

1. 0 мин 18 сек | 0 мин 18 сек

Здравствуйте, уважаемая комиссия!

Меня зовут Степанов Сергей и сегодня я хочу Вам рассказать о результатах, полученных в рамках работы над магистерской диссертацией по теме: Проектирование и реализация расчётного модуля для системы автоматического формирования генеральных планов площадных объектов капитального строительства.

2. 1 мин 10 сек | 1 мин 30 сек

Генеральным планом называется проектный документ, на основании которого осуществляется планировка, застройка, реконструкция и иные виды градостроительного освоения территорий.

За проектирование крупных технологических объектов отвечают проектные институты, а именно инженеры-проектировщики. Системы автоматического проектирования, которыми пользуются инженеры, не обладают полноценным инструментом стоимостного моделирования, поэтому инженер не может узнать заранее, насколько его решение экономически эффективно для проекта в целом, так как расчёт затрат происходит позже на этапе формирования сметы.

В силу высокой сложности проектирования, рынок нуждается в системе способной как формировать генпланы в автоматическом режиме, так и верифицировать существующие решения в соответствии с заданным набором правил, а также способной оценивать и сравнивать решения с точки зрения экономической эффективности.

Формирование генплана является сложной задачей, для которой отсутствуют готовые методики решения. Для создания подобной системы требуется проведение массы исследований в области алгоритмов. Срок проведения исследований может занимать не один год.

С целью снижения издержек на проведение научных изысканий требуется создание дополнительного инструментария, позволяющего повысить эффективность данного процесса.

3. 0 мин 24 сек | 1 мин 56 сек

Целью данной работы является **упрощение процесса проведения научных изысканий** в области автоматического формирования ге-

неральных планов площадных объектов **путём создания программного компонента.**

Исходя из данной цели можно выделить следующие *задачи*:

- сбор и анализ требований пользователей системы,
- анализ возможной нагрузки и вариативности используемых данных,
- формирование системной и программной архитектуры,
- реализация полученного решения.

4. X мин X сек | X мин X сек

Есть несколько различных вариантов использования системы, рассмотрим самый распространенный: добавлен новый расчётный элемент или изменены требования размещения к существующему.

На слайде изображена диаграмма процесса проведения исследований и внедрения алгоритмической методики. Для оперативности её понимания, приведем пример её использования.

Заказчик попросил построить эстакады трёх различных типов: подземные, наземные, надземные. Бизнес-аналитик обращается к эксперту с просьбой предоставить проектные данные, где есть коммуникации, тех типов, которые можно разбить на разные типы эстакад. На основе этих данных аналитик формирует кейс в формате системы. Далее аналитик ставит задачу исследователям и сообщает, как получить данные кейса. Далее начинается процесс исследований в рамках которого и определяется способ получения необходимого результата.

Если аналитик делает вывод, что данная методика, позволяет получить решение, соответствующее требованиям заказчика, то результаты эксперимента уже показываются техническому эксперту со стороны заказчика, чтобы он дополнительно убедился, что реализация данной методики не противоречит другим требованиям проектирования генпланов.

5. X мин X сек | X мин X сек

После проведенного интервью с пользователями системы был проведен анализ и сформированы, как функциональные требования системы.

6. X мин X сек | X мин X сек

Так и нефункциональные. Выполнение этих требований будет рассмотрено далее.

7. X мин X сек | X мин X сек Систему можно разделить на две части: бизнес-часть и расчётную часть.

Бизнес-часть приложения используется для отображения полученных результатов внешним экспертам со стороны заказчика.

В расчётной части системы используются объекты внутренней модели данных. Эти внутренние объекты уже не могут быть интерпретированы внешними экспертами.

Именно проектирование и реализация расчётной части системы является результатом данной работы.

8. X мин X сек | X мин X сек Для решения поставленной задачи предлагается следующая архитектура расчётного модуля. Разрабатываемые компоненты выделены на диаграмме синим цветом.

- *Plan Design Algorithms* – математическая библиотека, содержащая алгоритмы для построения генеральных планов.
- *Plan Design Model* – расчётная модель данных.
- *Executor* – сервис, отвечающий за запуск методов математической библиотеки.
- *Storage* – сервис, отвечающий за чтение/запись расчётной модели данных.
- *Orchestrator* – сервис, контролирующий выполнение расчётных задач.

Самой ресурсоёмкой частью приложения является запуск математических методов. Сервис *Executor*, отвечающий за это, отвязан от использования других компонент системы и может работать полностью изолированно. Поэтому, в случае необходимости, возможно оперативное развертывание нескольких экземпляров данного сервиса на других вычислительных узлах.

Согласно первому нефункциональному требованию нам доступен только один сервер для развертывания системы, поэтому использование дополнительных архитектурных компонент является избыточным.

9. X мин X сек | X мин X сек

Расчётная модель данных подвержена сильным изменениям в процессе исследований. Новые алгоритмы решения задачи могут давать куда более качественный результат, но только с учетом того, что будут использованы дополнительные структуры данных. В то время как внешняя модель данных относительно стабильна и представлена объектами целиком и полностью понятными внешним экспертам.

Расчётная модель данных представлена следующими классами.

- **Model** – верхнеуровневый класс, который и является расчётной моделью данных, использующийся при передаче данных между сервисами. Состоит из расчётных элементов и глобального перечня типов объектов.
- **Feature** – расчётный элемент. Являются составными частями расчётной модели данных. Например, расчётными элементами являются местоположения сооружений, внутриплощадочные проезды и т.д.
- **Glossary** – объект, являющийся глобальным перечнем типов различных объектов. Например, типы сооружений, такие как "Сепаратор", "Трансформаторная подстанция", "Насосная" хранятся именно здесь. Помимо описания типа, именно здесь хранятся его различные характеристики. Для типа "Электрокабель 35 кВ" определены параметра диаметра сечения, удельного сопротивления и используемого материала.

10. X мин X сек | X мин X сек Математическая библиотека представляет собой набор математических методик. Математическая методика является классом алгоритмов, которые могут решать ту или иную задачу. Каждая математическая методика используется хотя бы в одном математическом методе.

Каждый метод запускается путём вызова метода *calculate* в классе *Calculator*. Для каждой методики может быть реализовано несколько интерфейсов *Calculator*, которые будут использовать один класс алгоритмов, но давать несколько иной результат.

Каждая реализация *Calculator* принимает входные данные *InputData* и возвращает выходные данные *OutputData*. Если методика подразумевает использование различных параметров настройки, то *Calculator* дополнительно принимает на вход *Configuration*. Если параметры настройки не требуются, то класс *Configuration* является *NULL*.

11. X мин X сек | X мин X сек На диаграмме изображена архитектура сервиса запуска расчётных задач. Самым важным модулем, который обеспечивает запуск расчётных задач является *Execution*. Он выделен синим цветом.

12. X мин X сек | X мин X сек Компонент *Execution* состоит из 4 компонент, которые обеспечивают выполнение той или иной части расчёта.

- *Task* – расчётная задача. Результатом выполнения задачи является генеральный план площадного объекта. Задача состоит из набора этапов, выполняющихся строго последовательно. Результат выполнения следующего этапа зависит от результатов предыдущего.
- *Stage* – этап расчётной задачи. Для каждого этапа определен хотя бы один расчёт. Расчёты могут выполняться параллельно. Результатом этапа может быть только один расчёт.
- *Calculation* – расчёт. Состоит из последовательности методов.
- *Method* – метод. Способ применения математической методики.

13. X мин X сек | X мин X сек Теперь перейдем к разработке. Помимо программной реализации необходимо уделить особое внимание вопросам развертывания приложения. Gitlab CI/CD является инструментом, используемым в компании для непрерывной интеграции и развертывания.

Главным объектом в системе Gitlab CI является Pipeline. Pipeline состоит из Stage, где каждый этап состоит из задач (Job).

На рисунке приведен скриншот Pipeline-а, используемого в проекте. Он состоит из пяти этапов: сборки, тестирования, проверки качества кода, развертывания и публикации пакетов.

14. X мин X сек | X мин X сек

Pipeline в Gitlab CI/CD представляют собой Directed Acyclic Graph(DAG). Stage выполняются строго последовательно, но между задача-ами можно устанавливать зависимости.

На рисунке приведен скриншот части интерфейса, на котором отображено дерево зависимостей.

Установления порядка выполнения задач позволяет оптимизировать время прохождения Pipeline-а. Если зависимости установлены, то выполнение задач следующего этапа может начаться не дожидаясь полного выполнения всех задач предыдущего этапа.

15. X мин X сек | X мин X сек

На данном слайде отражен пример отображение отчета о тестировании в формате JUnit интерфейсе GitLab.

16. X мин X сек | X мин X сек

На данном слайде приведен пример интерфейса по управлению окружениями в GitLab.

GitLab предоставляет возможность автоматически разворачивать изолированные тестовые окружение по нажатию кнопки в интерфейсе. Данный механизм позволяет безопасно проводить тестирование и интеграцию с новым функционалом независимо для каждой задачи в рамках проекта.