скриншот показывает второй из них:

****

На данном скриншоте можно наблюдать подключение стандартных библиотек “user32.dll” с неуправляемым кодом. Два основных метода этой библиотеки, которые играют роль в проекте есть “keydb\_event” и “GetForegroundWindow”. Первый из этих методов позволяет передавать код клавиши таблицы ASCII в активное окно системы. Второй метод непосредственно получает это окно, чтобы вносить определенные изменения. Для предотвращения ситуаций типа невозможности перехода снова в окно после нажатия клавиши виртуальной клавиатуры используются другие методы, показанные в скриншоте, название которых говорит само за себя.

Далее приведены примеры использования некоторых написанных мною методов для работы с такими ситуациями, как нажатие нескольких клавиш одновременно, в том числе и с тремя клавишами, и, конечно же, нажатие и передача одной клавиши:

public static void PressKeyCombination(byte firstKeyCode, byte secondKeyCode)

{

keybd\_event(firstKeyCode, 0, 0, 0); // Ctrl Press

keybd\_event(secondKeyCode, 0, 0, 0); // C Press

keybd\_event(secondKeyCode, 0, KEYEVENTF\_KEYUP, 0); // C Release

keybd\_event(firstKeyCode, 0, KEYEVENTF\_KEYUP, 0);

}

public static void PressKeyCombination(byte firstKeyCode, byte secondKeyCode, byte thirdKeyCode)

{

keybd\_event(firstKeyCode, 0, 0, 0);

keybd\_event(secondKeyCode, 0, 0, 0);

keybd\_event(thirdKeyCode, 0, 0, 0);

keybd\_event(thirdKeyCode, 0, KEYEVENTF\_KEYUP, 0);

keybd\_event(secondKeyCode, 0, KEYEVENTF\_KEYUP, 0);

keybd\_event(firstKeyCode, 0, KEYEVENTF\_KEYUP, 0);

}

public static int SendKey(VKeys key)

{

keybd\_event((byte)(key), 0x45, KEYEVENTF\_EXTENDEDKEY, 0);

keybd\_event((byte)(key), 0x45, KEYEVENTF\_EXTENDEDKEY | KEYEVENTF\_KEYUP, 0);

return 0;

}

Для нормальной работы “keydb\_event” нам следует знать код каждой клавиши в таблице ASCII. Далее приведено перечисление со стандартными названиями кодов клавиш и их кодами соответственно:

public enum VKeys

{

VK\_LBUTTON = 1,

VK\_RBUTTON = 2,

VK\_CANCEL = 3,

VK\_MBUTTON = 4,

VK\_BACK = 8,

VK\_TAB = 9,

VK\_CLEAR = 12,

VK\_RETURN = 13,

VK\_SHIFT = 16,

VK\_CONTROL = 17,

VK\_MENU = 18,

VK\_PAUSE = 19,

VK\_CAPITAL = 20,

VK\_ESCAPE = 27,

VK\_SPACE = 32,

VK\_PRIOR = 33,

VK\_NEXT = 34,

VK\_END = 35,

VK\_HOME = 36,

VK\_LEFT = 37,

VK\_UP = 38,

VK\_RIGHT = 39,

VK\_DOWN = 40,

VK\_SELECT = 41,

VK\_PRINT = 42,

VK\_EXECUTE = 43,

VK\_SNAPSHOT = 44,

VK\_INSERT = 45,

VK\_DELETE = 46,

VK\_HELP = 47,

VK\_0 = 48,

VK\_1 = 49,

VK\_2 = 50,

VK\_3 = 51,

VK\_4 = 52,

VK\_5 = 53,

VK\_6 = 54,

VK\_7 = 55,

VK\_8 = 56,

VK\_9 = 57,

VK\_A = 65,

VK\_B = 66,

VK\_C = 67,

VK\_D = 68,

VK\_E = 69,

VK\_F = 70,

VK\_G = 71,

VK\_H = 72,

VK\_I = 73,

VK\_J = 74,

VK\_K = 75,

VK\_L = 76,

VK\_M = 77,

VK\_N = 78,

VK\_O = 79,

VK\_P = 80,

VK\_Q = 81,

VK\_R = 82,

VK\_S = 83,

VK\_T = 84,

VK\_U = 85,

VK\_V = 86,

VK\_W = 87,

VK\_X = 88,

VK\_Y = 89,

VK\_Z = 90,

VK\_NUMPAD0 = 96,

VK\_NUMPAD1 = 97,

VK\_NUMPAD2 = 98,

VK\_NUMPAD3 = 99,

VK\_NUMPAD4 = 100,

VK\_NUMPAD5 = 101,

VK\_NUMPAD6 = 102,

VK\_NUMPAD7 = 103,

VK\_NUMPAD8 = 104,

VK\_NUMPAD9 = 105,

VK\_SEPARATOR = 108,

VK\_SUBTRACT = 109,

VK\_DECIMAL = 110,

VK\_DIVIDE = 111,

VK\_F1 = 112,

VK\_F2 = 113,

VK\_F3 = 114,

VK\_F4 = 115,

VK\_F5 = 116,

VK\_F6 = 117,

VK\_F7 = 118,

VK\_F8 = 119,

VK\_F9 = 120,

VK\_F10 = 121,

VK\_F11 = 122,

VK\_F12 = 123,

VK\_SCROLL = 145,

VK\_LSHIFT = 160,

VK\_RSHIFT = 161,

VK\_LCONTROL = 162,

VK\_RCONTROL = 163,

VK\_LMENU = 164,

VK\_RMENU = 165,

VK\_PLAY = 250,

VK\_ZOOM = 251,

VK\_LWinKey = 91,

VK\_RWinKey = 92,

VK\_OEM\_MINUS = 0xBD,

VK\_OEM\_PLUS = 187,

VK\_OEM\_1 = 0xba,

VK\_Simple = 0xBB,

VK\_OEM\_COMMA = 0xbc,

VK\_OEM\_PERIOD = 0xbe,

VK\_OEM\_2 = 0xbf,

VK\_OEM3 = 0xc0,

VK\_OEM\_4 = 0xdb,

VK\_OEM\_5 = 0xdc,

VK\_OEM\_6 = 0xdd,

VK\_OEM\_7 = 0xde,

VK\_OEM\_8 = 0xdf,

VK\_NONE = 255,

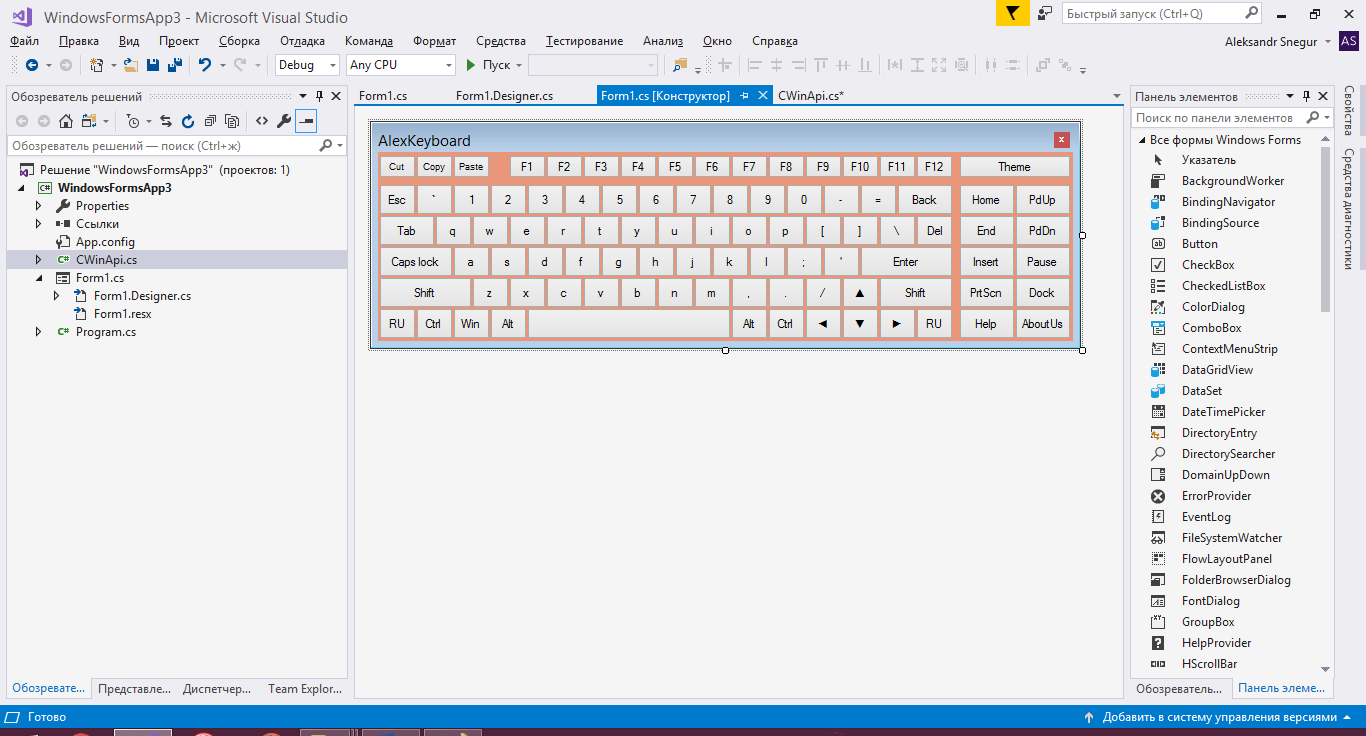
VK\_SQBRAKET = 0xDB,

VK\_QUEST = 0xBF,

};

Насколько можно видеть, тут имеются абсолютно все клавиши, которые я в последствие использовал в своей работе.

Что касается второго модуля, там приведены все основные методы для корректной работы программы и методы для создания качественного стильного интерфейса. Конструктор формы проекта имеет следующий вид:



Каждый элемент на форме есть кнопка, нажав на которую происходит событие передачи кода клавиши, которой соответствует клавиша, в систему:

private void Esc\_Click(object sender, EventArgs e)

{

Send(VKeys.VK\_ESCAPE);

}

private void button1\_Click(object sender, EventArgs e)

{

Send(VKeys.VK\_OEM3);

}

private void button2\_Click(object sender, EventArgs e)

{

Send(VKeys.VK\_1);

}

При нажатии на клавишу, которая может служить для создания команд, используется булевая переменная, отслеживающая нажатия той или иной клавиши:

private void Send(VKeys key)

{

CapsLockSynchronyze();

if (hWndTarget.ToInt32() != 0)

{

CWinApi.SetForegroundWindow(hWndTarget);

}

switch (key)

{

case VKeys.VK\_LSHIFT:

case VKeys.VK\_RSHIFT:

isShiftPressed = (isShiftPressed) ? false : true;

CWinApi.SendKey(key);

ShowPressedKey(button6, button45, isShiftPressed);

if (isShiftPressed && isEnLang)

ENBig();

else if (isShiftPressed && !isEnLang)

RUSBig();

else if (isEnLang)

ENLittle();

else

RUSLittle();

if ((isShiftPressed && isAltPressed)) //change language

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_LMENU, (byte)(key));

isEnLang = (isEnLang) ? false : true;

if (isEnLang)

ENLittle();

else

RUSLittle();

}

break;

case VKeys.VK\_SUBTRACT:

;

break;

case VKeys.VK\_LCONTROL:

case VKeys.VK\_RCONTROL:

isCtrlPressed = (isCtrlPressed) ? false : true;

CWinApi.SendKey(key);

ShowPressedKey(button65, button61, isCtrlPressed);

break;

case VKeys.VK\_LMENU:

case VKeys.VK\_RMENU:

isAltPressed = (isAltPressed) ? false : true;

CWinApi.SendKey(key);

ShowPressedKey(button64, button62, isAltPressed);

if ((isShiftPressed && isAltPressed))

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_LSHIFT, (byte)(key));

isEnLang = (isEnLang) ? false : true;

if (isEnLang)

ENLittle();

else

RUSLittle();

}

break;

case VKeys.VK\_CAPITAL:

isCapsLockPressed = (isCapsLockPressed) ? false : true;

CWinApi.SendKey(key);

ShowPressedKey(button5, isCapsLockPressed);

if (isCapsLockPressed && isEnLang)

ENBigCaps();

else if (isCapsLockPressed && !isEnLang)

RUSBigCaps();

else if (isEnLang)

ENLittle();

else

RUSLittle();

break;

case VKeys.VK\_LWinKey:

isWinPressed = (isWinPressed) ? false : true;

ShowPressedKey(button63, isWinPressed);

break;

default:

{

if ((isCtrlPressed & isAltPressed))

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_CONTROL, (byte)VKeys.VK\_MENU, (byte)(key));

}

else if ((isCtrlPressed & isShiftPressed))

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_CONTROL, (byte)VKeys.VK\_SHIFT, (byte)(key));

}

else if ((isAltPressed & isShiftPressed))

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_MENU, (byte)VKeys.VK\_SHIFT, (byte)(key));

}

else if (isShiftPressed)

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_SHIFT, (byte)(key));

}

else if (isCtrlPressed)

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_CONTROL, (byte)(key));

}

else if (isAltPressed)

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_MENU, (byte)(key));

}

else if (isWinPressed)

{

CWinApi.PressKeyCombination((byte)VKeys.VK\_LWinKey, (byte)(key));

if (key == VKeys.VK\_SPACE)

{

isEnLang = (isEnLang) ? false : true;

if (isEnLang)

ENLittle();

else

RUSLittle();

}

}

else

{

CWinApi.SendKey(key);

}

}

break;

}

System.Threading.Thread.Sleep(10);

}

Этот метод самый основной и самый сложный в проекте. Он учитывает нажатие системных клавиш, комбинаций, смену языка и возможность продемонстрировать нажатие системных клавиш подсветкой для удобного использования клавиатуры.

Следующий перегруженный метод отвечает за подсветку нажатых системных клавиш:

private void ShowPressedKey(Button buttonName, bool key)

{

if (key)

{

buttonName.BackColor = Color.DarkSalmon;

}

else

{

buttonName.BackColor = Color.Transparent;

}

}

private void ShowPressedKey(Button firstButtonName, Button secondButtonName, bool key)

{

if (key)

{

firstButtonName.BackColor = Color.DarkSalmon;

secondButtonName.BackColor = Color.DarkSalmon;

}

else

{

firstButtonName.BackColor = Color.Transparent;

secondButtonName.BackColor = Color.Transparent;

}

}

Стоит упомянуть, что проект так же сожержит возможность отслеживания нажатия каких-либо клавиш вне работы самого приложения, чтобы предотвратить неприятные случаи сбоя корректности программы. За это в проекте отвечает метод “Form1\_KeyDown\_1”. Метод следит за нажатием клавиш вне программы и вносит соответствующие изменения в клавиатуру, подсвечивая нужные клавиши автоматически:

private void Form1\_KeyDown\_1(object sender, KeyEventArgs e)

{

if (e.Shift)

{

isShiftPressed = (isShiftPressed) ? false : true;

ShowPressedKey(button6, button45, isShiftPressed);

if (isShiftPressed && isEnLang)

ENBig();

else if (isShiftPressed && !isEnLang)

RUSBig();

else if (isEnLang)

ENLittle();

else

RUSLittle();

}

else if (e.KeyCode == Keys.CapsLock)

{

isCapsLockPressed = (isCapsLockPressed) ? false : true;

ShowPressedKey(button5, isCapsLockPressed);

if (isCapsLockPressed && isEnLang)

ENBigCaps();

else if (isCapsLockPressed && !isEnLang)

RUSBigCaps();

else if (isEnLang)

ENLittle();

else

RUSLittle();

}

else if (e.KeyCode == Keys.F1)

HelpInfo();

}

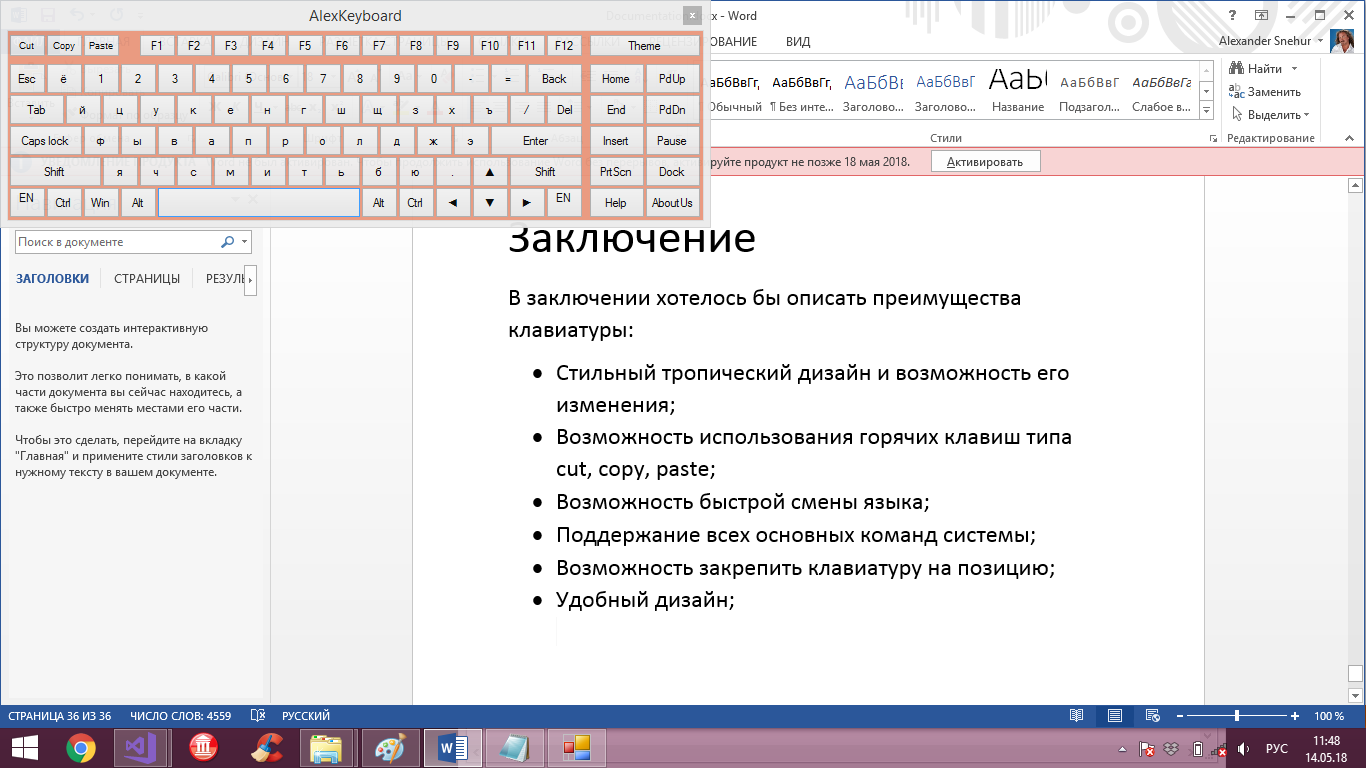
Конечно, тут перечислены только основные методы работы с программой, роспись каждого из них заняла бы очень много времени, поэтому стоит опустить такие примитивные методы, как смена языка, или вызов сообщения об авторе, или краткую документацию о проекте.

Следующая заключительная часть расскажет об особенностях и преимуществах данного проекта.

Заключение

В заключении хотелось бы описать преимущества клавиатуры:

* Стильный тропический дизайн и возможность его изменения;
* Возможность использования горячих клавиш типа cut, copy, paste;
* Возможность быстрой смены языка;
* Поддержание всех основных команд системы;
* Возможность закрепить клавиатуру на позицию;
* Удобный дизайн;



Написание данного документа путем использования проекта, 2018, фото в цвете.

Список литературы:

[1] - http://encicl.narod.ru/klav.htm

[2] - <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>