**Руководство программиста**

**«**Информационная система для фирмы по продаже комплектующих для ПК»

ОГЛАВЛЕНИЕ

[1. Инсталляция программы: 3](#_Toc182147814)

[1.1. Пользовательский вариант 3](#_Toc182147815)

[1.2. Вариант поддержания и модернизации 3](#_Toc182147816)

[2. Проверка работоспособности информационной системы 4](#_Toc182147817)

[3. Поддержка программы 5](#_Toc182147818)

[4. Модернизация 6](#_Toc182147819)

1. Инсталляция программы:
   1. Пользовательский вариант

* Перейдите по ссылке <https://github.com/DanMitozz/IS_For_PC_Parts_Company> и скачайте файлы для работы программы.
* Убедитесь, что все необходимые файлы, включая файл main, находятся в папке проекта.
* Запустите приложение и проверьте его работоспособность.
  1. Вариант поддержания и модернизации
* Убедитесь, что на вашем компьютере установлен Python с библиотекой PyQt5.
* Склонируйте репозиторий программы с GitHub с помощью команды: git clone <https://github.com/DanMitozz/IS_For_PC_Parts_Company>;
* Перейдите в папку вашего проекта, используя команду;
* Установите необходимые зависимости;
* Для проверки успешности установки запустите программу с помощью команды: python main.py.

1. Проверка работоспособности информационной системы

Для проверки работоспособности системы выполните следующие действия:

1. Запустите приложение.
2. Проверьте функции добавления и редактирования товаров, категорий и заказов.
3. Убедитесь, что все таблицы с данными о товарах, категориях и заказах отображаются корректно, и данные в них обновляются правильно.
4. Проверьте работу всех кнопок, включая добавление, удаление, редактирование данных, оформление заказов и генерацию отчетов, чтобы убедиться, что они работают корректно.
5. Поддержка программы

Для поддержания работоспособности и актуальности информационной системы выполните следующие действия:

1. **Обновление зависимостей**: Регулярно обновляйте Python и все используемые библиотеки, такие как PyQt5, до последних стабильных версий. Это поможет избежать проблем с совместимостью и повысит безопасность приложения.
2. **Отслеживание отзывов и ошибок**: Проверяйте репозиторий на GitHub на предмет новых отзывов, сообщений об ошибках и запросов от пользователей. При возникновении проблем оперативно анализируйте и исправляйте их, чтобы поддерживать высокое качество работы программы.
3. **Уведомления об изменениях**: При добавлении новых функций или изменении существующего функционала уведомляйте пользователей о нововведениях с помощью сообщений на GitHub. Это поможет пользователям всегда быть в курсе обновлений.
4. **Ведение документации**: Обновляйте и поддерживайте документацию к программе, включая инструкции по устранению проблем и ответы на часто задаваемые вопросы, и храните документацию в репозитории на GitHub.
5. **Резервное копирование данных**: Регулярно выполняйте резервное копирование базы данных, чтобы предотвратить потерю данных при сбоях или отказах системы. Храните резервные копии в надежном месте, отдельно от основного репозитория, чтобы обеспечить их сохранность.
6. **Оптимизация производительности**: Следите за производительностью программы и базы данных, проводите оптимизацию кода по мере необходимости, чтобы повысить стабильность и эффективность работы системы.
7. Модернизация

Для того чтобы информационная система для фирмы по продаже комплектующих для ПК оставалась актуальной и удовлетворяла потребности пользователей, важно регулярно проводить модернизацию. Вот основные действия, которые следует выполнять:

1. **Анализ обратной связи**. Постоянно собирайте и анализируйте отзывы и предложения пользователей. Это поможет выявить новые требования к системе и определить, какие улучшения или функции будут наиболее полезны. Таким образом, система будет становиться удобнее и функциональнее.
2. **Создание ветки для разработки**. Перед тем как приступить к разработке новых функций, создайте отдельную ветку в репозитории с помощью команды git checkout -b feature-branch. Это даст возможность вести разработку параллельно с основной версией программы, не нарушая ее работу, и минимизирует вероятность возникновения конфликтов при последующем слиянии кода.
3. **Реализация новых функций**. На основе потребностей пользователей и анализа работы системы реализуйте новые функции и улучшения. Используйте Python для разработки основной логики и PyQt5 для интерфейса. Примеры возможных улучшений включают добавление новых функций для управления товарами и заказами, улучшение интерфейса для повышения удобства использования и оптимизацию работы базы данных для повышения производительности.
4. **Тестирование изменений**. Проведите детальное тестирование добавленных функций и улучшений перед их выпуском. Это необходимо для проверки корректности их работы и исключения ошибок. Используйте инструменты для тестирования Python-кода. Это поможет избежать сбоев в работе системы.
5. **Обновление документации**. После добавления новых функций или изменений обновите документацию, чтобы она отражала актуальную версию программы. Это обеспечит пользователей полными и точными инструкциями по работе с новыми функциями. После обновления зафиксируйте изменения в репозитории, чтобы они были доступны для всех пользователей.
6. **Оптимизация и обновление зависимостей**. Регулярно проводите оптимизацию кода программы и базы данных, чтобы поддерживать стабильную и производительную работу системы. Также обновляйте используемые библиотеки и зависимости, чтобы избежать проблем с совместимостью и повысить безопасность системы.