

УДК 004

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ
МОНИТОРИНГА УСТОЙЧИВОСТИ БОРТОВ КАРЬЕРОВ**

Сиводедова М.В., студент гр. ПИМ-241, II курс,
Черепанов П.В., студент гр. ПИМ-241, II курс
Научный руководитель: Тайлакова А.А., к.т.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

Проблема обеспечения безопасности на открытых горнодобывающих предприятиях всегда была острой и актуальной.

При открытых горнодобывающих работах важно вовремя обнаружить опасные тенденции в движении откосов. А также определить направление вектора такого движения, для дальнейшего принятия решения о дополнительном укреплении откоса или прекращении добычи.

Есть множество методик обработки результатов натурных наблюдений и разнообразие маркеров, сигнализирующих о рисках обрушения. Универсальную формулу вывести крайне сложно, поскольку оценка устойчивости состоит из комплекса критериев. Набор и степень важности, которых может варьироваться от карьера к карьере из-за разницы в породах, например.

Часто выбор инструмента автоматизации расчета и анализа данных сводится к компромиссу между гибкостью электронных таблиц, удобных для визуализации без жесткой привязки к методике и узким кругом специальных программ, каждая из которых реализует свой уникальный подход к вопросу оценки стабильности откосов.

Именно пример такого программного обеспечения предлагается разобрать. В основе любой программы, обрабатывающей данные будет.

Разработанная база данных включает семь таблиц:

- Coalpits (карьеры) – содержит информацию об объектах мониторинга;
- SettingsGraph (настройки графика) – хранит параметры отображения и границы зон;
- Repers (реперы) – каталог наблюдательных знаков;
- Coordinates (координаты) – массив исходных координат реперов;
- DeltaCoordinates (разности координат) – таблица для хранения вычисленных смещений между замерами;
- Results (результаты) – итоговые данные после пересчета;
- ResultsVectors (векторы смещений) – данные о результирующих векторах перемещений.

Схема связей между таблицами в нотации «сущность–связь» представлена на рисунке 1.

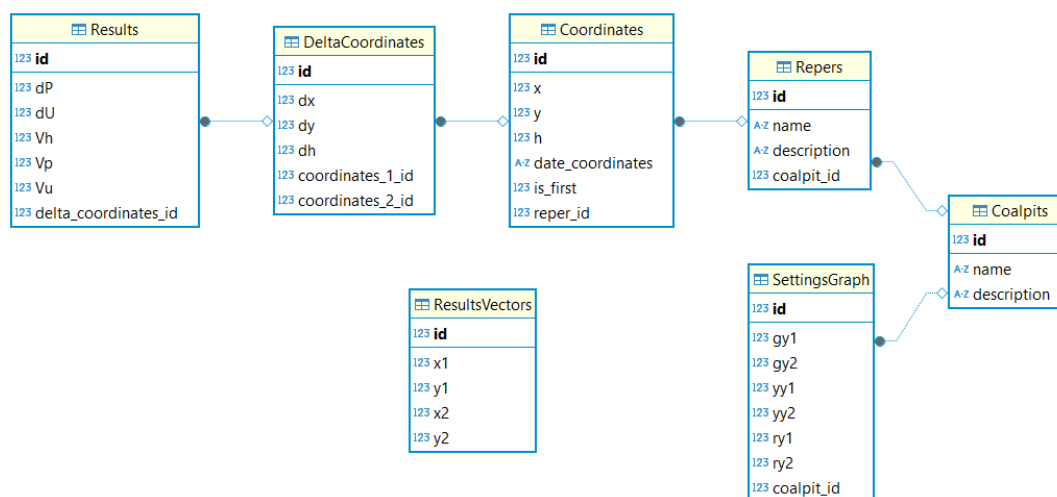


Рисунок 1– Структура базы данных

В проекте используется компактная система управления базами данных SQLite. Её ключевое преимущество – отсутствие необходимости в выделенном сервере: база данных хранится локально на устройстве пользователя. Это делает решение автономным, упрощает развертывание и идеально подходит для проектов с небольшим количеством специалистов.

Разработка велась в среде PyCharm с использованием Qt Designer для проектирования интерфейса. Исходный код написан на языке Python с использованием библиотеки PySide для графической оболочки. Кратко охарактеризуем основные компоненты:

- PyCharm – мощная среда разработки.
- Qt Designer – инструмент визуального проектирования интерфейсов, который позволяет создавать пользовательский интерфейс через перемещение и размещение элементов в окне редактирования приложения.
- Python – гибкий язык программирования, для которого доступно множество библиотек для обработки и анализа данных.

В ходе разработки ключевого функционала использовались различные библиотеки. Для создания кроссплатформенного настольного приложения была использована библиотека PySide, так как она имеет лицензию свободнее, чем PyQt. Для визуализации данных, а именно для построения графиков смещений и векторов перемещений, применялась Matplotlib. Взаимодействие с базой данных организовано через SQLite3, а для упаковки проекта в исполняемый exe-файл использовался PyInstaller. Экспорт данных в формат DXF, совместимый с AutoCAD, выполняется с помощью Ezdxf. Работа с Excel-файлами, а именно импорт и экспорт данных из форматов xlsm и xls, была реализована через openpyxl и xlrd. А для формата CSV задействована встроенная библиотека csv.

Разработанное приложение представляет собой законченное решение для автоматизации обработки данных геомеханического мониторинга на открытых горных разработках. Благодаря продуманной структуре базы данных и гибкому набору поддерживаемых форматов (Excel, CSV, DXF) программа легко интегрируется в существующие рабочие процессы. Использование

Python и библиотеки PySide позволило создать кроссплатформенное настольное приложение, сочетающее скорость вычислений специализированного ПО с удобством интуитивно понятного интерфейса. (Рис. 2).



Рисунок 2 – интерфейс приложения

Таким образом, предложенный подход демонстрирует, что эффективная автоматизация мониторинга откосов не требует громоздких серверных решений. Компактное локальное приложение, построенное на описанной архитектуре, способно обеспечить оперативную обработку результатов измерений, своевременное выявление опасных тенденций и, как следствие, повысить общий уровень безопасности горных работ.

Список литературы:

1. PySide6 6.11.0 // pypi.org URL: <https://pypi.org/project/PySide6/> (дата обращения: 28.03.2026).
2. Matplotlib 3.10.8 documentation // matplotlib.org URL: <https://matplotlib.org/stable/index.html> (дата обращения: 28.03.2026).
3. sqlite3 – DB-API 2.0 interface for SQLite databases // docs.python.org URL: <https://matplotlib.org/stable/index.html> (дата обращения: 28.03.2026).