

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ КемГУ
Дата и время: 2025-04-23 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436
Федеральное государственное бюджетное
образовательное учреждение высшего образования
"Кемеровский государственный университет"
Новокузнецкий институт (филиал)

Факультет информатики, математики и экономики
Кафедра математики, физики и математического моделирования

А.Д. Ульянов

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Методические рекомендации по выполнению внеаудиторной самостоятельной работы
для обучающихся по направлениям подготовки
01.04.02 Прикладная математика и информатика, профиль «Математическое
моделирование»*

Часть 1

Новокузнецк

2020

Ульянов А.Д.

Технология разработки программного обеспечения: методические рекомендации по выполнению внеаудиторной самостоятельной работы для студентов факультета информатики, математики и экономики, обучающихся по направлениям подготовки 01.04.02 Прикладная математика и информатика, профиль «Математическое моделирование»: в 2 ч. Ч 1. / Ю.С. Гаврилова; Новокузнецкий ин-т (фил.) Кемеров. гос. ун-та. – Новокузнецк : НФИ КемГУ, 2020 – 9 с.

В настоящих методических указаниях для студентов представлены материалы по выполнению внеаудиторной самостоятельной работы по дисциплине «Технология разработки программного обеспечения»: основные теоретические сведения в форме конспектов лекций с примерами решения типовых задач по темам «Методология проектирования программных продуктов», «Технологии разработки», «Средства информационной поддержки», «Внедрение и сопровождение программных продуктов». Также представлены банк задач для контрольных работ, методические рекомендации по решению и оформлению, оценивание работ в балльно-рейтинговой системе и список основной и дополнительной литературы.

Методические рекомендации предназначены для наиболее рациональной организации внеаудиторной самостоятельной работы студентов при подготовке к выполнению контрольных работ и теста.

Рекомендовано
на заседании кафедры
математики, физики и математического
моделирования
22 октября 2020г.
Заведующий кафедрой

 / Е.В. Решетникова

Ульянов А.Д., 2020
Федеральное государственное
бюджетное образовательное
учреждение высшего образования
«Кемеровский государственный
университет», Новокузнецкий
институт (филиал), 2020

Текст представлен в авторской редакции

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	4
1. МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ	6
1.1. Теоретические сведения	6
1.2. Банк заданий к лабораторной работе №1 по разделу «Средства информационной поддержки»	7
1.3. Особенности оценивания лабораторной работы №1 в балльно-рейтинговой системе	7
2. СПИСОК РЕКОМЕНДУЕМОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ	9

ПРЕДИСЛОВИЕ

Настоящие методические рекомендации адресованы студентам, получающим квалификацию магистр по направлению подготовки: 01.04.02 Прикладная математика и информатика, профиль «Математическое моделирование» и направлены на оказание помощи студентам в подготовке к выполнению лабораторных работ по темам «Средства информационной поддержки» дисциплины «Технология разработки программного обеспечения для математического моделирования».

Специалисту в области разработки программного обеспечения необходимо владеть не только практическими и теоретическими навыками кодирования, но и основами методологии проектирования и разработки ПО. Данный курс включает рассмотрение принципов agile (гибкий, проворный) касаясь работы над кодом программ небольших групп. Так же будет рассмотрена технология визуального программирования, реализованная в среде визуального программирования "Алгозит".

Agile software development (гибкая разработка ПО) - обобщение над рядом подходов, основанных на 4-х ценностях и 12-ти принципах манифеста гибкой разработки программного обеспечения.

Ценности манифеста:

1. Люди и взаимодействие важнее процессов и инструментов.
2. Работающий продукт важнее исчерпывающей документации.
3. Сотрудничество с клиентом важнее согласования условий контракта.
4. Готовность к изменениям важнее следования первоначальному плану.

Основные принципы:

1. Наивысшим приоритетом является удовлетворение потребностей клиента, благодаря регулярной и ранней поставке ценного программного обеспечения.
2. Изменение требований приветствуется, даже на поздних стадиях разработки.
3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.
4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.
5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.
6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.
7. Работающий продукт — основной показатель прогресса.
8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно.
9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.
10. Простота — искусство минимизации лишней работы — крайне необходима.
11. Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.
12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

Примеров различных методик, соответствующей данным принципам много, при этом подходы могут быть комбинированы на усмотрение лидера команды. Одним из примеров является "экстремальное программирование" (XP).

Основная идея подхода состоит в выведении полезных практик программирования на максимальный уровень. Подход реализуется соблюдением 12-ти приемов:

1. Разработка через тестирование (Test-driven development) - перед программированием функции для нее пишется подробный тест.
2. Игра в планирование - быстро сформировать приблизительный план работы и постоянно обновлять его.

3. Заказчик всегда рядом - заказчик фиксирует всякое отклонение от собственных потребностей.
4. Парное программирование - за одним компьютером работают 2 программиста (один сконцентрирован на написании кода, второй - проверяет код по мере его написания).
5. Непрерывная интеграция - все модули программы интегрируются по мере реализации, а не в пред релизных стадиях.
6. Рефакторинг - изменение внутренней структуры модуля без изменения интерфейса с целью облегчения понимания и чтения кода.
7. Частые небольшие релизы - продукт постоянно обновляется у пользователя (все недочеты и неточности выявляются потребителем сразу).
8. Простота проектирования - обеспечивает понимание разработчиками структуры всей программы.
9. Метафора системы - даёт команде представление о том, каким образом система работает в настоящее время, в каких местах добавляются новые компоненты, и какую форму они должны принять (иными словами - архитектура системы).
10. Коллективное владение кодом - любой разработчик может подавать идеи для любого модуля системы.
11. Стандарт оформления кода - еще один прием, облегчающий чтение чужого кода в проекте.
12. 40-часовая рабочая неделя - обеспечение разработчиков необходимым отдыхом, для поддержания творческой энергии.

Реализаций подходов к разработке ПО множество, но в целом они следуют принципам agile. Однако каждый руководитель проекта должен оценивать каждую команду и проект. Только таким образом возможно проводить мероприятия по разработке максимально эффективно, так как **универсальных решений не существует.**

1. МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ

1.1. Теоретические сведения

Следуя гибкой технологии программирования, при проектировании программного продукта необходимо обеспечивать простоту. Простота общей структуры необходима для вовлечения в процесс коррекции общих положений проекта каждого члена команды, работающего над модульными задачами. Однако невозможно представить современный проект слишком простой схемой. Поэтому проектировщику необходимо учитывать этот баланс.

Простота позволяет представить проект наглядно - понятной схемой или диаграммой. Новое слово в разработке архитектуры приложений это функционально-объектный подход, использующий визуальное программирование. Данная концепция разработки реализована в среде визуального программирования "Алгозит".

Среда программирования "Алгозит" является синтезом CASE-средств информационный поддержки программирования и, в то же время, средой времени выполнения. Наглядность архитектуры приложения обеспечена визуальными средствами программирования в виде функционально-объектных зависимостей функциональных объектов (конечных автоматов, или *алгома*тов). В то же время, разрабатываемая расчетная программа реализует объектно-ориентированный подход (каждый *алгома*т - это объект специального (функционального) класса).

Пример функционально-объектной программы в среде "Алгозит" представлен на рисунке 1:

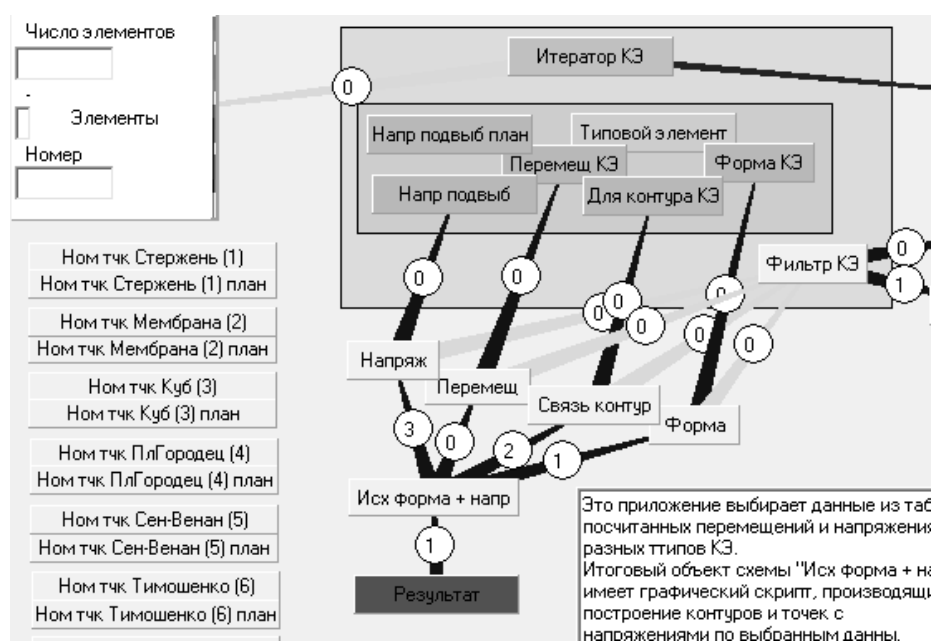


Рисунок 1 - Верхний уровень функционально-объектной программы

На рисунке 1 представлен верхний уровень функционально-объектных зависимостей программы. Панели "Напр. подвыб. план.", "Форма КЭ" и т.д. являются алгоматами скрытой схемы, представленной агрегатом "Типовой элемент", и доступны на текущем уровне в виде панелей - ссылок. Таким образом детали реализации внутренних процессов вычисления данных этих панелей скрыты на данном уровне, но доступны для редактирования при переходе на вложенный уровень.

Отмечу, что редактирование логики работы алгомата можно осуществлять не только коррекцией поведения на уровне класса C++, но и при установке в события панели скриптов на встроенном языке "Ядро".

Важной особенностью данного подхода является:

1. Возможно заблаговременное построение различных вариантов схем и их быстрая замена на этапе внедрения или сопровождения продукта. Это говорит о следовании принципу готовности к изменениям в любое время. 2. Визуальный подход позволяет любому разработчику в реальном времени наблюдать достаточно простую архитектуру приложения и, при необходимости, вносить в нее изменения.

3. Заказчик всегда может оценить взаимосвязи различных компонент системы и вовремя указать на недочеты или дать рекомендации.

4. Между различными модулями зависимости строятся во время разработки, что говорит о следовании принципу непрерывной интеграции.

Таким образом, предложенная методология разработки ПО соответствует с технической стороны методологии "экстремального программирования". С другой стороны, средства информационной поддержки являются одновременно средствами разработки, что комбинирует в себе задачи разработчика повышая производительность труда.

1.2. Банк заданий к лабораторной работе №1 по разделу «Средства информационной поддержки»

Реализовать итерационный алгоритм средствами визуального программирования для вычисления:

- 1) факториала числа,
- 2) корня квадратного,
- 3) синуса,
- 4) косинуса,
- 5) тангенса,
- 6) котангенса,
- 7) логарифма,
- 8) экспоненты.

1.3. Особенности оценивания лабораторной работы №1 в балльно-рейтинговой системе

Семестровая работа по разделу «Средства информационной поддержки» является промежуточной формой контроля знаний студентов и представляет собой выполнение определенных заданий с использованием компьютера. Она предназначена для проверки знаний студентов по учебной дисциплине «Технология разработки программного обеспечения программного обеспечения для математического моделирования», а также служит для закрепления полученных знаний, умений и навыков. В семестровой работе студентам предлагаются задачи, сформулированные на основании материала, изложенного в лекциях проработанного на практических занятиях или самостоятельно изученного студентами. Перед тем как приступить к выполнению работы, студентам следует ознакомиться с теоретическим материалом и разобраться с разобранными в нем типовыми задачами.

Варианты семестровой работы состоят из восьми заданий, которые представлены в разделе "Банк заданий к лабораторной работе №1 по разделу "Средства информационной поддержки". Варианты отличаются тематикой разрабатываемого алгоритма.

Система оценивания заданий семестровой работы №1 представлена в таблице 1.1.

Таблица 1.1 Оценивание семестровой работы №1 в БРС.

Критерии оценивания заданий	Количество баллов
Логично и последовательно выполнены все шаги решения,	20

рассуждения имеют четкое обоснование.	
Ход решения задания верный, но аргументация неполная.	15
Нарушена логическая цепочка рассуждений, решение неполное.	10
Максимальное количество баллов за лабораторную работу №1	20

Оформление контрольной работы должно соответствовать Правилам оформления учебных работ студентов¹, принятым в НФИ КемГУ.

2. СПИСОК РЕКОМЕНДУЕМОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ

Основная учебная литература:

1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189951> (дата обращения: 14.11.2020). – Режим доступа: по подписке.

Дополнительная учебная литература:

1. Лупин, С. А. Технологии параллельного программирования : учебное пособие / С.А. Лупин, М.А. Посыпкин. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2021. — 206 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0853-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189950> (дата обращения: 14.11.2020). – Режим доступа: по подписке.

2. Бедердинова, О. И. Программирование на языках высокого уровня : учеб. пособие / О.И. Бедердинова, Т.А. Минеева, Ю.А. Водовозова. — Москва : ИНФРА-М, 2019. — 159 с. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1044396> (дата обращения: 14.11.2020). – Режим доступа: по подписке.

3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г. Н. Федорова. — Москва : КУРС : ИНФРА-М, 2020. — 336 с. - (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1047718> (дата обращения: 15.11.2020). – Режим доступа: по подписке.

4. Черников, Б. В. Управление качеством программного обеспечения : учебник / Б.В. Черников. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2020. — 240 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0902-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1046280> (дата обращения: 15.11.2020). – Режим доступа: по подписке.

5. Ананьева, Т. Н. Стандартизация, сертификация и управление качеством программного обеспечения : учебное пособие / Т.Н. Ананьева, Н.Г. Новикова, Г.Н. Исаев. — Москва : ИНФРА-М, 2020. — 232 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014887-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1062373> (дата обращения: 15.11.2020). – Режим доступа: по подписке.

6. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем : учебное пособие / Л.Г. Гагарина. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2021. — 384 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0735-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1214882> (дата обращения: 15.11.2020). – Режим доступа: по подписке.

7. Черников, В. Н. Разработка мобильных приложений на C# для iOS и Android : практическое руководство / В. Н. Черников. - Москва : ДМК Пресс, 2020. - 188 с. - ISBN 978-5-97060-805-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1094956> (дата обращения: 15.11.2020). – Режим доступа: по подписке.

Литература для оформления учебных работ:

Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. СоинаКутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный.