

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА МОДУЛЯ КОНВЕРТАЦИИ ВХОДНЫХ
ДАНЫХ И КОМПОНОВКА ПРИЛОЖЕНИЯ GEOSTATSGU**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.04.01 — Информатика и вычислительная техника
факультета КНиИТ
Хомякова Вадима Сергеевича

Научный руководитель
Старший преподаватель

М. В. Белоконь

Заведующий кафедрой
доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы обусловлена стремительным ростом объёмов метеорологических и климатологических данных, формируемых современными системами численного моделирования атмосферы и спутниковыми комплексами. Эффективная обработка, анализ и визуализация многомерных полей (NetCDF, GRIB) являются критически важными этапами синоптического и климатического анализа. Однако в настоящее время на кафедре метеорологии и климатологии географического факультета данные операции выполняются преимущественно вручную с использованием табличных редакторов, таких как Microsoft Excel. Подобный подход требует значительных временных затрат на рутинное преобразование форматов, очистку данных и построение графиков, что существенно снижает эффективность непосредственно научного анализа и отнимает время у решения прикладных задач.

Кроме того, в качестве основного инструмента конвертации данных из формата NetCDF в табличный вид традиционно применяется программный комплекс Panoply Data Viewer. В условиях изменившейся технологической среды и ограничений доступа к ряду зарубежных научных инструментов на территории РФ, использование данного ПО стало сопряжено с техническими и лицензионными трудностями, что создаёт дополнительные риски для бесперебойности исследовательского и учебного процессов. В связи с этим возникает объективная потребность в разработке отечественного, локализованного и кроссплатформенного десктопного приложения, объединяющего функции загрузки, конвертации, математической обработки и интерактивной визуализации метеоданных в единой среде без зависимости от внешних проприетарных решений.

Целью настоящей преддипломной практики является разработка модуля конвертации входных данных и компоновка приложения GeostatSGU.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить особенности представления, структуры и метаданных исходных метеорологических данных в форматах NetCDF и GRIB, а также проанализировать существующие подходы к их обработке.
2. Выбрать и обосновать стек технологий, архитектурные паттерны и программные библиотеки для реализации десктопного приложения с гра-

фическим интерфейсом.

3. Разработать модуль конвертации входных данных, обеспечивающий чтение многомерных массивов, парсинг атрибутов, выполнение физических и статистических расчётов, а также экспорт результатов в формат Excel (.xlsx) с сохранением метаданных.
4. Интегрировать разработанные программные модули в единую архитектуру приложения, обеспечить взаимодействие компонентов, реализовать пользовательский интерфейс для картографической визуализации и оптимизировать отзывчивость системы при работе с большими объёмами данных.

Объектом исследования выступают процессы обработки, анализа и визуализации многомерных метеорологических данных в формате NetCDF.

Предметом исследования являются методы и программные средства конвертации и интерактивного отображения метеорологических полей в составе десктопного приложения GeostatSGU, а также подходы к оптимизации пользовательского интерфейса и производительности при работе с крупными сетками.

Научная новизна работы заключается в разработке интеграционной архитектуры, объединяющей функции парсинга многомерных метео данных, выполнения специализированных физических расчётов, картографической визуализации и экспорта результатов в единой локальной среде с открытым исходным кодом. Предложен подход к асинхронной обработке и ленивой загрузке чанков NetCDF-файлов в десктопном приложении на Python, позволяющий стабильно работать с глобальными моделями высокого разрешения без перегрузки оперативной памяти.

Практическая значимость исследования состоит в создании готового к внедрению программного инструмента, который позволяет автоматизировать рутинные операции преобразования и анализа данных, сокращая время подготовки материалов для климатологического анализа. Приложение может быть непосредственно использовано в учебном и исследовательском процессе кафедры метеорологии и климатологии СГУ, а также научными сотрудниками и студентами для оперативной верификации численных моделей, построения аномалий и экспорта выборок в табличные форматы. Модульная архитектура и использование библиотек с пермиссивными лицензиями обес-

печивают возможность дальнейшего развития системы, адаптации под специфические исследовательские задачи и интеграции новых алгоритмов без модификации ядра приложения.

В работе использованы методы объектно-ориентированного анализа, архитектурного проектирования, асинхронного и многопоточного программирования.

Бакалаврская работа состоит из введения, трёх разделов, заключения, списка использованных источников и приложений. Во введении сформулированы цель, задачи, объект и предмет исследования, обоснована актуальность темы, раскрыты научная новизна и практическая значимость. Первый раздел посвящен анализу предметной области и выбору инструментальных средств. Во втором разделе описывается проектирование архитектуры приложения и модуля конвертации данных. Третий раздел содержит описание реализации, интеграции компонентов, тестирования и результатов работы приложения. В заключении подводятся итоги выполненной работы и намечаются перспективы дальнейшего развития проекта.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первом разделе проведён комплексный анализ предметной области обработки многомерных метеорологических данных. Детально рассмотрена специфика форматов NetCDF и GRIB: описана логическая модель NetCDF, базирующаяся на четырёх фундаментальных компонентах (переменные, размерности, атрибуты, типы данных), и её преимущества для сохранения естественной многомерной топологии атмосферных полей. Показано, что в отличие от табличных форматов, NetCDF позволяет выполнять эффективные срезы по произвольным осям (широта, долгота, время, вертикальный уровень) без предварительной «развёртки» массива, что критически важно для синоптического анализа.

Описана иерархическая структура формата GRIB, оптимизированного для компактного хранения гридованных данных: использование алгоритмов сжатия с контролируемыми потерями позволяет сократить объём файлов в 3–5 раз, а поддержка разнообразных сеток (регулярных, икосаэдрических, спектральных) обеспечивает совместимость с выходными данными глобальных и региональных моделей. Подчёркнута роль стандарта Climate and Forecast (CF) Metadata Conventions в обеспечении интероперабельности: атрибуты `standard_name`, `units`, `valid_range` позволяют автоматически интерпретировать смысл переменных и корректно выполнять операции интерполяции, усреднения и расчёта производных величин.

Выполнен детальный сравнительный анализ четырёх наиболее распространённых программных средств визуализации и анализа климатических данных:

- Panoply (NASA GISS) — кроссплатформенное приложение с интуитивным интерфейсом, однако с ограниченным функционалом математической обработки и отсутствием встроенного экспорта в табличные форматы;
- GrADS (IGES) — мощная скриптовая среда для углублённого анализа, но с высоким порогом входа и устаревшим графическим интерфейсом;
- QGIS с метеорологическими плагинами — развитые картографические возможности, но ограниченная нативная поддержка многомерных временных рядов NetCDF;
- Metview (ECMWF) — профессиональная среда с полным циклом ана-

лиза, но со сложностью развёртывания и ограничительной лицензией. Систематизированы ограничения существующих решений: лицензионные и инфраструктурные зависимости от внешних серверов, высокий порог входа, отсутствие встроенных модулей для физических расчётов, фрагментарная поддержка экспорта в Excel. На основе анализа сформулированы требования к разрабатываемому приложению: полная локальность работы, интуитивный графический интерфейс, модульная расширяемость, интеграция расчётов и визуализации в единой среде, соответствие лицензионным требованиям за счёт использования библиотек с пермиссивными лицензиями.

Обоснован выбор архитектурного паттерна Model-View-ViewModel (MVVM) как оптимального для десктопных приложений с реактивным интерфейсом. Показано, что в отличие от MVC и MVP, MVVM обеспечивает двустороннюю привязку данных через слой ViewModel, что минимизирует объём связующего кода и гарантирует предсказуемость поведения системы при изменении состояния. Описана адаптация паттерна для экосистемы PyQt6 через комбинацию класса QObject, механизма сигналов/слотов и явного управления состоянием.

Сформирован и детально обоснован стек технологий:

- Python 3.10+: выбран за зрелую экосистему научного ПО, кроссплатформенность, поддержку статической типизации и возможность интеграции с низкоуровневыми библиотеками на C/C++;
- numpy: библиотека для работы с N-мерными массивами с именованными размерностями, нативной поддержкой NetCDF/GRIB и механизмами ленивой загрузки и чанкинга;
- PyQt6: фреймворк для построения графического интерфейса с развитой событийной моделью и поддержкой многопоточности через QThread;
- Cartopy + Matplotlib: связка для картографической визуализации с поддержкой проекций, географических признаков и интерактивной навигации;
- openpyxl: библиотека для программного создания структурированных Excel-файлов с сохранением форматирования и метаданных.

Во втором разделе спроектирована модульная архитектура приложения GeostatSGU на базе паттерна MVVM с централизованным управлением состоянием через объект AppState. Определены границы ответственности и

контракты взаимодействия трёх изолированных слоёв:

Слой Model инкапсулирован в классе WeatherDataModel и отвечает за низкоуровневую работу с данными. Реализована многоуровневая стратегия выбора вычислительного движка: для файлов GRIB активируется библиотека cfgrib, для NetCDF применяется каскадный перебор движков с автоматическим фоллбэком. Особое внимание уделено кроссплатформенной надёжности: при обнаружении не-ASCII символов в пути к файлу автоматически создаётся временная копия в директории с ASCII-идентификатором, что обеспечивает совместимость с низкоуровневыми C/C++ библиотеками в среде Windows.

Разработан иерархический алгоритм извлечения географических координат: сначала анализируются координаты, явно привязанные к переменной, затем — координаты уровня датасета с использованием пула синонимичных имён (lat, latitude, y, XLAT, grid_latitude), и наконец — поиск по атрибуту standard_name. Такой подход обеспечивает автоматическую совместимость с выходными данными различных численных моделей без ручной настройки парсера.

Слой ViewModel представлен набором специализированных классов (LoadFilesVM, VisualizationVM, TableVM, CalculationsVM, MeanFieldsVM), каждый из которых наследует QObject и использует декораторы @pyqtSlot для обработки команд. Ключевым архитектурным решением является введение класса AppState — реактивного хранилища состояния приложения, содержащего путь к файлу, индексы времени и уровня, выбранные переменные и результаты расчётов. При изменении любого свойства ViewModel обновляет AppState и испускает соответствующий сигнал (plot_data_ready, file_list_updated, levels_updated), что обеспечивает автоматическую синхронизацию всех вкладок интерфейса.

Слой View реализован в виде набора PyQt6-виджетов (MainView, LoadFilesView, VisualizationView, MapCanvas), отвечающих исключительно за отрисовку интерфейса и обработку пользовательских событий. Виджет MapCanvas инкапсулирует логику работы с matplotlib и cartopy, предоставляя метод plot_2d_field(), принимающий готовые numpy.ndarray координат и данных, что сохраняет строгую изоляцию слоёв.

Для работы с файлами, объём которых превышает доступную оперативную память, внедрены принципы ленивой загрузки и чанкинга. При вызове `xr.open_dataset()` создаётся объектная обёртка, содержащая только метаданные; физическое чтение массивов происходит исключительно при материализации срезов. Стратегия чанкинга применяется по измерениям времени и вертикального уровня: вызов `.isel(time=index)` обращается к файловой системе только за необходимыми блоками данных, что минимизирует количество системных вызовов ввода-вывода. Архитектура проектировалась с учётом прозрачной интеграции с библиотекой Dask для распараллеливания вычислений между ядрами CPU.

Спроектирована подсистема визуализации на базе класса `MapCanvas`, наследующего `FigureCanvasQTAgg`. Для отображения данных на географической карте используется библиотека `Cartopy`: при создании осей указывается проекция (`PlateCarree`, `Mercator`, `LambertConformal`), а все вызовы методов отрисовки сопровождаются параметром `transform`, обеспечивающим корректное преобразование широтно-долготной сетки. Реализован девятиэтапный процесс рендеринга: очистка фигуры, валидация входных данных, определение диапазона значений, подготовка координатной сетки через `pr.meshgrid`, отрисовка растрового поля методом `pcolormesh`, наложение географических признаков, добавление цветовой шкалы и сетки координат, фиксация области отображения, финальная перерисовка.

Разработан асинхронный механизм вычислений на базе класса `CalcWorker` (файл `workers/calc_worker.py`), наследующего `QThread`. Воркер выполняет все тяжёлые операции (расчёт градиентов, дивергенции, вертикальной интерполяции) в методе `run()`, изолированном от основного цикла событий GUI. Для безопасной коммуникации с интерфейсом используются сигналы `progress` (промежуточный статус), `finished` (результат вычислений) и `error` (обработка исключений). Управление состоянием загрузки реализовано через метод `set_loading_state(bool)`: при запуске задачи интерфейс блокируется и отображается прогресс-бар, по завершении — происходит автоматическая разблокировка.

В третьем разделе осуществлена программная реализация спроектированных компонентов и проведено тестирование их совместного функционирования.

Реализация модуля конвертации и экспорта в Excel выполнена в рамках слоя TableVM и пользовательского интерфейса TableView. Метод `request_table_data(var_name, time_index, filters)` реализует многоэтапный конвейер подготовки данных: извлечение и валидация исходного массива, применение координатных срезов с автоматическим определением имён осей, пространственная фильтрация через `.where()`, матричное преобразование в `pandas.DataFrame`. В отличие от плоской развёртки, реализована оптимизированная матричная структура: индексы строк соответствуют массивам широты, столбцы — массивам долготы, что сохраняет естественную топологию грида и минимизирует объём памяти при отображении. Метаданные переменной (`units, long_name, time, level`) автоматически сохраняются в атрибуте `df.attrs` и при экспорте формируют отдельный лист Metadata в Excel-файле.

Для программного создания файлов `.xlsx` применяется связка `pandas.ExcelWriter` и движка `openpyxl`. Метод `export_to_excel()` реализует отказоустойчивый экспорт с автоматическим формированием многолистовой книги: лист Data содержит основную числовую матрицу, лист Metadata — пары «свойство–значение», лист Statistics — описательную статистику. Архитектура экспорта через `pd.ExcelWriter` обеспечивает буферизованную запись, предотвращая избыточное потребление оперативной памяти. Матричная структура данных также выступает инженерным ограничителем: даже для глобальных сеток высокого разрешения (0.25°) количество строк остаётся в пределах лимитов Excel (1 048 576), тогда как плоская развёртка могла бы превысить это ограничение.

Реализация подсистемы визуализации обеспечила бесшовную интеграцию научной графики `matplotlib` и геопространственных проекций Cartopy в событийную модель PyQt6.

Интеграция модулей, разработанных параллельно, выполнена через централизованную событийную архитектуру. Координирующим ядром выступает класс `MainView`, в методе `_connect_signals()` которого декларативно определяется вся топология взаимодействия: сигналы пользовательского интерфейса связываются со слотами `ViewModel`, результаты обработки маршрутизируются обратно в методы обновления представления. Объект `AppState` реализует паттерн «Единый источник правды», централизованно храня ключевые параметры сессии и устраняя необходимость прямой передачи объектов

между независимыми модулями. Физическая структура проекта организована по принципу изоляции доменов (`src/views/`, `src/viewmodels/`, `src/models/`, `src/workers/`), что минимизирует конфликты слияния кода в системе контроля версий.

Оптимизация производительности достигнута за счёт комбинации многопоточной архитектуры и механизмов эффективного использования памяти. Все ресурсоёмкие операции выполняются в фоновых потоках `QThread`, что гарантирует отсутствие блокировок основного цикла событий GUI. Ленивая загрузка и чанкинг через `chunk` позволяют инициализировать сеансы с файлами объёмом 50+ ГБ при потреблении <100 МБ RAM на этапе навигации. Реализована многоуровневая система валидации: фильтрация пропущенных значений через `pr.nanmin/pr.nanmax`, строгая проверка размерности перед вычислениями, автоматическое переключение в резервный режим визуализации при отсутствии опциональных зависимостей.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы была достигнута поставленная цель — разработана модульная архитектура и реализовано десктопное приложение GeostatSGU с графическим интерфейсом, предназначенное для загрузки, обработки, анализа и визуализации многомерных метеорологических данных в форматах NetCDF и GRIB, с возможностью экспорта результатов в табличный формат Excel.

Для достижения цели последовательно решены четыре задачи:

1. Изучить особенности представления, структуры и метаданных исходных метеорологических данных в форматах NetCDF и GRIB, а также проанализировать существующие подходы к их обработке.
2. Выбрать и обосновать стек технологий, архитектурные паттерны и программные библиотеки для реализации десктопного приложения с графическим интерфейсом.
3. Разработать модуль конвертации входных данных, обеспечивающий чтение многомерных массивов, парсинг атрибутов, выполнение физических и статистических расчётов, а также экспорт результатов в формат Excel (.xlsx) с сохранением метаданных.
4. Интегрировать разработанные программные модули в единую архитектуру приложения, обеспечить взаимодействие компонентов, реализовать пользовательский интерфейс для картографической визуализации и оптимизировать отзывчивость системы при работе с большими объёмами данных.

Полученные результаты полностью соответствуют требованиям кафедры метеорологии и климатологии географического факультета. Приложение демонстрирует отсутствие блокировок графического интерфейса при выполнении длительных операций, стабильную работу с глобальными сетками высокого разрешения, корректный экспорт данных в формат .xlsx с сохранением метаданных, а также готовность к подключению сторонних расчётных модулей через стандартизированные сигналы и форматы данных.

Практическая значимость работы заключается в автоматизации рутинных процессов обработки метеорологических данных на кафедре метеорологии и климатологии. Разработанное приложение заменяет трудоёмкие ручные операции в универсальных просмотрщиках (Panoply) и табличных редакто-

рах, сокращая время подготовки материалов для синоптического и климатического анализа. Открытая архитектура, использование библиотек с пермиссивными лицензиями (BSD, MIT, Apache-2.0) и кроссплатформенность обеспечивают возможность легального применения в учебном процессе, научных исследованиях и дальнейшего развития системы силами студентов и сотрудников.

Перспективы развития проекта включают:

- поддержку облачных форматов данных (Zarr) и интеграцию с распределёнными хранилищами (S3);
- полную интеграцию библиотеки Dask для параллельных вычислений на многопроцессорных системах;
- подключение алгоритмов машинного обучения для автоматического выявления синоптических аномалий и кластеризации погодных режимов;
- адаптацию интерфейса для разработки веб-версии приложения с использованием фреймворков на базе Python;
- расширение библиотеки физических расчётов: термодинамические индексы, параметры неустойчивости атмосферы, статистические климатические нормы и вероятностный анализ.

Бакалаврская работа выполнена в полном объёме, все задачи решены. Приложение GeostatSGU готово к внедрению в учебно-исследовательскую практику кафедры метеорологии и климатологии Саратовского национального исследовательского государственного университета имени Н. Г. Чернышевского. Разработанная модульная архитектура, механизмы оптимизации работы с большими данными и событийная модель интеграции создают надёжную основу для дальнейшего развития инструментальной базы метеорологического анализа и вносят вклад в развитие отечественного программного обеспечения для научных исследований в области климатологии и численного прогнозирования погоды.

Основные источники информации:

- 1 Unidata. NetCDF User Guide. Версия 4.9. [Электронный ресурс] URL: <https://docs.unidata.ucar.edu/nug/current/> (дата обращения: 12.11.2025)
- 2 Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S. et al. NetCDF Climate and Forecast (CF) Metadata Conventions (1.13). CF Community, 2025. DOI: 10.5281/zenodo.17801666

- 3 Hoyer, S. et al. xarray: N-D labeled arrays and datasets in Python // Journal of Open Research Software. 2017. Vol. 5, Iss. 1. DOI: 10.5334/jors.148
- 4 The Qt Company. PyQt6 Documentation. [Электронный ресурс] URL: <https://doc.qt.io/qtforpython-6/> (дата обращения: 15.04.2026)
- 5 Met Office / SciTools. Cartopy: A cartographic Python library with a Matplotlib interface. [Электронный ресурс] URL: <https://scitools.org.uk/cartopy/docs/latest/> (дата обращения: 15.04.2026)
- 6 openpyxl Documentation. [Электронный ресурс] URL: <https://openpyxl.readthedocs.io/> (дата обращения: 15.04.2026)
- 7 Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2003. 560 p.