



Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»

КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по дисциплине «Модели и методы анализа проектных решений»

на тему

«@Тема работы@»

Студент @РК6-5ХБ@
группа

подпись, дата

@Фамилия И.О.@
ФИО

Руководитель КП

подпись, дата

@Фамилия И.О.@
ФИО

Консультант

подпись, дата

@Фамилия И.О.@
ФИО

Консультант

подпись, дата

ФИО

Нормоконтролёр

подпись, дата

@Фамилия И.О.@
ФИО

Москва, 2021

Министерство науки и высшего образования Российской Федерации
 федеральное государственное бюджетное образовательное
 учреждение высшего профессионального образования
 «Московский государственный технический университет имени Н.Э. Баумана
 (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой РК-6
индекс

_____ А.П. Карпенко

« ____ » _____ 2021 г.

ЗАДАНИЕ

на выполнение курсового проекта

Студент группы: @РК6-5ХБ@

@Фамилия Имя Отчество@

(фамилия, имя, отчество)

Тема курсового проекта : @Тема работы@

Источник тематики (кафедра, предприятие, НИР): кафедра

Тема курсового проекта утверждена на заседании кафедры «Системы автоматизированного проектирования (РК-6)», Протокол № _____ от « ____ » _____ 2021 г.

Техническое задание

Часть 1. Аналитический обзор литературы.

Более подробная формулировка задания. Следует сформировать, исходя из исходной постановки задачи, предоставленной руководителем изначально. Формулировка включает краткое перечисление подзадач, которые требовалось реализовать, включая, например: анализ существующих методов решения, выбор технологий разработки, обоснование актуальности тематики и др. Например: «В рамках аналитического обзора литературы должны быть изучены вычислительные методы, применяемые для решения задач кластеризации больших массивов данных. Должна быть обоснована актуальность исследований.»

Часть 2. Математическая постановка задачи, разработка архитектуры программной реализации, программная реализация.

Более подробная формулировка задания. В зависимости от поставленной задачи: а) общая тема части может отличаться от работы к работе (например, может быть просто «Математическая постановка задачи» или «Архитектура программной реализации»), что определяется целесообразностью для конкретной работы; б) содержание задания должно несколько детальнее раскрывать заголовки. Например: «Должна быть создана математическая модель распространения вирусной инфекции и представлена в форме системы дифференциальных уравнений.»

Часть 3. Проведение вычислительных экспериментов, отладка и тестирование.

Более подробная формулировка задания. Должна быть представлена некоторая конкретизация: какие вычислительные эксперименты требовалось реализовать, какие тесты требовалось провести для проверки работоспособности разработанных программных решений. Формулировка задания должна включать некоторую конкретику, например: какими средствами требовалось пользоваться для проведения расчетов и/или вычислительных экспериментов. Например: «Вычислительные эксперименты должны быть проведены с использованием разработанного в рамках ВКР программного обеспечения».

Оформление курсового проекта :

Расчетно-пояснительная записка на 15 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

количество: 0 рис., 0 табл., 9 источн.
[здесь следует ввести количество чертежей, плакатов]

Дата выдачи задания «DD» месяц 2021 г.

Студент

подпись, дата

@Фамилия И.О.@
 ФИО

Руководитель курсового проекта

подпись, дата

@Фамилия И.О.@
ФИО

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

РЕФЕРАТ

курсовой проект : 15 с., 12 глав, 0 рис., 0 табл., 9 источн.

@KEYWORDSRU@.

@Начать можно так: “Работа посвящена...”. Объём около 0.5 страницы. Здесь следует кратко рассказать о чём работа, на что направлена, что и какими методами было достигнуто. Реферат должен быть подготовлен так, чтобы после её прочтения захотелось перейти к основному тексту работы.@

Тип работы: курсовой проект .

Тема работы: «@Тема работы@».

Объект исследования: @Объект исследований@.

Основная задача, на решение которой направлена работа: @Основная задача, на решение которой направлена работа@.

Цели работы состоят в: @Цель выполнения работы@

В результате выполнения работы: 1) предложено ...; 2) создано ...; 3) разработано ...; 4) проведены вычислительные эксперименты ...

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. Постановка задачи	9
1.1. Концептуальная постановка задачи	9
1.2. Математическая постановка задачи (представляется в зависимости от задачи)	9
2. Вычислительный метод	9
3. Программная реализация	10
3.1. Архитектура	10
4. Тестирование и отладка	11
4.1.	11
5. Вычислительный эксперимент	12
5.1.	12
6. Анализ результатов	13
6.1.	13
ЗАКЛЮЧЕНИЕ	14
Литература	15

ВВЕДЕНИЕ

При разработке любого программного обеспечения разработчики ставят перед собой несколько очень важных задач: эффективность и гибкость дальнейшей поддержки программного обеспечения. В современной разработке существует множество паттернов проектирования, архитектур и вспомогательных систем, которые позволяют быстрее создавать программное обеспечение и в дальнейшем упрощают его поддержку для новых разработчиков. Таким образом, даже в небольших проектах, над которыми работает небольшая команда разработчиков стараются использовать системы контроля версий, юнит-тестирование. Если приложение представляет из себя API при его разработке используют специальные системы для быстрой документации API. Такие подходы считаются стандартом при разработке программного обеспечения рассчитанное на удовлетворение потребностей конечных пользователей - разработка мобильных приложений, проектирование и реализация API и прочее.

Однако при появлении первых ЭВМ в 1930-х годах [1] программирование использовалось для решения научных задач. Научное программирование сильно отличается от других видов программирования, при разработке научного программного обеспечения очень важно получить корректный и стабильный конечный продукт, а также четко разделить интерфейсную и научную часть. Из этого следует, что разработчик научного программного обеспечения должен быть экспертом в предметной области, а сам разработчик обычно является конечным пользователем [2], в то время как в индустриальном программировании разработчик зачастую не является конечным пользователем и от него не требуется быть экспертом в предметной области разрабатываемого приложения.

Также научное программирование отличается от индустриального программирования тем, что стандарты проектирования научного программного обеспечения вырабатываются существенно медленнее или не вырабатываются вовсе, что приводит к отсутствию каких-либо системных подходов к разработке. Это приводит к сложностям при валидации и дальнейшей поддержке кода. Так, код, написанный эффективно и корректно, может оказаться бесполезным в том если поддержка кода оказывается затруднительной, это приводит к формированию и накоплению “best practices” при программировании численных методов. Вследствие чего стали появляться системы, которые позволяют минимизировать написание кода и снизить трудозатраты на его поддержку.

В основном существующие платформы используют визуальное программирование [3]. Одними из самых популярных и успешных разработок в этой области являются Simulink и LabView. Simulink позволяет моделировать вычислительные методы с помощью графических блок-диаграмм и может быть интегрирован со средой MATLAB. Также Simulink позволяет автоматически генерировать код на языке C для реализации вычислительного метода в режиме реального времени. LabView используется для аналогичных задач - моделирование технических систем и устройств [4]. Среда позволяет создавать виртуальные приборы с помощью графической блок-диаграммы, в которой каждый узел соответствует выполнению какой-либо

функции. Представление программного кода в виде такой диаграммы делает его интуитивно понятным инженерам и позволяет осуществлять разработку системы более гибко и быстро. В составе LabView есть множество специализированных библиотек для моделирования систем из конкретных технических областей.

Существует множество других систем и языков программирования для реализации вычислительных методов, каждый из них является узкоспециализированным и решает определенную задачу. Так, например, система визуального моделирования FEniCS [5] используется для решения задач с использованием метода конечных элементов. Система имеет открытый исходный код, а также предоставляет удобный интерфейс для работы с системой на языках Python или C++. FEniCS предоставляет механизмы для работы с конечно-элементными расчетными сетками и функциями решения систем нелинейных уравнений, а также позволяет вводить математические модели в исходной интегрально-дифференциальной форме.

Отдельного упоминания стоит система TensorFlow. TensorFlow представляет из себя библиотеку с открытым исходным кодом, которая используется для машинного обучения. Аналогично LabView и Simulink, TensorFlow позволяет строить программные реализации численных методов. Стоит обратить внимание, что в основе TensorFlow лежит такое понятие как граф потока данных. В самом графе ребра - тензоры, представляют из себя многомерные массивы данных, а узлы - математические операции над ними.

Применение ориентированных графов очень удобно для построения архитектур процессов обработки данных (как в автоматическом, так и в автоматизированном режимах). Вместе с тем многочисленные возникающие в инженерной практике задачи предполагают проведение повторяющихся в цикле операций. Самым очевидным примером является задача автоматизированного проектирования (АП). Эта задача предполагает, как правило, постановку и решение некоторой обратной задачи, которая в свою очередь, часто, решается путём многократного решения прямых задач (простым примером являются задачи минимизации некоторого функционала, которые предполагают варьирование параметров объекта проектирования с последующим решением прямой задачи и сравнения результата с требуемым согласно заданному критерию оптимизации). Отметим, что прямые задачи (в различных областях) решаются одними методами, тогда как обратные - другими. Эти процессы могут быть очевидным образом отделены друг от друга за счет применения единого уровня абстракции, обеспечивающего определение интерпретируемых архитектур алгоритмом, реализующих методы решения как прямой, так и обратной задач. Очевидным способом реализации такого уровня абстракции стало использование ориентированных графов.

А.П.Соколов и А.Ю.Першин разработали графоориентированный программный каркас для реализации сложных вычислительных методов, теоретические основы представлены в работе [6], а принципы применения графоориентированного подхода зафиксированы в патенте [7]. Заметим, что в отличие от TensorFlow в представленном графоориентированном программном каркасе узлы определяют фиксированные состояния общих данных, а ребра определяют

функции преобразования данных. Для описания графовых моделей был разработан формат aDOT, который расширяет формат описания графов DOT [8], входящий в пакет утилит визуализации графов Graphviz. В aDOT были введены дополнительные атрибуты и определения, которые описывают функции-предикаты, функции-обработчики и функции перехода в целом. Подробное описание формата aDOT приведено в [9].

После исследования графоориентированного программного каркаса разработанного А.П.Соколовым и А.Ю.Першиным стало очевидно, что для построения графовых моделей необходим удобный графический редактор, который будет также выступать в роли визуализатора, то есть должен иметь возможность экспортировать графовую модель в формат aDOT, а также возможность и загружать граф из этого формата.

1 Постановка задачи

1.1 Концептуальная постановка задачи

В разделе концептуальная постановка задачи должны быть представлены: объект исследований (разработки), цель исследования (разработки), кратко задачи (по пунктам, не более 8), исходные данные (если предусмотрены), что требуется получить.

Обязательность представления: раздел обязателен.

Объём: как правило, не должен быть больше 1-2 страниц.

1.2 Математическая постановка задачи (представляется в зависимости от задачи)

Раздел математическая постановка задачи обязателен для проектов, предполагающих применение методов математического моделирования и, как следствие, проведение вычислительных экспериментов.

Если проект предполагает разработку программного обеспечения и не предполагает проведение вычислений, то этот раздел не обязателен.

В разделе математическая постановка задачи подробно по подразделам следует описать планируемые к применению математические модели, вычислительные методы. Следует описывать особые ситуации их применения, которые предполагается изучить. Модели следует описывать с использованием математически строгих формулировок, не допускающих неоднозначности прочтения.

Обязательность представления: раздел представляется в зависимости от задачи.

Объём: как правило, может составлять около 10 страниц.

Список использованных источников

- 1 Timothy Williamson. History of computers: A brief timeline. 2021.
- 2 J. E. Hannay C. MacLeod J. S. e. a. How do scientists develop and use scientific software? 2009.
- 3 Robert van Liere. CSE. A Modular Architecture for Computational Steering. 2015.
- 4 А.В. Коргин М.В. Емельянов В.А. Ермаков. Применение LabView для решения задач сбора и обработки данных измерений при разработке систем мониторинга несущих конструкций. 2013.
- 5 Mortensen Mikael Langtangen Hans Petter Wells Garth N. A FEniCS-Based Programming Framework for Modeling Turbulent Flow by the Reynolds-Averaged Navier-Stokes Equations. 2011.
- 6 Соколов А.П., Першин А.Ю. Графоориентированный программный каркас для реализации сложных вычислительных методов. 2019.
- 7 Соколов А.П., Першин А.Ю. Патент на изобретение RU 2681408. Способ и система графо-ориентированного создания масштабируемых и сопровождаемых программных реализаций сложных вычислительных методов. 2019.
- 8 Stephen C. North Eleftherios Koutsofios. Drawing Graphs With Dot. 1999.
- 9 Соколов А.П. Описание формата данных aDOT (advanced DOT) [Электронный ресурс]. Облачный сервис SA2 Systems. [Офиц. сайт]. 2020.