Частное учреждение образования

«Колледж бизнеса и права»

СОЗДАНИЕ ПРОГРАММНОГО СРЕДСТВА ПО УЧЕБНОМУ ПРЕДМЕТУ ИНСТРУМЕНТАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ИЗУЧЕНИЯ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА И РЕШЕНИЯ БАЗОВЫХ ЗАДАЧ С ВОЗМОЖНОСТЬЮ ПРОВЕРКИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Основы алгоритмизации и программирования»

КП Т.295018.401

Руководитель проекта (Дж. А. Миронова)

Учащийся (М. А. Симанович)

2024

Изм.

Лист

№ докум.

Подпись

Датаа

Лист

3

КП Т.295018.401 ПЗ

Разраб.

Симанович М.А.

Провер.

Миронова Дж.А.

Реценз.

Н. Контр.

Утверд.

Создание программного средства по учебному предмету «Инструментальное программное обеспечение» для изучения теоретического материала и решения базовых задач с возможностью проверки

Лит.

Листов

КБиП

50

У

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc170314480)

[1 Описание задачи 5](#_Toc170314481)

[1.1 Анализ предметной области 5](#_Toc170314482)

[1.2 Постановка задачи 6](#_Toc170314483)

[2 Проектирование системы 7](#_Toc170314484)

[2.1 Требования к приложению 7](#_Toc170314485)

[2.2 Проектирование модели 7](#_Toc170314486)

[2.3 Концептуальный прототип 8](#_Toc170314487)

[3 Описание реализации программного средства 11](#_Toc170314488)

[3.1 Инструменты разработки и применяемые технологии 11](#_Toc170314489)

[3.2 Функции: логическая и физическая организация 11](#_Toc170314490)

[3.3 Входные и выходные данные 15](#_Toc170314491)

[3.4 Функциональное тестирование 15](#_Toc170314492)

[3.5 Описание справочной системы 17](#_Toc170314493)

[4 Применение 18](#_Toc170314494)

[4.1 Назначение программного средства 18](#_Toc170314495)

[4.2 Условия применения 18](#_Toc170314496)

[Заключение 19](#_Toc170314497)

[Список использованных источников 20](#_Toc170314498)

[Приложение А 21](#_Toc170314499)

[Приложение Б 48](#_Toc170314500)

Введение

В современном мире трудно представить жизнь без IT-технологий. В этой цифровой эпохе использование программного обеспечения становится неотъемлемой частью нашей реальности. Все ищут способы попасть в IT, выучить язык программирования и вложить свой вклад в развитие технологий. Это инструмент, который облегчает обучение, улучшает процессы обучения и управления информацией, а также способствует развитию профессиональных навыков.

Важность создания программного средства трудно переоценить. Этот инструмент не только обеспечит удобное управление учебным материалом и выполнение практических заданий, но и предоставит пользователю возможность эффективной работы с информацией и обмена опытом с коллегами.

Целью разработки курсового проекта на тему «Создание программного средства по учебному предмету Инструментальное программное обеспечение» является реализация инструмента, который позволит студентам учиться эффективно, проверять свои знания и обмениваться опытом с коллегами, благодаря удобному интерфейсу, возможности выполнять практические задания, формированию отчетов, изучению и пометок в документации.

Актуальность программного средства заключается в том, что оно отвечает современным требованиям к образованию и управлению информацией, предоставляя пользователю инновационные инструменты для обучения и развития профессиональных навыков.

В первом разделе «Описание задачи» основное внимание уделяется выявлению ключевых задач, необходимых для успешной реализации проекта. Это включает в себя выявление основных функциональных требований к программному средству, а также план разработки.

Во втором разделе «Проектирование системы» представлено описание структуры программного средства, включая создание лаконичного интерфейса с использованием меню, кнопочных форм и панелей инструментов, а также организацию иерархии классов и работы с информацией.

В третьем разделе «Реализация программного средства» затронуты технические аспекты создания программного средства, включая выбор используемых технологий, алгоритмы и архитектуру приложения, а также реализацию всех функций и возможностей, описанных в задании.

Четвертый раздел «Применение» посвящен обзору возможностей и сфер применения разработанного программного средства, а также его влиянию на повседневную деятельность пользователей, обучение и управление информацией.

В Заключении проведен анализ результатов разработки, выделены основные достижения и определены перспективы дальнейшего развития проекта.

1 Описание задачи

1.1 Анализ предметной области

Целью разработки курсового проектирования на тему «Создание программного средства по учебному предмету Инструментальное программное обеспечение» является реализация программного средства, которое предоставляет пользователям возможность входа в личный аккаунт или создания нового, просмотра доступных учебных материалов, регистрации в приложении, добавления материалов в избранное и получения краткого описания учебных курсов.

Предметная область языка программирования Python охватывает широкий спектр применений, что делает его одним из самых популярных языков в мире. Python используется во множестве сфер, например В области машинного обучения и искусственного интеллекта Python занимает центральное место, поддерживая такие инструменты как TensorFlow, Keras и PyTorch для разработки и обучения моделей машинного обучения. Python также используется для автоматизации задач и написания скриптов, облегчая системное администрирование и рутинные задачи. В разработке игр Python используется с библиотекой Pygame для создания 2D игр и прототипирования игровых концепций и многое другое.

Компьютерная обработка значительно упрощает процесс доступа к учебным материалам по сравнению с реальным визитом. Она повышает спрос пользователей и их заинтересованность, обеспечивает быстрый доступ к доступным курсам и их стоимости.

Потенциальными пользователями программного средства являются студенты, поскольку учебные курсы востребованы среди молодежи. Программное средство разработано с целью объединения всех необходимых средств для изучения языка программирования.

1.2 Постановка задачи

Задачей проекта является разработка программного средства для просмотра и прохождения учебных курсов. Программа обеспечивает просмотр и ознакомление с курсами, их прохождения, просмотр видеороликов и прохождения практических заданий.

Программное средство будет включать в себя функционал для поиска по названию курса, что позволяет пользователям ориентироваться в большом объеме данных. Также будет возможность сохранения и просмотра пройденных курсов.

На рисунке 1.1 представлен аналог разрабатываемого программного средства «Stepik».

Рисунок 1.1 «Stepik»

Достоинства аналога:

1. Интерфейс: Простой и интуитивно понятный интерфейс облегчает навигацию и использование приложения.

2. Доступность: Возможность регистрации в приложении в любое время и из любого места, используя персональный компьютер

3. История прохождения курсов и отчеты: Хранение истории пройденных курсов, возможность создания отчетов

Недостатки аналога:

1. Зависимость от интернета: Необходимость постоянного подключения к интернету может ограничивать использование приложения в условиях слабого или отсутствующего интернета.

2. Технические проблемы: Возможные ошибки и сбои в работе приложения могут вызвать неудобства и снизить доверие пользователей.

3. Актуальность информации: Необходимость постоянного обновления информации о доступных курсах и их расписании может потребовать значительных усилий от организаторов.

Разрабатываемое программное средство будет включать в себя различные особенности, позволяющие более комфортно взаимодействовать с приложением.

2 Проектирование системы

2.1 Требования к приложению

Безопасность в приложении обеспечена на высоком уровне односторонним шифрованием в хэш и записью в бинарные файлы (.bin). В такие файлы записываются логин, пароль, личные данные пользователя из полей. В приложении реализована авторизация и регистрация, она локальная, потому что приложение поддерживается никакими онлайн серверами и является полностью локальным, что еще больше минимизирует риски взломов.

Интерфейс лаконичен и интуитивно понятен, он не требует от пользователя специальных технических знаний; все элементы управления расположены логично и последовательно. Реализована система вкладок в основной форме, чтобы не нагружать пользователя новыми дополнительными окнами.

Для удобства работы пользователей размеры кнопок и полей ввода, используется современный, читаемый шрифт, Century Gothic размером от 8 до 25 пунктов. Цветовая схема интерфейса темно-синего оттенка, что уменьшает нагрузку на глаза при длительной работе. Применяются сторонние элементы управления из специальных фреймворков, такие как закругленные, анимированные кнопки, выпадающие закругленные списки и закругленные поля ввода, соответствующие принципам удобства использования и красоты.

Минимальные требования к аппаратным и операционным ресурсам:

* OC: Windows 10
* Процессор: Pentium® 4 1.5 GHz / Athlon® XP
* Оперативная память: 2048 MB ОЗУ
* Видеокарта: Встроенная графика или любая дискретная видеокарта.

2.2 Проектирование модели

Для отображения функциональной структуры объектов программного средства необходимо воспользоваться унифицированным языком моделирования UML, а именно диаграмму классов и диаграмму работы алгоритмов.

Диаграмма работы алгоритмов — это структурная диаграмма в языке UML является графическим представлением последовательности шагов выполнения алгоритма. Она помогает визуализировать логику алгоритма и его структуру, делая процесс понятным и легко читаемым. Данная диаграмма полезна при проектировании, отладке и понимании алгоритмов. Они помогают программистам и разработчикам лучше организовать свои мысли, выделить ключевые шаги и легко обнаружить ошибки или улучшить производительность алгоритма. Диаграмма работы алгоритмов представлена в графической части на листе 1.

Диаграмма классов — это структурная диаграмма в языке UML, которая отображает структуру системы, описывая классы, их атрибуты, методы и отношения между ними. Диаграмма классов помогает визуализировать архитектуру программного обеспечения и отношения между объектами. Данная диаграмма помогает разработчикам лучше понять структуру системы, улучшить проектирование программного обеспечения и облегчить коммуникацию между участниками проекта. Они являются важным инструментом для проектирования и документирования программных систем. Диаграмма классов представлена в графической части на листе 2.

2.3 Концептуальный прототип

Концептуальный прототип – это общее видение продукта, его функций, логики взаимодействия с другими компонентами. Цель концептуального прототипа — получить обратную связь от пользователей и заинтересованных сторон на ранних этапах разработки, что позволяет уточнить требования и улучшить дизайн до начала полноценной разработки.

Окно авторизации включает в себя список кнопок, которые выполняют различные действия. Графическое представление прототипа окна авторизации представлено на рисунке 2.1.

Рисунок 2.1 – прототип окна авторизации

Кнопка регистрация оставляет пользователя на этой форме, добавляет кнопку зарегистрироваться и textBox для подтверждения пароля. Прототип представлен на рисунке 2.2.

Рисунок 2.2 – окно регистрации

Кнопка вход перебрасывает пользователя на рабочее окно приложения. Прототип рабочего окна приложения представлен на рисунке 2.3.

Рисунок 2.3 – прототип рабочего окна приложения

После нажатия кнопки регистрации идет перенаправление на окно заполнения профиля. Где он должен подтвердить свой email и указать личные данные. Прототип предоставлен на рисунке 2.4

Рисунок 2.4 – прототип окна заполнения профиля

3 Описание реализации программного средства

3.1 Инструменты разработки и применяемые технологии

Разработка приложения на С++ с использованием фреймворков Guna2UI, SiticoneDesktop и BunifuFramework в среде Visual Studio представляет собой процесс, направленный на создание высокопроизводительного клиентского приложения с современным и эстетичным пользовательским интерфейсом.

Язык программирования C++ выбран из-за его высокой производительности и возможности эффективного управления ресурсами, что особенно важно для приложений, взаимодействующих с данными из интернета, такими как данные о криптовалютах. Visual Studio, в качестве интегрированной среды разработки, обеспечивает удобную платформу для создания, отладки и развертывания приложений на C++. Она интегрирует в себя все необходимые инструменты и ресурсы для эффективной работы над проектами.

Фреймворки Guna2UI, SiticoneDesktop и BunifuFramework предоставляют разнообразные графические компоненты и элементы управления, которые позволяют создавать современные пользовательские интерфейсы. Они предоставляют не только стандартные элементы WinForms, но и расширенные возможности для стилизации и анимации интерфейса, что значительно повышает его привлекательность и удобство использования для конечного пользователя.

Такой подход позволяет разработать полноценное обучающее приложение, способное эффективно обрабатывать код, предоставляя пользователю удобный и современный интерфейс для взаимодействия с документацией, задачами и другими данными.

3.2 Функции: логическая и физическая организация

Реализация системы регистрации. Эта функция выполняется на стадии регистрации логина и пароля, она проверяет поля на корректность, в случае если какое-либо поле некорректно, она завершает свою работу. В ином случае функция вызывает другую, которая записывает в файл данные.

Void auth::buttonRegister\_Click(Object^ sender, EventArgs^ e) {

// Проверка на пустые поля

if (textBoxLogin1->Text == "") {

textBoxLogin1->BorderColor = Color::Red;

MessageWarning->Show("Введите логин");

return;

}

if (textBoxPassword1->Text == "" || textBoxPassword\_1->Text == "") {

textBoxPassword1->BorderColor = Color::Red;

textBoxPassword\_1->BorderColor = Color::Red;

MessageWarning->Show("Введите пароль");

return;

}

// Проверка на совпадение паролей

if (textBoxPassword1->Text != textBoxPassword\_1->Text) {

textBoxPassword1->BorderColor = Color::Red;

textBoxPassword\_1->BorderColor = Color::Red;

MessageError->Show("Пароли не совпадают", "Ошибка");

return;

}

// Проверка на соглашение

if (checkBoxTerms->Checked == false) {

MessageError->Caption = "Предупреждение";

MessageError->Text = "Для регистрации вам нужно прочесть условия использования";

MessageError->Show();

return;

}

// Проверка на существование уже такого пользователя

if (Directory::Exists(textBoxLogin1->Text)) {

MessageError->Caption = "Предупреждение";

MessageError->Text = "Пользователь под таким логином уже существует";

MessageError->Show();

return;

}

// Регистрация в случае если все условия соблюдены

panelMain->Visible = false;

Register();

}

Реализация функции Register(). Эта функция сохраняет логин, пароль пользователя.

Void auth::Register() {

String^ cliLogin = textBoxLogin1->Text;

String^ cliPassword = textBoxPassword1->Text;

USER = cliLogin;

labelWelcome->Text += cliLogin;

SetCenter(pageWelcome, labelWelcome, 1);

std::string Login = msclr::interop::marshal\_as<std::string>(cliLogin);

std::string Password = msclr::interop::marshal\_as<std::string>(cliPassword);

String^ folderPath = cliLogin;

Directory::CreateDirectory(folderPath);

String^ filePath = Path::Combine(folderPath, "data.bin");

std::hash<std::string> hasher;

size\_t hashPassword = hasher(Password);

FileStream^ fileStream = gcnew FileStream(filePath, FileMode::Create, FileAccess::Write);

BinaryWriter^ binaryWriter = gcnew BinaryWriter(fileStream);

binaryWriter->Write(hashPassword);

binaryWriter->Close();

fileStream->Close();

animation\_mode = 1;

sec = 4;

timerTransition->Start();

}

Реализация функции входа в приложение:

Обработчик события кнопки «Войти» проверяет корректность введенных данных в поля, активировал ли пользователь функцию запоминания данных при последующих входах, и вызывает функцию входа типа Boolean, которая проверяет данные сохранённые в файлы и введенные пользователем. В случае сходства данных функция возвращает true иначе false.

Если функция возвращает true, обработчик активирует анимацию перехода на основную форму. Приветствие показывается если пользователь включил это в настройках.

Void auth::buttonComeIn\_Click(Object^ sender, EventArgs^ e) {

if (textBoxLogin->Text == "") {

textBoxLogin->BorderColor = Color::Red;

MessageWarning->Show("Введите логин");

return;

}

if (textBoxPassword->Text == "") {

textBoxPassword->BorderColor = Color::Red;

MessageWarning->Show("Введите пароль");

return;

}

if (ComeIn()) {

if (SwitchRemember->Checked == true) {

CreateLogFile();

}

if (greeting) {

animation\_mode = 4;

sec = 4;

}

else {

animation\_mode = 5;

sec = 1;

}

labelWelcome->Text += USER;

SetCenter(pageWelcome, labelWelcome, 1);

timerTransition->Start();

}

}

Boolean auth::ComeIn() {

String^ cliLogin = textBoxLogin->Text;

String^ cliPassword = textBoxPassword->Text;

std::string Login = msclr::interop::marshal\_as<std::string>(cliLogin);

std::string Password = msclr::interop::marshal\_as<std::string>(cliPassword);

String^ folderPath = cliLogin;

String^ filePath = Path::Combine(folderPath, "data.bin");

if (!Directory::Exists(folderPath)) {

MessageError->Caption = "Ошибка";

MessageError->Text = "Пользователя с указанным логином не существует";

MessageError->Show();

return false;

}

std::hash<std::string> hasher;

size\_t hashPasswordInput = hasher(Password);

FileStream^ fileStream = gcnew FileStream(filePath, FileMode::Open, FileAccess::Read);

BinaryReader^ binaryReader = gcnew BinaryReader(fileStream);

size\_t hashPasswordFile = binaryReader->ReadUInt64();

binaryReader->Close();

fileStream->Close();

if (hashPasswordInput == hashPasswordFile) {

USER = cliLogin;

return true; // Пароль совпадает

}

else {

MessageError->Caption = "Предупреждение";

MessageError->Text = "Введен неверный пароль";

MessageError->Show();

return false; // Пароль не совпадает

}

}

Текст остальных программных модулей представлен в приложении А.

3.3 Входные и выходные данные

Входные данные:

1. Логин и пароль: Строка, содержащая логин из букв и (или) цифр.
2. Имя и Фамилия: Строка, содержащая имя пользователя из букв и (или) цифр.
3. Дата рождения: Выбранные элементы из массива.
4. Электронная почта: Строка, являющаяся адресом, которая имеет ключевые символы.
5. Исходный код на Python: Строка, содержащая код задачи на языке Python.
6. Аватар пользователя: Изображение, имеющее jpg или png формат.

Выходные данные:

1. Логин: Папка, имеющая наименование равное логину пользователя
2. Файл с паролем: Хешированная строка, записанная в бинарном файле.
3. Файл с датой рождения: Дата, содержащая информацию о дате рождения пользователя, записанная в бинарном виде.
4. Файл с именем: Строка, содержащая имя пользователя.
5. Файл с полом: Строка, содержащая пол пользователя.
6. ­Файл с фамилией: Строка, содержащая фамилию пользователя, если он её указывал.
7. Файл с прогрессом: Число, преобразованное в строку и записанная в бинарный файл с помощью специального ключа шифрования.
8. Файл с уровнем: Строка, содержащая информацию о уровне пользователя.
9. Файл с состоянием задач: Словарь, содержащий информацию о состояния решения каждой задачи.
10. Тесты для задачи: Скрипт, содержащий Unit-тесты написанные на Python, который проверяет исходный код пользователя на Python.
11. Файл отчета скрипта: Строка, сообщающая программе результаты тестов.
12. Файл решения: Отчёт, содержащий код решения задачи пользователя.

3.4 Функциональное тестирование

Функциональное тестирование – это вид тестирования, который проверяет соответствие функциональности продукта тому, как он был задуман. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Тест-кейсы для основных функций представлены в таблице 3.1.

Таблица 3.1 – Функциональное тестирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №№ | Модуль / Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
|  | Авторизация пользователей | 1. Ввести существующие и верные логин и пароль. 2. Нажать кнопку «Войти» | Открытие главного окна на странице «Главная» | Результат представлен на рисунке Б.1, Б.2 и Б.3 |
|  | Функция «Запомнить меня» | 1. Ввести существующие и верные логин и пароль. 2. Включение функции нажатием на кнопку «Запомнить меня». 3. Нажать кнопку «Войти» | Открытие главного окна на странице «Главная» при запуске файла | Результат представлен на рисунке Б.4 и Б.5 |
|  | Регистрация пользователей | 1. Ввести корректные логин, пароль и пароль повторно. 2. Прочитать и принять условия пользования нажав «Я принимаю условия использования» 3. Нажать на кнопку «Регистрация» | Перенаправление на начальное окно заполнения профиля | Результат представлен на рисунке Б.4 и Б.5 |
|  | Заполнение начального профиля | 1. Ввести корректные личные данные 2. Ввести Email 3. Нажать на кнопку «Отправить письмо» 4. Ввести полученный код в поле 5. Загрузить аватар 6. Нажать на кнопку «Продолжить» | Перенаправление на страницу настроек | Результат представлен на рисунке Б.6, Б.7 и Б.8 |
|  | Первоначальная настройка | 1. Изменить нужные параметры 2. Нажать «Продолжить» | Перезапуск программного средства | Результат предоставлен на рисунке Б.9 и Б.10 |
|  | Просмотр личной информации пользователя | 1. Выбрать страницу «Профиль». 2. Ознакомиться с информацией | Предоставленные личные данные пользователю | Результат предоставлен на рисунке Б.11 |
|  | Редактирование личной информации пользователя | 1. Выбрать страницу «Профиль» 2. Нажать на кнопку «Редактировать» 3. Редактировать разрешенные поля | Измененные личные данные, сохранение в файлы | Результат представлен на рисунке Б.12 |
|  | Редактирование Email | 1. Выбрать страницу «Профиль» 2. Нажать на кнопку «Редактировать» 3. Изменить Email в поле 4. Нажать кнопку «Отправить письма» 5. Ввести первый полученный с ранее привязанной почты код в поле «Код с старой почты» 6. Ввести второй полученный код в поле «Код с новой почты» 7. Нажать кнопку «Подтвердить» | Измененный Email, сохранение в файл | Результат представлен на рисунке Б.13, Б.14 |
|  | Смена пароля | 1. Ввести действительный пароль в поле «Старый пароль» 2. Ввести новый пароль в поле «Новый пароль» 3. Ввести новый пароль еще раз в поле «Повтор пароля» 4. Нажать кнопку «Изменить» | Измененный пароль, сохранение в файл |  |
|  | Удаление аккаунта | 1. Ввод пароля в поле «Старый пароль» 2. Нажать на гиперссылку «Хотите удалить аккаунт?» 3. Подтвердить действие | Удаление директории пользователя и его содержимого |  |
|  | Переход на другую страницу без сохранения профиля | 1. Изменить данные профиля 2. Перейти на другую страницу 3. Подтвердить действие | Переход без сохранения данных |  |
|  | Изменение настроек | 1. Выбрать страницу «Настройки» 2. Изменить значения 3. Нажать кнопку «Сохранить» | Измененные настройки, сохранение xml конфигурации |  |
|  | Посетить страницу «Задания» впервые | 1. Выбрать страницу «Задания» 2. Просмотреть анимацию 3. Подтвердить ознакомление с политикой задач | Просмотренная анимация |  |
|  | Реализация теоретического материала по курсам, с возможностью создания пометок в предоставленном материале | 1. Выбрать страницу «Книга» 2. Ознакомиться с материалом 3. Выделить текст 4. Пометить нужным способом 5. Очистить ненужные пометки | Изученный материал, сохраненные пометки в файлы |  |
|  | Реализация возможности прохождения практических заданий в виде решения задач при помощи интерпретатора Python и предоставление результата при о пройденных заданиях. | 1. Выбрать страницу «Задания» 2. Выбрать задание 3. Ознакомиться с поставленной задачей 4. Написать код в поле 5. Нажать «Проверить» 6. Нажать «Сохранить» | Пройденная задача, предоставленный результат, сохранение данных в файлы |  |
|  | Формирование отчетов прохождения практических заданий за выбранный промежуток времени. | 1. Выбрать страницу «Задания» 2. Выбрать промежуток времени, нажав «Период» 3. Нажать «Отчет» | Предоставленный отчет, возможность экспорта |  |
|  | Экспортирование отчетов | 1. Нажать «Отчет» 2. Подтвердить действие, нажав «Да» 3. Выбрать путь к файлу 4. Нажать «Сохранить» | Экспортированный отчет в выбранную директорию |  |
|  | Организация лаконичного интерфейса программного средства с помощью создания меню, кнопочных форм, панелей инструментов | * Выбор страницы * Просмотр анимации * Настройка внешнего вида | Удобный переход по страницам, анимированные элементы |  |
|  | Справочная система | 1. Нажать на иконку в виде книги с вопросом 2. Ознакомиться со справочной системой | Открытая справочная система |  |
|  | Реализация сохранения данных в файлы формата \*.docx, \*.txt, \*.bin, \*.rtf и чтение этих данных из них | * Изменить профиль * Изменить настройки * Решить задачу * Провести время в программном средстве | Сохранение данных в файлы |  |

3.5 Описание справочной системы

Средством разработки для создания справочной системы является программное средство Dr.Explain, которое помогает создавать файлы справки, документацию и руководства пользователя [7].

Справочная система состоит из начальной страницы, оглавления, 9-ти разделов и 3-х подразделов.

Раздел 1 «Запуск приложения» Предоставляет описание процесса запуска приложения в системе.

Подраздел 1.1 «Шаги по установке Python» является инструкцией по установке Python, для работы скрипта программного средства.

Раздел 2 «Авторизация» Является подробным обзором страницы входа.

Раздел 3 «Регистрация» Предоставляет инструкцию по регистрации.

Подраздел 3.1 «Заполнение профиля» Предоставляет подробную инструкцию по начальному заполнению профиля и личной информации. Подраздел 3.2 «Начальная настройка» Предоставляет инструкцию по ранней настройке внешнего вида приложения.

Раздел 4 «Главная страница» Предоставляет информацию об элементах на главной странице программного средства.

Раздел 5 «Книга» Предоставляет инструкцию по использованию функций книги.

Раздел 6 «Задания» Предоставляет подробную информацию о странице задач и о самих задачах.

Подраздел 6.1 «Отчеты» Предоставляет инструкция об отчетах

Раздел 7 «Профиль» Предоставляет пользователю информацию о его личных данных и профиле, а также возможности редактирования этих данных.

Раздел 8 «Статистика» Обозревает функции страницы статистики.

Раздел 9 «Страница настроек» Явно показывает функционал и советы по настройке.

Справочная система вызывается c помощью нажатия клавиши F1 на клавиатуре.

4 Применение

4.1 Назначение программного средства

Программное средство «PythonWave» предназначено для изучения языка программирования Python. Пользователи могут использовать это программное средство для обучения, создание отчётов своей продуктивности и получение базы Python. Так же познакомившись с этим языком, пользователь может понять симпатизирует ли ему этот язык в целом.

Ограничение, накладываемое на представленное программное средство, включает следующее: необходимость наличия операционной системы Windows.

4.2 Условия применения

Для функционирования программы требуется операционная система Windows, интерпретатор Python (рекомендуемые версии 3.12.3 или 3.12.4). Процесс запуска приложения заключается в перемещения содержимого папки «PythonWave» в выбранную директорию на компьютере и запуске файла «PythonWave.exe».

Заключение

Целью курсового проектирования было создание программного средства с целью обучения пользователя. Оно позволяет выполнять практические задачи, изучать теоретический материал, создавать отчеты, настраивать приложение. Была поставлена задача реализовать данное программное средство в среде разработки Visual Studio, на языке программирования C++ с использованием фреймворка CLI .NET. В процессе разработки серьезных трудностей не возникало.

В процессе разработки курсового проектирования были решены следующие задачи: Организация работы с файлами и xml конфигуратором; разработка классов для отслеживания прогресса и решенных задач; специальный класс анимации для управления прозрачности формы; обработка состояний наведения на элементы формы; обеспечение безопасности путем шифрования специальными ключами и хэшем, а также использованием электронной почты; разработка unit-тестов на Python для проверки кода.

Достоинством программного средства является простое взаимодействие пользователя с приложением. Удобный и лаконичный интерфейс с анимациями делает работу с программой максимально эффективной и приятной. Пользователю не нужно тратить много времени на изучение сложных инструкций или освоение новых навыков — все действия выполняются легко и быстро. Программа также обеспечивает высокий уровень интерактивности, что способствует лучшему усвоению материала.

Недостатком является то, что приложение является локальным. Пользователь не сможет войти в свой аккаунт на других устройствах, где нет сохраненных файлов. Это ограничивает мобильность и гибкость использования программного средства.

Улучшение данного программного средства возможно путем полного перехода на язык программирования C#, так как на нем корректно работают сторонние фреймворки. Подключение базы данных SQL для сохранения данных пользователя онлайн позволит создать систему учета прогресса, доступную с любого устройства. Также стоит рассмотреть доработку и пополнение задач для повышения их разнообразия и уровня сложности, что сделает обучение более комплексным и полезным.

Дополнительно можно рассмотреть внедрение возможностей для совместного обучения, таких как обмен задачами между пользователями и поддержка учебных групп, что позволит создать более интерактивное и социальное учебное окружение. Интеграция с популярными облачными сервисами для хранения данных также может повысить удобство использования приложения.

Список использованных источников

1. Багласова, Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г. Багласова, К.О. Якимович. – Минск : КБП, 2013. – 29 c.
2. Блок-схема [Электронный ресурс]. – Wikipedia, 2024. – Режим доступа : https://ru.wikipedia.org/wiki/Блок-схема. – Дата доступа : 15.04.2024.
3. Диаграмма классов [Электронный ресурс]. – Wikipedia, 2024. – Режим доступа : https://ru.wikipedia.org/wiki/Диаграмма\_классов. – Дата доступа : 17.04.2024.
4. C++ [Электронный ресурс]. – Wikipedia, 2024. - Режим доступа : https://ru.wikipedia.org/wiki/C++. – Дата доступа : 11.05.2024.
5. Visual Studio 2022 [Электронный ресурс]. – Microsoft, 2024. – Режим доступа : https://visualstudio.microsoft.com/ru/vs. – Дата доступа : 13.05.2024.
6. Windows Forms [Электронный ресурс]. – Wikipedia, 2024. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/. Дата доступа : 15.05.2024.
7. Guna Framework [Электронный ресурс]. – gunaui.com, 2024. – Режим доступа: <https://gunaui.com/>. – Дата доступа : 17.05.2024.
8. Bunifu Framework [Электронный ресурс]. – bunifuframework.com, 2024. – Режим доступа: https://bunifuframework.com/. – Дата доступа : 20.05.2024.
9. Dr.Explain [Электронный ресурс]. – drexplain.ru, 2024. – Режим доступа: https://www.drexplain.ru/. – Дата доступа : 25.05.2024.

Приложение А

**(обязательное)**

**Текст программных модулей**

Приложение Б

**(справочное)**

**Результаты работы приложения**

На рисунках Б.1 - Б.10 представлены результаты тестирования приложения.

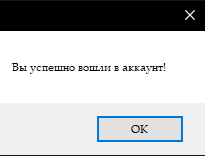


Рисунок Б.1 – Предупреждение об успешном входе

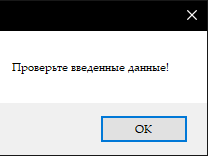


Рисунок Б.2 – Предупреждение об введении некорректных данных

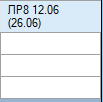


Рисунок Б.3 – Добавление лабораторной работы в таблицу

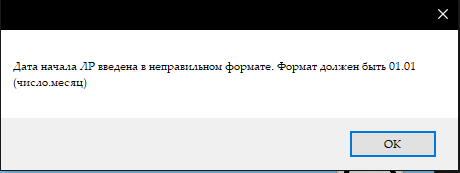


Рисунок Б.4 – Предупреждение об введении некорректных данных

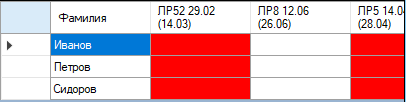


Рисунок Б.5 – Сортировка таблицы

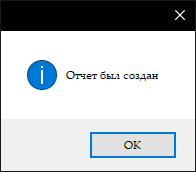


Рисунок Б.6 – Предупреждение об успешном создании отчета

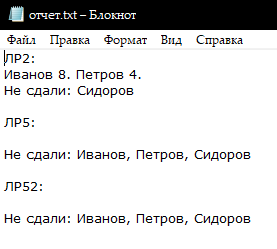


Рисунок Б.7 – Файл отчета

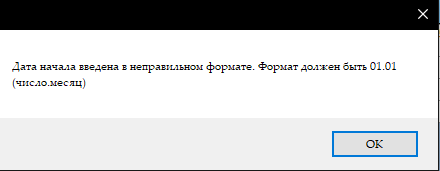


Рисунок Б.8 – Предупреждение об введении некорректных данных

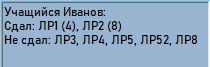


Рисунок Б.9 – Поиск данных из таблицы

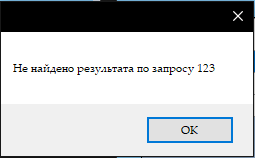


Рисунок Б.10 – Предупреждение об отсутствии результата