Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**«Анализ исходных программ и компонентов программных средств»**

**«МДК 06.02** **Инженерно-техническая поддержка сопровождения информационных систем»**

Выполнил: студент учебной группы

ИСПк-302-52-00

Холин П.Э.

Преподаватель:

Самоделкин П.А.

Киров

2024

**Цель работы -** анализ исходных программ и компонентов программных средств, необходимых для сопровождения программных продуктов, с целью определения их структуры, функциональности, зависимостей и возможных улучшений.

**Задание**:

1. Из официальных источников найти и изучить документацию, связанную с программными средствами, установленными на рабочем месте. Это может быть описание функций, предназначение, а также другая информация, которая может помочь лучше понять структуру и особенности программных средств.
2. Проанализировать исходное состояние программ, необходимых для сопровождения, включая их версию, последнее обновление, структуру и основные компоненты, чтобы полностью понять, как работает программное средство.
3. Выявить зависимости между различными компонентами программного средства, как внутренними (внутри программы), так и внешними (например, зависимости от сторонних библиотек или сервисов). Это поможет понять, какие изменения в одной части программного средства могут повлиять на другие его части.
4. Выявить улучшения и возможности для апдейта / апгрейда, а также предложить улучшения или оптимизацию в исходных программах и компонентах программных средств на основе своего анализа.
5. Рассмотреть вопрос эффективности программного средства, его актуальности для той или иной задачи сопровождения.
6. В заключение лабораторного занятия необходимо задокументировать результаты своего исследования и анализа, включая описание программных средств, выявленные зависимости и предложенные улучшения. Это поможет им укрепить свои знания и навыки в анализе программного обеспечения (рабочего окружения). Отчет сдать преподавателю.

**Результаты выполнения задания**

Программа состоит из нескольких ключевых функций, каждая из которых выполняет определенные задачи для достижения основной цели программного средства. Функции можно классифицировать следующим образом:

* функции для пользовательского интерфейса: используются для отображения интерфейса и обработки взаимодействия пользователя с программой.
* функции для логики приложения: отвечают за выполнение основных расчетов и операций, таких как обработка данных, выполнение алгоритмов, реализация игровой логики.
* функции для работы с файлами: читают и записывают данные в файлы, обрабатывают входные и выходные данные.
* функции оптимизации: включают методы для работы с большими числами, символьными выражениями и многопоточностью, что позволяет ускорять сложные вычисления и решать задачи оптимизации, например, с матрицами и потоками.

**Анализ исходного состояния программы**

Программа написана на Python и используется в PyCharm. Текущая версия программы совместима с Python 3.10+, последние изменения в коде были внесены недавно, с акцентом на улучшение логики и производительности.

**Зависимости между компонентами**

Программа зависит как от внутренних связей между модулями, так и от внешних библиотек:

Внутренние зависимости:

– модуль интерфейса взаимодействует с бизнес-логикой через сигналы и слоты (если используется PyQt).

– модуль бизнес-логики опирается на функции для работы с данными (например, чтение матриц из файла).

– модули оптимизации и расчета данных взаимодействуют с потоками данных через матрицы и символьные выражения, поступающие от пользователя.

Внешние зависимости:

– NumPy используется для работы с матрицами.

– SymPy необходим для выполнения сложных математических вычислений.

– PyQt6 обеспечивает создание графического интерфейса и управление событиями.

**Возможности для улучшения**

– оптимизация вычислений: использование многопоточности и векторизации может быть расширено, чтобы повысить скорость обработки больших матриц и сложных символьных вычислений.

– модульность и рефакторинг: улучшить структуру кода, разделив большие модули на более мелкие, что повысит читаемость и тестируемость программы.

– улучшение работы с файлами: внедрить проверку целостности и правильности данных на этапе чтения файлов. Также возможно добавить поддержку других форматов данных.

– интерфейс пользователя: пересмотреть дизайн интерфейса с целью улучшения пользовательского опыта, добавив больше интерактивных элементов и визуальных подсказок.

– модернизация алгоритмов: провести анализ производительности алгоритмов и, если необходимо, обновить их, внедрив более современные подходы, такие как жадные алгоритмы или использование линейного программирования для оптимизационных задач.

**Эффективность программы и ее актуальность**

Программное средство эффективно выполняет задачи, связанные с обработкой данных и пользовательским взаимодействием. Однако в зависимости от объема данных или сложности задачи могут возникнуть проблемы с производительностью. Поэтому текущие обновления и апгрейды направлены на обеспечение устойчивой работы программы при высоких нагрузках и работе с большими объемами данных.

Для поддержки и сопровождения программы целесообразно внедрить автоматизированные тесты, которые проверят корректность работы всех функций после внесения изменений.

**Выводы по работе**

В результате работы были проанализированы исходные программы и компоненты программных средств, необходимых для сопровождения программных продуктов, с целью определения их структуры, функциональности, зависимостей и возможных улучшений. В качестве рассматриваемой программы был выбран Pycharm – кроссплатформенная интегрированная среда разработки для языка программирования Python