



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    «Робототехника и комплексная автоматизация»

КАФЕДРА        «Системы автоматизированного проектирования (РК-6)»

## РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе

на тему

«Разработка компонентов графоориентированного  
программного каркаса для реализации сложных  
вычислительных методов»

Студент РК6-81Б  
                 группа

\_\_\_\_\_  
подпись, дата

Тришин И.В.  
                 ФИО

Руководитель ВКР

\_\_\_\_\_  
подпись, дата

Соколов А.П.  
                 ФИО

Консультант

\_\_\_\_\_  
подпись, дата

Першин А.Ю.  
                 ФИО

Нормоконтролёр

\_\_\_\_\_  
подпись, дата

Грошев С.В.  
                 ФИО

Москва, 2022

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

« \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

## ЗАДАНИЕ

### на выполнение выпускной квалификационной работы

Студент группы: РК6-81Б

Тришин Илья Вадимович

\_\_\_\_\_  
(фамилия, имя, отчество)

Тема выпускной квалификационной работы: Разработка компонентов графоориентированного программного каркаса для реализации сложных вычислительных методов

Источник тематики (кафедра, предприятие, НИР): кафедра

Тема выпускной квалификационной работы утверждена распоряжением по факультету РК № \_\_\_\_\_ от « \_\_\_\_\_ » \_\_\_\_\_ 2022 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 заголовок. Например: «Должна быть создана математическая модель распространения вирусной инфекции и представлена в форме системы дифференциальных уравнений».

**Часть 3.** Проведение вычислительных экспериментов, 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 выпускной квалификационной работы:**

Расчетно-пояснительная записка на 33 листах формата А4.

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

количество: 1 рис., 0 табл., 13 источн.
[здесь следует ввести количество чертежей, плакатов]

Дата выдачи задания «08» февраля 2022 г.

Студент

Руководитель

выпускной

квалификационной работы

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
подпись, дата

Тришин И.В.  
\_\_\_\_\_  
ФИО

Соколов А.П.  
\_\_\_\_\_  
ФИО

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

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)

РК  
 ФАКУЛЬТЕТ  
 КАФЕДРА РК-6  
 ГРУППА РК6-81Б

УТВЕРЖДАЮ  
 Заведующий кафедрой РК-6  
индекс  
 \_\_\_\_\_ *А.П. Карпенко*  
 « \_\_\_\_ » \_\_\_\_\_ 2022 г.

## КАЛЕНДАРНЫЙ ПЛАН выполнения выпускной квалификационной работы

Студент группы: РК6-81Б

Тришин Илья Вадимович

(фамилия, имя, отчество)

Тема выпускной квалификационной работы: Разработка компонентов графоориентированного программного каркаса для реализации сложных вычислительных методов

№ п/п	Наименование этапов выпускной квалификационной работы	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулировка проблемы, цели и задач работы	18.02.2022	18.02.2022	Руководитель ВКР	Соколов А.П.
2.	1 часть: аналитический обзор литературы	18.02.2022	31.03.2022	Руководитель ВКР	Соколов А.П.



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

**НАПРАВЛЕНИЕ  
НА ЗАЩИТУ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Председателю  
Государственной Экзаменационной Комиссии № \_\_\_\_\_

факультета «Робототехника и комплексная автоматизация» МГТУ им. Н.Э. Баумана

Направляется студент Фамилия Имя Отчество группы РК6-81Б

на защиту выпускной квалификационной работы Тема

Декан факультета подпись декана «11» мая 2020 г.

**Справка об успеваемости**

Студент Фамилия Имя Отчество за время пребывания в МГТУ имени Н.Э. Баумана с 2017 г. по 2020 г. полностью выполнил учебный план со следующими оценками: отлично – [процент] %, хорошо – [процент] %, удовлетворительно – [процент] %.

Инспектор деканата от руки

**Отзыв руководителя выпускной квалификационной работы**

*Студент Фамилия И.О. в процессе выполнения ВКР проявил себя как ... Результаты, полученные в процессе реализации задания, позволили сделать вывод о ... целесообразности/нецелесообразности выбранных путей решения поставленной задачи, ... невозможности применения ... Работа выполнена автором самостоятельно, в полном объеме, в полном соответствии с заданием и календарным планом. Несмотря на сделанные замечания студент достоин «отличной» оценки... и присвоения звания бакалавр техники и технологий по направлению «Информатика и вычислительная техника».*

Руководитель ВКР \_\_\_\_\_ «\_\_\_\_» \_\_\_\_\_ 2020 г.  
(подпись) А.П. Соколов (ФИО) (дата)

Студент \_\_\_\_\_ «\_\_\_\_» \_\_\_\_\_ 2020 г.  
(подпись) И.О. Фамилия (ФИО) (дата)

## РЕФЕРАТ

выпускная квалификационная работа: 33 с., 1 рис., 0 табл., 13 источн.

CASE-СИСТЕМЫ, РАСПРЕДЕЛЁННЫЕ ВЫЧИСЛЕНИЯ, ГРАФООРИЕНТИРОВАННЫЙ ПОДХОД, СЛОЖНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ, ОПИСАНИЕ БИЗНЕС-ЛОГИКИ.

Данная работа посвящена разработке программного инструментария, позволяющего описывать и реализовать логику решения различных задач, требующих большого количества трудоёмких вычислений. При описании применяется т.н. графоориентированный подход, который позволяет пользователю задавать действия алгоритма или вычислительного метода в виде переходов между состояниями данных. Формируемое описание затем интерпретируется и выполняется с применением стандартных или пользовательских реализаций каждого из переходов.

Реализованные программные средства позволяют структурировать и ускорить разработку наукоёмкого программного обеспечения, применяемого при анализе больших объёмов данных и научно-технических исследованиях.

**Тип работы:** выпускная квалификационная работа.

**Тема работы:** *«Разработка компонентов графоориентированного программного каркаса для реализации сложных вычислительных методов».*

**Объект исследования:** подходы к описанию бизнес-логики в системах автоматизированной разработки программного обеспечения.

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

**Цели работы:** @Цель выполнения работы@

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



## СОКРАЩЕНИЯ

**API** – прикладной программный интерфейс (Application Programming Interface).

**DFD** – диаграмма потоков данных (Data Flow Diagram).

**GBSE** – графоориентированный подход к разработке программного обеспечения (graph based software engineering).

**JSON** – файловый формат для хранения структур данных (Javascript Object Notation).

**LCPD** – платформы малокодовой разработки (low-code development platforms).

**ПО** – программное обеспечение.

# СОДЕРЖАНИЕ

<b>СОКРАЩЕНИЯ</b> .....	9
<b>ВВЕДЕНИЕ</b> .....	11
<b>1 Постановка задачи</b> .....	22
1.1 Концептуальная постановка задачи . . . . .	22
1.2 Математическая постановка задачи (представляется в зависимости от задачи) . . . . .	22
<b>2 Вычислительный метод</b> .....	23
<b>3 Программная реализация</b> .....	24
3.1 Архитектура . . . . .	24
<b>4 Тестирование и отладка</b> .....	25
4.1 ... . . . .	25
<b>5 Вычислительный эксперимент</b> .....	26
5.1 ... . . . .	26
<b>6 Анализ результатов</b> .....	27
6.1 ... . . . .	27
<b>ЗАКЛЮЧЕНИЕ</b> .....	28
<b>Литература</b> .....	29
<b>ПРИЛОЖЕНИЯ</b> .....	32
<b>А</b> .....	32

## ВВЕДЕНИЕ

Современные научно-технические исследования зачастую включают в себя задачи, при решении которых требуется большое количество вычислений, для которых задействуются большие вычислительные мощности. К таким задачам относятся, например, задачи анализа, определения характеристик материалов или технических объектов, моделирования сложных динамических процессов. Как правило, для решения подобных задач применяется или разрабатывается специализированное программное обеспечение (далее – ПО).

Среди прочих применяются программные продукты, предоставляющие пользователю формальный язык описания математических выражений и его интерпретатор, выполняющий необходимые вычисления на машине пользователя. К таким системам относятся, например, Mathcad. Также стоит отметить системы специализирующиеся на символьной алгебре, такие, как Maple[1] и Wolfram Mathematica. В настоящее время данные программные комплексы поддерживают решение задач из различных областей математики, включающих в себя теорию графов, теорию множеств и т.д., предоставляют инструменты визуализации и анализа результатов. Все они позволяют выполнять математическое моделирование, в том числе, сложных технических объектов. При всех их преимуществах необходимость формулировать математические постановки решаемых задач (т.е. формировать математические модели, составлять системы уравнений и т.д.) остаётся за пользователем. Зачастую требуется решать множество задач с схожей постановкой, но с различными входными параметрами. Такая необходимость, например, возникает при решении задач оптимизации, где критерием является некоторая характеристика, получаемая в результате решения задачи анализа. Следовательно, целесообразны автоматизированные средства решения типовых задач анализа и моделирования.

Данные средства относятся к специализированному ПО, а потому при их разработке требуются глубокие познания в предметной области. Кроме того, важно, чтобы создаваемая кодовая база была рассчитана на дальнейшую поддержку, что предъявляет соответствующие требования к

структуре исходного кода и документации. Таким образом целесообразно применение некоторых средств, позволяющих организовать разработку программного обеспечения для решения задач моделирования и анализа и повысить его поддерживаемость.

В наши дни популярность приобретает применение т.н. научных систем управления потоком задач (англ. scientific workflow systems). Они предоставляют средства организации этапов решения вычислительной задачи и управления вычислительными ресурсами. Процесс работы с подобными системами состоит из 4 основных этапов:

- 1) составление описания операций обработки данных и зависимостей между ними;
- 2) распределение процессов обработки данных по вычислительным ресурсам;
- 3) выполнение обработки данных;
- 4) сбор и анализ результатов и статистики [2].

Примерами подобных систем могут служить Pegasus[3], Kepler[4] и pSeven[5]. Помимо инструментов загрузки пользовательских реализаций этапов решения задачи они, как правило, представляют библиотеку типовых действий и преобразований, таких, как считывание данных и их сохранение в файлы одного из поддерживаемых форматов, операции со строками, работы с базами данных, и т.д. На рисунке В.1 изображён пример описания некоторого процесса в системе Kepler.

Кроме того, для облегчения процесса разработки трудоёмкого ПО существуют т.н. платформы малокодовой разработки (англ. low-code development platforms, LCPD)[6]. В них, подобно системам управления потоком задач, логика разрабатываемого программного продукта описывается при помощи некоторого формального языка или с использованием графического редактора. От системы к системе подход к описаниям варьируется. Может применяться структурный подход, описывающий шаги алгоритма, или предметно-ориентированный, при котором описываются взаимодействующие сущности. Некоторые системы позволяют по созданному описанию генерировать готовые компоненты

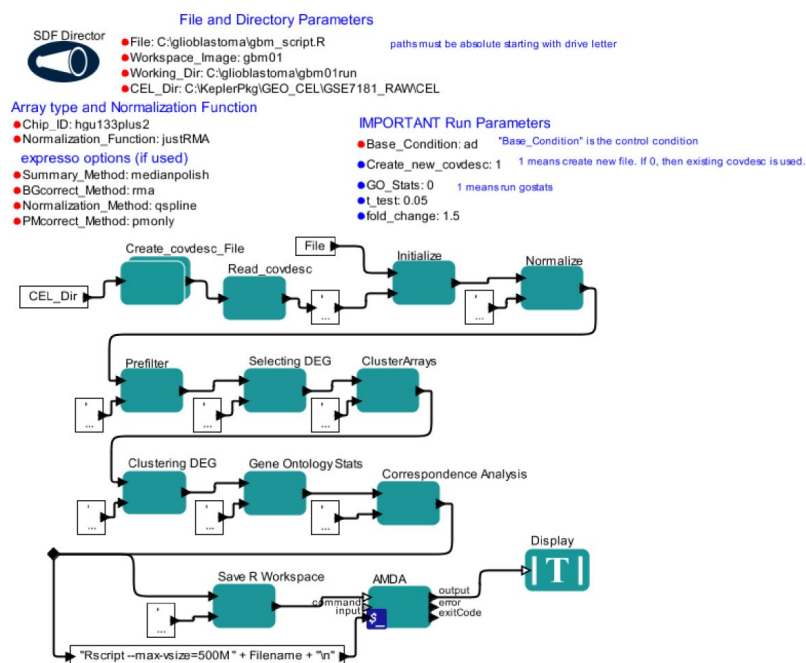


Рисунок В.1. Описание процесса обработки данных в системе Kepler

будущего программного продукта. Так платформа Codebots реализует предметно-ориентированный подход и по составленным UML-диаграммам взаимодействующих сущностей позволяет генерировать API, JSON-схемы данных и документацию[6]. Тем не менее, при реализации сложных вычислительных методов целесообразнее использовать структурный подход.

Одной из ключевых особенностей описанных технологических решений является выделение операций обработки данных в отдельные программные модули (функции, подпрограммы, скрипты). Как правило, при создании описаний алгоритмов в них используется следующий подход. Поскольку известно, что выходные данные одного программного модуля могут являться входными для одного или нескольких других модулей, можно сказать, что между ними формируются зависимости по входным и выходным данным. Тогда возможно составить такой ориентированный граф, описывающий общую логику алгоритма, в котором узлами являются операции обработки данных, а рёбрами – пути данных. Такой подход получил название “диаграммы потоков данных” (англ. Dataflow Diagram, DFD). При известных входных и выходных данных каждого модуля становится возможной их независимая разработка[7]. Таким образом, уменьшается объём работы по написанию исходных кодов, приходящийся на одного исследователя. Это в свою очередь облегчает отладку и

написание документации, что положительно сказывается на общем качестве реализуемого ПО.

!!! ----- WARNING ! MISSING PART ----- !!!

!!! Здесь нужен какой-то переход к тому, зачем может потребоваться вводить абстракцию над обрабатываемыми данными

!!! ----- !!!

Таким образом, в некоторых случаях может быть целесообразен такой подход к построению описания логики реализуемого решения, что в нём не указываются конкретные обрабатываемые данные. Последовательность выполнения отдельных этапов в таком случае должна задаваться явно. В предпринимательстве и управлении проектами подобный подход широко распространён и реализован в сетевых графиках. Сетевой график представляет собой ориентированный граф, в котором вершины – это события или состояния проекта, а рёбра – это работы. В работе [8] рассматривается применение идеи переходов между состояниями при описании логики вычислительных алгоритмов. Описанный подход получил название *graph-based software engineering (GBSE)*. Кроме того в указанной работе описана реализация GBSE в библиотеке *comsdk* для языка C++.

Был проведён сравнительный анализ программного каркаса *comsdk* с одной из реализаций DFD. В качестве такой реализации был рассмотрен программный комплекс *pSeven*, разработанный отечественной компанией DATADVANCE. Он направлен в первую очередь на решение конструкторских, оптимизационных задач и, помимо этого, задач анализа данных, что в первом приближении делает его аналогом *comsdk* по предметному назначению.

В терминах *pSeven*: графовое описание процесса решения задачи называется *расчетной схемой* (англ. *workflow*); узлам оргграфа поставлены в соответствие процессы обработки данных (используется термин *блоки*), а рёбра определяют *связи* между блоками и направления передачи данных между процессами [5]. При работе с *pSeven* используются следующие понятия:

- расчётная схема – формальное описание процесса решения некоторой задачи в виде ориентированного графа;
- блок – программный контейнер для некоторого процесса обработки данных, входные и выходные данные для которого задаются через порты (см. ниже);
- порт – переменная конкретного<sup>1</sup> типа, определённая в блоке и имеющая уникальное имя в его пределах;
- связь – направленное соединение типа “один к одному” между выходным и входным портами разных блоков.

С учётом данных понятий можно описать используемую методологию диаграмм потоков данных следующим образом. Расчётная схема содержит в себе набор процессов обработки данных (блоков), каждый из которых имеет (возможно, пустой) набор именованных входов и выходов (портов). Данные передаются через связи. Для избежания т.н. гонок данных (англ. data races) множественные связи с одним и тем же входным портом не поддерживаются. Для начала выполнения каждому блоку требуются данные на всех входных портах. Все данные на выходных портах формируются по завершении исполнения блока [5].

Результаты проведённого сравнения представлены в таблице В.2.

---

<sup>1</sup>Динамическая типизация не поддерживается.

Таблица В.2. Сравнительная таблица

№	Признак	pSeven	GBSE
1	Предметное назначение	Задачи оптимизации, анализ данных	Задачи автоматизированного проектирования, алгоритмизация сложных вычислительных методов, анализ данных
2	Принцип формирования графовых моделей	Узлы – блоки (процессы), рёбра – связи (направление передачи данных) [5].	Узлы – состояния данных, рёбра – переходы между состояниями, с указанием функций перехода [8].
3	Формат описания орграфа	Расчетная схема (в форме орграфа) сохраняется в двоичный файл закрытого формата с расширением .p7wf.	Графовая модель (определяет алгоритм проведения комплексных вычислений в форме орграфа) сохраняется в текстовом файле открытого формата, подготовленного на языке aDOT[9], являющегося “сужением” (частным случаем) известного формата DOT (Graphviz).



4	Файловая структура	Проект состоит из непосредственно файла проекта, в котором хранятся ссылки на созданные расчётные схемы и локальную базу данных, сами расчётные схемы, файлы с их входными данными, файлы отчётов, где сохраняются выходные данные последних расчётов и результаты их анализа.	Проект состоит из .aDOT файла с описанием графа, .aINI-файлов с описанием форматов входных данных, библиотек функций-обработчиков, функций-предикатов и функций-селекторов , файлов, куда записываются выходные данные.
5	особенности работы с входными и выходными данными графовых моделей	Входные данные должны быть указаны при настройках внешних входных портов расчётной схемы. Данные с выходных портов схемы сохраняются в локальной базе данных. Для их записи в файлы для обработки/анализа вне pSeven необходимо воспользоваться специально предназначенными для этого блоками.	Входные данные хранятся в файле в формате aINI[10], откуда считываются при запуске обхода графа [11]. Для записи выходных/промежуточных данных в файлы или базы данных необходимо добавить соответствующие функции-обработчики. Формат выходных данных не регламентирован.

6	Особенности передачи параметров между узлами графовых моделей	Данные между узлами передаются согласно определенным связям, которые на уровне выполнения создают пространство в памяти для ввода и вывода данных для выполняемых в отдельных процессах блоков. Транзитная передача данных, которые не изменяются в данном блоке, на выход невозможна.	Поскольку узлами графа являются состояния данных, существует возможность задействовать в расчётах только часть данных, оставляя их другую часть неизменной.
---	---	--	---

7	Поддержка ветвлений и циклов	<p>Присутствует. Достигается за счёт специальных управляющих блоков, которые отслеживают выполнение условий: для ветвления используется блок "Условие"(англ. condition), который перенаправляет данные на один из выходных портов в зависимости от выполнения описанного условия (подробнее см. [12]); Для реализации циклов в общем случае используются блоки "Цикл"(англ. loop)[13], но для некоторых задач существуют специализированные блоки, организующие логику работы цикла (например, блок "Оптимизатор"(англ. optimizer))</p>	Присутствует по умолчанию
---	------------------------------	---	---------------------------

8	Поддержка параллельной обработки данных	Присутствует. Блоки, входящие в состав различных ветвлений схемы могут быть выполнены параллельно, поскольку они не зависят друг от друга по используемым данным.	Присутствует. Существует возможность обойти различные ветвления графа одновременно.
9	Возможность выбрать из набора однотипных промежуточных результатов расчётов некоторые экземпляры и продолжить расчёт только для них;	Производится на этапе анализа результатов с помощью отчётов, где можно задать фильтрацию выходных данных согласно указанным критерия. В случае, если результаты являются промежуточными, расчётную схему приходится разбивать на части.	Планируется реализовать средство визуализации данных, которое в совокупности с автоматической генерацией форм ввода[11] позволят отбирать корректные результаты промежуточных вычислений во время обхода графовой модели.

10	Возможность доопределения значений входных данных в процессе обхода графа	Отсутствует	Частично реализована при помощи функций-обработчиков специального типа, создающих формы ввода
----	---	-------------	---

# **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.

## 2 Вычислительный метод

В разделе следует представить описание применяемого (планируемого к применению) вычислительного метода. Метод следует описывать с использованием математически строгих формулировок, не допускающих неоднозначности прочтения.

Обязательность представления: раздел представляется в зависимости от поставленной задачи. Объём: около 5 страниц.

## **3 Программная реализация**

### **3.1 Архитектура**

В разделе следует представить в форме ссылок применяемые (планируемые к применению) технологии разработки, включая языки программирования. Следует подробно описать предлагаемые алгоритмы, реализуемые в виде программ. Следует активно использовать графические способы представления информации: иерархий классов, реляционных моделей данных, графовые модели, диаграммы потоков данных, блок-схемы и прочие. Следует минимизировать текстовые нерепрезентативные способы описания программных объектов (например, в форме листингов).

Обязательность представления: раздел представляется в зависимости от постановки задачи. Объём: около 5 страниц.



## 4 Тестирование и отладка

### 4.1 ...

В разделе следует представить описания тестовых примеров, включая входные данные, принципы запуска и указать ожидаемый результат и фактически полученный.

Допускается включение скриншотов, однако, каждый должен быть подписан и представлено обоснование его включения в РПЗ.

Обязательность представления: раздел представляется в зависимости от постановки задачи.

Объём: около 4-5 страниц.

## 5 Вычислительный эксперимент

### 5.1 ...

В разделе следует представить описания каждого вычислительного эксперимента, включая указание особенностей их проведения, используемые программные средства, используемые исходные данные, принципы запуска с указанием ожидаемого и полученного результата.

Обязательно представление графических результатов в форме графиков, поверхностей.

Обязательность представления: раздел представляется в зависимости от постановки задачи.

Объём: объём не ограничен, но, как правило, не должен быть меньше 5-6 страниц.

## **6 Анализ результатов**

### **6.1 ...**

В разделе следует представить анализ полученных результатов, включая указание перспектив развития созданных научно-технических решений.

Обязательность представления: раздел обязателен.

Объём: объём не ограничен, но, как правило, не должен быть меньше 2 страниц.

## ЗАКЛЮЧЕНИЕ

В разделе следует представить выводы по работе в целом. Каждый вывод **не должен** быть банальным указанием факта реализации поставленных задач. Каждый вывод должен быть результатом проведенной работы в целом, включая результаты тестирования, вычислительных экспериментов и анализа результатов.

Обязательность представления: раздел обязателен.

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

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 The design of Maple: A compact, portable, and powerful computer algebra system / Bruce W. Char, Keith O. Geddes, W. Morven Gentleman [и др.] // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1983. T. 162 LNCS. С. 101 – 115.
- 2 Workflows and e-Science: An overview of workflow system features and capabilities / D. E., G. D., S. M. et al. // Future Generation Computer Systems. 2009. Vol. 25, no. 5. P. 528 – 540.
- 3 Pegasus in the cloud: Science automation through workflow technologies / Deelman E., Vahi K., Rynge M. [и др.] // IEEE Internet Computing. 2016. T. 20, № 1. С. 70 – 76.
- 4 Kepler: An extensible system for design and execution of scientific workflows / Altintas I., Berkley C., Jaeger E. [и др.]. Т. 16. 2004. С. 423 – 424.
- 5 Alexey M. Nazarenko Alexander A. Prokhorov. Hierarchical Dataflow Model with Automated File Management for Engineering and Scientific Applications // Procedia Computer Science. 2015. Т. 66. URL: <https://www.sciencedirect.com/science/article/pii/S1877050915034055?pes=vor>.
- 6 Low-code development and model-driven engineering: Two sides of the same coin? / Davide Di Ruscio, Dimitris Kolovos, Juan de Lara [и др.] // Software and Systems Modeling. 2022. Т. 21, № 2. С. 437 – 446.
- 7 Данилов А.М., Лапшин Э.В., Беликов Г.Г., Лебедев В.Б. Методологические принципы организации многопоточной обработки данных с распараллеливанием вычислительных процессов // Известия вузов. Поволжский регион. Технические науки. 2001. № 4. С. 26–34.
- 8 Соколов А.П. Першин А.Ю. Графоориентированный программный каркас для реализации сложных вычислительных методов // Программирование. 2018. № X.
- 9 Соколов А.П. Першин А.Ю. Описание формата данных aDOT (advanced DOT). 2020.

10 Соколов А.П. Описание формата данных aINI (advanced INI) [Электронный ресурс]. Облачный сервис SA2 Systems. [Офиц. сайт]. 2020. URL: <https://sa2systems.ru/nextcloud/index.php/f/403527>.


11 Соколов А.П Першин А.Ю. Программный инструментарий для создания подсистем ввода данных при разработке систем инженерного анализа // Программная инженерия. 2017. Т. 8, № 12. С. 543–555.

12 Condition - pSeven 6.31.1 User Manual [Электронный ресурс] [Офиц. сайт]. 2022. (дата обращения 07.03.2022). URL: <https://www.datadvance.net/product/pseven/manual/6.31.1/blocks/Condition.html>.

13 Расчётные схемы - Руководство пользователя pSeven 6.27 [Электронный ресурс] [Офиц. сайт]. 2021. Дата обращения: 15.11.2021. URL: <https://www.datadvance.net/product/pseven/manual/ru/6.27/workflow.html#workflow-links>.

## Выходные данные

*Тришин И.В.. Разработка компонентов графоориентированного программного каркаса для реализации сложных вычислительных методов по дисциплине «Модели и методы анализа проектных решений». [Электронный ресурс] — Москва: 2022. — 33 с. URL: <https://sa2systems.ru>: 88 (система контроля версий кафедры РК6)*

Постановка:  канд. физ.-мат. наук, Соколов А.П.

Решение и вёрстка:  студент группы РК6-81Б, Тришин И.В.

*2022, весенний семестр*

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

**АКТ**  
**проверки выпускной квалификационной работы**

Студент группы РК6-81Б

Фамилия Имя Отчество  
(Фамилия, имя, отчество)

Тема выпускной квалификационной работы: [Тема]

Выпускная квалификационная работа проверена, размещена в ЭБС «Банк ВКР» в полном объеме и соответствует / не соответствует требованиям, изложенным в Положении о порядке подготовки и защиты ВКР.  
ненужное зачеркнуть

Объем заимствования составляет            % текста, что с учетом корректного заимствования соответствует / не соответствует требованиям к ВКР  
ненужное зачеркнуть

← от руки

↑ бакалавра, специалиста, магистра  
от руки

**Нормоконтролёр**

Согласен:

**Студент**

Дата:

↑ от руки

↑ подписать  
С.В. Грошев  
(подпись) (ФИО)

↑ подписать  
И.О. Фамилия  
(подпись) (ФИО)

