Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина: Программирование

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Клавиатурный тренажер «KlavaRush»

Выполнил: студент гр. 653501

Тупицын М.Н.

Руководитель: ассистент кафедры информатики Козуб В.Н.

Минск 2017

**Оглавление**

Введение [3](#_Toc482476563)

Анализ предметной области4

Постановка задачи4

Теоретическая часть4

Исследование аналогов4

Разработка программы6

Структура программы6

Логика работы приложения8

Скриншоты13

Заключение17

Список литературы:18

**Введение**

Не так давно появился Интернет. Изначально его использование было ограниченно лишь передачей информации между лабораториями, однако со временем его использование стало доступно массам. Тогда появилось огромное количество вариантов его использования, одним из которых было использование Интернета как способа общения. Появилась электронная почта. Со временем появлялись форумы, социальные сети, и многие другие способы коммуникации. А значит и большое количество печатного текста. Трудно найти современного человека, не умеющего пользоваться смартфоном или компьютером, однако далеко не каждый из них обладает хорошими навыками набора текстов, работы с клавиатурой. Люди предпочитают тыкать по клавиатуре 2 пальцами, выискивая глазами каждую букву перед тем, как ее нажать, и, зачастую, не осознают, как непозволительно много времени они тратят на это ежедневно. В профессиональных сферах, связанных с набором текстов, со временем появилось множество методик, нацеленных на обучение людей быстрому и эффективному набору. С развитием IT появилось специальные программы, являющиеся программным воплощением этих методик, которые получили название клавиатурных тренажеров.

Клавиатурные тренажеры предназначены для обучения людей быстрому набору путем использования при наборе не 2 пальцев, как ранее, а всех 10, с запоминанием расположения клавиш на клавиатуре и приучением пальцев нажимать клавиши, предназначенные именно для него. Это существенно ускоряет набор текста, а значит экономит огромное количество времени, которое можно потратить более эффективно, что особенно актуально для различных крупных предприятий с большим количеством печатной документации. Поэтому клавиатурные тренажеры становятся как никогда популярны и полезны.

**Анализ предметной области**

*Постановка задачи:*

Целью курсовой работы является создание собственного клавиатурного тренажера KlavaRush, позволяющему развить свои навыки работы с клавиатурой и увеличить скорость набора (в дальнейшем, в качестве дополнения к данному курсовому проекту, будет разработан специальный курс упражнений, позволяющий пользователю, не знакомому с методикой «слепого десятипальцевого метода набора», обучиться ему в рамках выполнения упражнений, представленных в данном курсе. Пробный набор данных упражнений был создан путем оптимизирования упражнений из аналога «СОЛО на Клавиатуре» и в данный момент находится на стадии тестирования. При успешном результате, этот курс упражнений с внесенными в него изменениями войдет в следующую версию клавиатурного тренажера KlavaRush). Для этого он должен отвечать следующим требованиям:

1. Иметь систему учета пользователей, позволяющую сохранять свой прогресс между входами в клавиатурный тренажер.
2. Иметь простой, приятный для визуального восприятия и дружелюбный (интуитивно понятный) интерфейс.
3. Иметь хорошо проработанную систему упражнений, позволяющих эффективно развивать свои навыки набора.
4. Наличие психологической поддержки пользователя, помогающей ему в обучении.

*Теоретическая часть:*

Для разработки данного приложения мною была выбрана среда разработки C++ Builder и его встроенная библиотека VCL. Причинами данного выбора стали:

1. Опыт работы в данной среде с данной библиотекой. Это позволило максимально реализовать задуманный функционал клавиатурного тренажера.
2. Наличие некоторых встроенных событий, позволяющих эффективно реализовывать некоторые части программы.

*Исследование аналогов:*

Исследование аналогов на данный момент дало неоднозначные результаты. Большинство аналогов оказались откровенно неудачными проектами, не дающими никаких практических результатов их пользователям. Их функционал зачастую максимально ограничен, что сильно сказывается даже на теоретической возможности развить свои навыки. Существуют и более удачные аналоги, превосходящие KlavaRush по функционалу на данный момент и имеющие некоторые другие удачные решения, однако такие проекты в своем большинстве оказались полностью лишенными поддержки со стороны разработчика и не учитывают пожеланий своих пользователей. Основной упор в KlavaRush планируется сделать именно на взаимодействие с пользователями, на удовлетворении их пожеланий в графическом и функциональном наполнении программы, а также на качественно проработанную систему упражнений. Так как приложение на данный момент имеет скудное количество, оно отстает от некоторых своих аналогов, но по прошествии некоторого периода времени его предполагаемый уровень развития выше, чем у конкурентов.

**Разработка программы**

**Структура программы**

Исходный код программы содержит 7 форм и 4 файла, хранящих структуры данных, использованных в программе. Также к проекту прилагаются 22 файла, содержащих текст доступных на данный момент в программе упражнений и файл, хранящий данные зарегистрированных пользователей. С текстами упражнений можно ознакомится непосредственно в самой программе. Далее опишу структуры данных, используемые в приложении.

Файл Node содержит информацию об элементе списка пользователей, в частности указатель на данные самого пользователя и указатель на следующий элемент списка.

Файл List содержит в себе описание структуры списка, состоящего из узлов, описанных в файле Node, а также описание методов работы с этим списком.

Файл User содержит описание класса пользователя, хранящего в себе данные конкретного пользователя и методы, необходимые для работы с этими данными.

Среди прочих в списке данных пользователя хранится структура типа Stats, описанная в одноименном файле. В ней хранятся данные, касающиеся статистики данного пользователя.

Далее следует описать формы, используемые в приложении. Форма WelcomePageForm представляет собой стартовой окно программы. Она содержит в себе 3 кнопки компонента: кнопку начала работы с клавиатурным тренажером, кнопку, отображающую информацию о разработчике и текущей версии программы, и кнопку выхода из программы. После нажатия клавиши, начинающей работу с тренажером, управление переходит к следующей форме, осуществляющей авторизацию пользователя в программе. Эта форма имеет название LoginForm. На ней предоставляется возможность ввести свои данные для последующего перехода к своему личному кабинету, или же зарегистрироваться, если пользователь не делал этого ранее. При регистрации вызывается форма RegistrationForm. Здесь пользователь вводит данные, необходимые для регистрации. После успешной регистрации пользователя возвращает к LoginForm для осуществления авторизации в программе.

Следующая форма, которую видит пользователь – PersonalForm. Здесь находятся 4 кнопки: кнопка, позволяющая перейти к выбору упражнения (она вызывает ExerciseSelectForm), кнопка, позволяющая пользователю посмотреть свою статистику, кнопка, позволяющая пользователю перейти к настройкам, и кнопка, осуществляющая выход из текущего профиля, возвращающая пользователя к окну авторизации.

SettingsForm, упомянутая выше, позволяет пользователю переключить язык, изменив тем самым язык раскладки клавиатуры, для которой будет проводится тренировка, а также включить или отключить музыку (не работает в текущей версии) и содержит кнопку, позволяющую удалить свой профиль. После удаления своего профиля пользователя вернет к окну авторизации.

ExerciseSelectForm содержит список упражнения, доступных на данный момент. Он позволяет выбрать одно из 10 тренировочным заданий (для каждой представленной в программе раскладки), финальный тест (по сути, крупному упражнению, представляющему собой текст, позволяющий проверить свои навыки набора связного незнакомого текста) или режим случайной генерации, генерирующий случайный набор символов для набора (одно из главных преимуществ проекта; позволяет значительно отточить свои навыки набора на интуитивном уровне). После выбора упражнения управление передастся последней форме, используемой в проекте, ExerciseForm, в которой, непосредственно, и будет происходить выполнение выбранного упражнения.

**Логика работы приложения**

Логика работы приложения по сути представляет собой линейный алгоритм, поэтому описывать ее будет удобно в том порядке, в котором с ней будет знакомится новый пользователь.

Итак, при запуске программы на экране появляется WelcomeForm. На ней находятся три клавиши: «Начать тренировку», «Информация» и «Выход». Нажатие каждой из них вызовет событие OnClick, относящееся к данной кнопке. При нажатии кнопки «Начать тренировку» будет спрятана форма WelcomeForm и показана форма LoginForm. WelcomeForm прячется, а не закрывается, так как ее закрытие повлечет за собой закрытие всего приложения в целом. При нажатии кнопки «Информация» будет вызвана функция ShowMessage(), которая отобразит на экран небольшое окошко с сообщением об разработчике и текущей версии программы. Нажатие кнопки «Выход» повлечет за собой закрытие формы WelcomeForm, и, как следствие, прекратит работу приложения.

Переходим к форме LoginForm. При ее создании вызывается событие FormCreate. В ходе этого события открывается поток ввода, и из файла users.sav, хранящего информацию о пользователях, в свежесозданный экземпляр класса List UserList переписывается информация о зарегистрированных на данный момент пользователях. После этого поток ввода закрывается, и вся дальнейшая работа с пользователями выполняется с использованием этого списка. Форма содержит 2 компонента типа TEdit с именами NicknameEdit и PasswordEdit, предназначенные для ввода логина и пароля соответственно, кнопку «Войти», 2 компонента типа TLabel, поясняющий, что необходимо вводить в компоненты типа TEdit, компонент типа TLabel RegisterLabel, изменяющий свой цвет при наведении мышки (служит для регистрации), и компонент типа TGroupBox GroupBox, необходимый для лучшего визуального восприятие остальных компонентов. Нажатие клавиши «Войти» вызовет событие OnClick, в ходе которого будет метод класса List Check(), принимающий в качестве аргументов содержимое компонентов NicknameEdit->Text и PasswordEdit->Text (метод Check() класса List перегружен). Далее будет выполнен поиск пользователя с такой комбинацией логина и пароля в списке UserList. В случае успеха метод вернет указатель на позицию этого пользователя в списке UserList, и его значение будет присвоено указателю типа User \*CurrentUser. Далее будет выведено одно из 5 приветственных сообщений, выбранное случайным образом, с упоминанием имени пользователя (для его большего психологического комфорта). После этого будет спрятана форма LoginForm и отображена форма PersonalForm, которая будет рассмотрена подробнее чуть позже. Если же поиск пользователя в списке не дал результатов, с помощью функции ShowMessage() будет выведено сообщение об этом. Поле пароля будет очищено. Отдельно стоит упомянуть, что символы, вводимые в поле, содержащее пароль, заменяются на заранее предустановленный символ, что позволяет защитить ваш пароль от «чужих глаз» при его вводе, что делается при помощи свойства PasswordChar компонента PasswordEdit. Если пользователь не зарегистрирован, в этом ему поможет нажатие на компонент типа TLabel RegisterLabel, о чем ему услужливо сообщит текст, содержащийся в самом компоненте. После этого будет вызвана форма RegisterForm, а форма LoginForm спрятана. Закрытие же LoginForm запустит процесс выгрузки пользователей из UserList в файл Users.sav (это произойдет при нажатии крестика в правом верхнем углу формы). Это является уязвимостью программы, так как прогресс текущей сессии программы сохранится только при нормальном ее завершении, если же она «вылетит», прогресс сохранен не будет. В данный момент ищутся способы решения этой проблемы.

RegisterForm содержит 4 компонента типа TEdit, служащие для ввода имени, логина, и паролей (пароль вводиться 2 раза, чтобы избежать случайных опечаток в нем), 4 компонентов типа TLabel, содержащих подписи к вышеупомянутым компонентам, кнопки «Регистрация» и компонента типа TGroupBox, необходимого для лучшего визуального восприятия всех вышеперечисленных. При нажатии клавиши «Регистрация» произойдет следующее: программа проверит, все ли поля заполнены, и выведет сообщение если это не так. Если же все поля заполнены, то будет вызвать метод Check() у экземпляра класса List UserList, принимающий в качестве аргумента логин пользователя, указанный в компоненте NicknameEdit. Если такой логин уже встречается в списке пользователей, регистрация не будет осуществлена и будет выведено соответствующее сообщение. Если же выбранный логин свободен, то будет проведена проверка совпадения введенных паролей. Если они не совпадут, программа выдаст соответствующее сообщение, и очистит содержимое полей, содержащих пароли. Если пароли совпадут, будет произведена регистрация пользователя. Для этого будет создан указатель на новый экземпляр класса User \*NewUser. После этого с помощью методов класса User, таких как SetName(), SetNickname() и аналогичных, которые примут в качестве параметров данные, введенные в компонентах, будет произведено заполнение информации о данном пользователе. Затем, с помощью метода Add() класса List, принимающего в качестве параметра указатель на экземпляр типа User, будет произведено добавление нового пользователя в List UserList, выведено сообщение об успешной регистрации, а форма RegisterForm будет закрыта. При закрытии формы будет отображена форма LoginForm.

PersonalForm представляет собой личный кабинет пользователя. Здесь находятся 4 кнопки: «Упражнения» вызовет форму ExerciseSelectForm, «Статистика» выведет на экран статистику пользователя (использует User \*CurrentUser для этого), содержащую общее количество ошибок за все время, общее число выполненных упражнений и общее затраченное на них время (в разработке), «Настройки» вызовет форму SettingsForm, «Выход» закроет PersonalForm и вернет управление форме LoginForm.

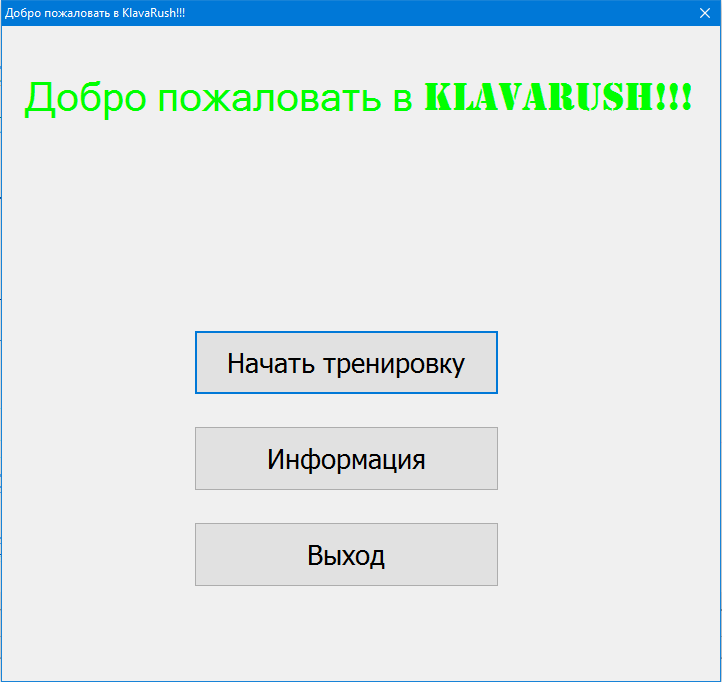
SettingsForm содержит 3 компонента: RadioGroup типа TRadioGroup, содержащий переключатель языка раскладки (на данный момент доступны английская и русская раскладки), MusicCheckBox типа TCheckBox, позволяющий включить или отключить музыку (не работает в данной версии), и кнопку «Удалить профиль». При нажатии кнопки будет вызван метод Delete() класса List, который примет в качестве аргумента логин пользователя и удалит его из списка пользователей, после чего выведет сообщение, выражающее сожаление о потере пользователя (психологический трюк), после этого управление будет передано форме LoginForm. Если закрыть форму через крестик в правом верхнем углу, то управление будет передано форме PersonalForm.

Перейдем к ExerciseSelectForm. Здесь располагается 3 компонента: 2 кнопки «Начать упражнение» и «Назад» и компонент ExerciseSelectListBox типа TListBox. В компоненте ExerciseSelectListBox содержится название всех доступных упражнений. Для перехода к упражнению пользователю необходимо выделить желаемое упражнение и нажать кнопку «Начать упражнение». При нажатии кнопки выполнится седующая последовательность действий: если не было выбрано ни одно упражнение, программа выведет об этом сообщение (используя ShowMessage()). Если же было выбрано одно из 10 упражнений (не включая случайную генерацию и финальный тест), программа проверит, есть ли у пользователя доступ к этому упражнению (для этого используется User \*CurrentUser). При отсутствии доступа программа выведет соответствующее сообщение, иначе начнется генерация строки FileName (используется выбранное упражнение ExerciseSelectListBox->ItemIndex), в которой после окончания генерации (она продолжиться в форме ExerciseForm) будет содержаться имя файла, в котором содержится упражнение, выбранное пользователем. После чего управление будет передано ExerciseForm. При закрытии формы ExerciseSelectForm с помощью крестика или нажатия кнопки «Назад» управление будет передано форме PersonalForm.

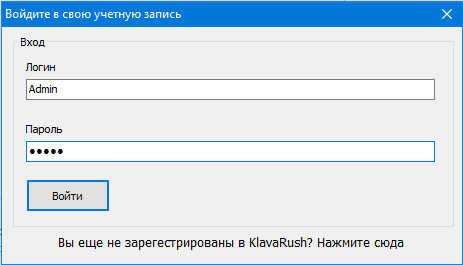
ExerciseForm содержит следующие компоненты: 2 компонента типа TLabel, предназначение которых состоит в том, чтобы отображать количество допущенных пользователем ошибок при выполнении текущего упражнения, компонент RichEdit типа TRichEdit, отображающий текст текущего упражнения, компонент EnterEdit типа TEdit, в который осуществляется ввод, и 48 компонентов типа TPanel, имеющих свои индивидуальные названия, отображающие подсказку расположения клавиш на клавиатуре (в «слепом десятипальцевом методе» нельзя смотреть на реальную клавиатуру для обучения, но можно смотреть на виртуальную подсказку на экране). Форма работает следующим образом. При создании формы заканчивается генерация строки FileName, содержащей имя необходимого файла. Если были выбраны одно из 10 упражнений или финальный тест, то в компонент RichEdit загружается текст, содержащийся в файле с именем FileName (RichEdit->Lines->LoadFromFile(FileName)). Если был выбран режим случайной генерации, она начинается именно в этот момент. Генерируется случайное число Lines (в промежутке от 10 до 40), далее, генерируется [Lines] строк, в каждой из которых находится случайное CharsAmount (в промежутке от 20 до 60) символов. Сгенерированные строки добавляются в компонент RichEdit. После этого идет инициализация виртуальной клавиатуры, где тексту на каждой из 48 панелей присваивается свое значение так, чтобы сформировалась виртуальная клавиатура-подсказка с учетом выбранного пользователем языка раскладки. На этом действие события FormShow заканчивается. Далее, при нажатии клавиши, срабатывает событие OnKeyPress компонента EnterEdit. Оно проверяет, совпадает ли набранный символ с тем, который было необходимо набрать (для этого использется переменная PosCounter). Если символ набран правильно, он заливается контрастным цветом в компоненте RichEdit, иначе программа выдаст случайное из 5 сообщений об ошибке (с поддержкой пользователя в них, психологический трюк), вернет цвет текста в компоненте RichEdit на начальный (черный), заставляет набрать всю строку сначала, не забыл увеличить счетчик допущенных ошибок, что сразу отразиться в компоненте типа TLabel AmountOfMistakesLabel. Так программа будет продолжать свою работу либо до завершения задания, либо до закрытия окна. В первом случае будет выведено одно из 5 сообщение с похвалой (психологический трюк), пользователю будет открыто следующее упражнение (если оно было закрыто), после чего у формы вызовется событие OnClose. При закрытии окна событие OnClose вызовется сразу. В ходе этого события пользователю в статистику добавится количество сделанных ошибок, компоненты RichEdit и EnterEdit будут очищены, а управление передастся форме ExerciseSelectForm.

**Скриншоты**

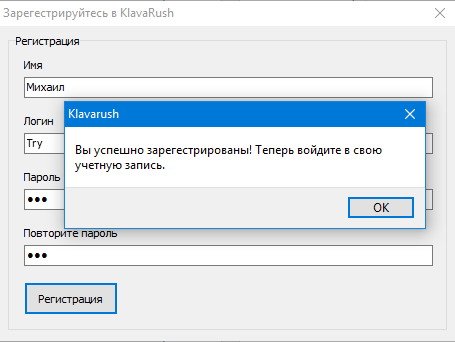
Первое, что пользователь видит при запуске – WelcomeForm:



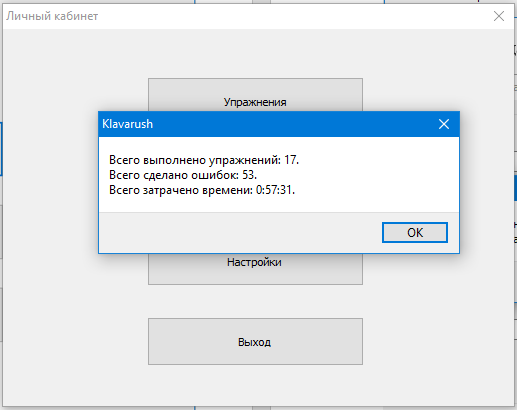
Следующая форма – LoginForm:



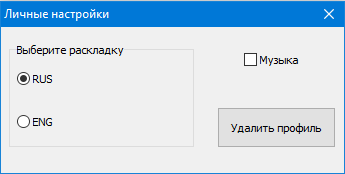
Пример удачной регистрации в RegistrationForm:



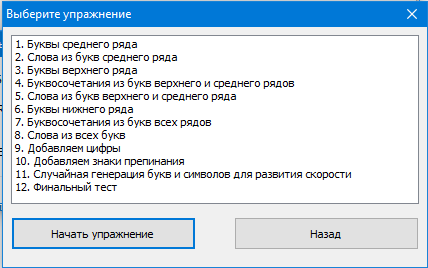
PersonalForm с выведенной статистикой:



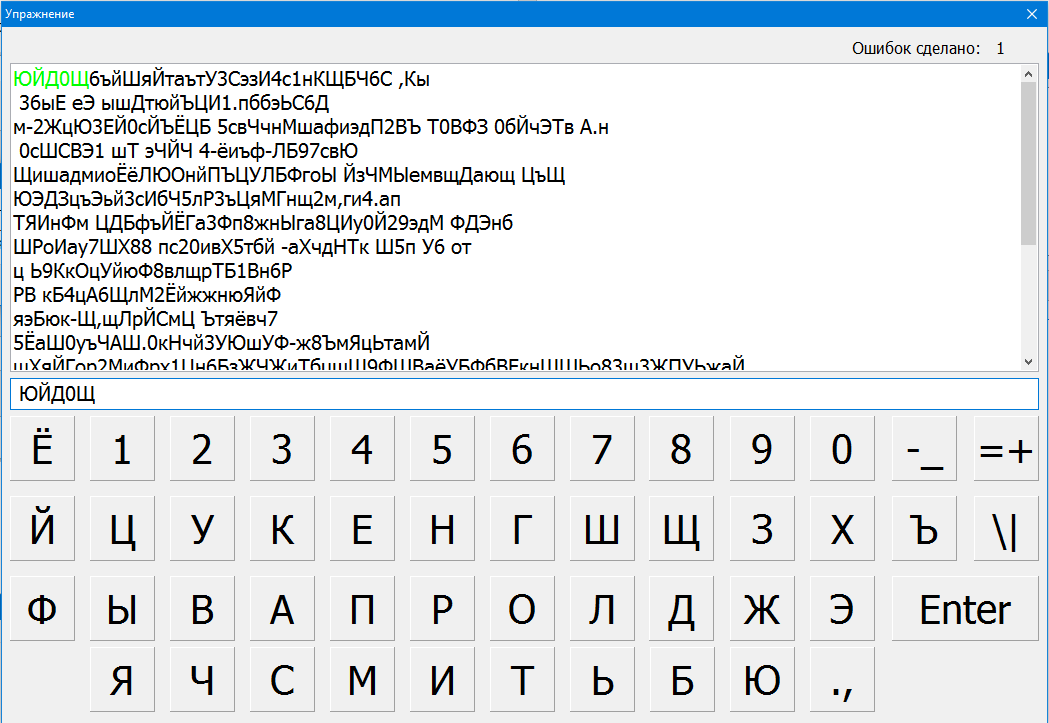
SettingsForm:



ExerciseSelectForm:



ExerciseForm со случайной генерацией с использованием русской раскладки:



**Заключение**

В ходе выполнения курсовой работы были выполнены следующий задачи:

- Был разработан клавиатурный тренажер KlavaRush, в ходе создания которого я учел свои пожелания по проектированию клавиатурного тренажера, а также прислушался к мнению немногочисленных пользователей.

- Была начата разработка собственной методики обучения «слепому десятипальцевому методу» набора текста.

- Был разработан план по дальнейшему улучшению и доработке готового приложения.

**Список литературы**

1. Шилдт, Герберт. Полный справочник по C++, 4-е издание. . Пер. с англ. — М. : Издательский дом “Вильямс”, 2006. — 800 с. : ил. — Парал. тит. англ.
2. Жвакина, А. В. Разработка Windows-приложений в среде визуального программирования: Электронное учебное издание (Пособие). / А. В. Жвакина. – Минск: ВА РБ, 2016.