

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»

**А. В. Снытников**

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА НАУКОЕМКОГО  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Учебно-методическое пособие по изучению дисциплины для студентов  
магистратуры по направлению  
09.04.01 Информатика и вычислительная техника

Калининград  
Издательство ФГБОУ ВО «КГТУ»  
2023

Рецензент:

кандидат педагогических наук, доцент кафедры прикладной информатики  
ФГБОУ ВО «Калининградский государственный технический университет»

Е. Ю. Заболотнова

**Снытников, А. В.**

Проектирование и разработка наукоемкого программного обеспечения: учебно-методическое пособие по изучению дисциплины для студентов магистратуры по направлению подготовки 09.04.01 Информатика и вычислительная техника / А. В. Снытников. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2023. – 21 с.

Учебно-методическое пособие является руководством по изучению дисциплины «Проектирование и разработка наукоемкого программного обеспечения» для студентов магистратуры по направлению подготовки 09.04.01 Информатика и вычислительная техника. Содержит характеристику дисциплины (цель и планируемые результаты изучения дисциплины, место дисциплины в структуре основной профессиональной образовательной программы), тематический план с описанием для каждой темы формы проведения занятия, вопросы для изучения, методические материалы к занятиям.

Табл. 4, список лит. – 5 наименований

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики 03 июля 2023 г., протокол № 13.

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в учебном процессе в качестве локального электронного методического материала методической комиссией Института цифровых технологий 5 июля 2023 г., протокол № 8

© Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Калининградский государственный  
технический университет» 2023 г.  
© Снытников А. В., 2023 г.

## Оглавление

Введение .....	4
1. ТЕМАТИЧЕСКИЙ ПЛАН .....	7
2. СОДЕРЖАНИЕ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ .....	8
Тема 1. Введение. Научное программное обеспечение .....	8
Тема 2. Модели жизненного цикла разработки ПО. ....	9
Тема 3. Специфика разработки научного программного обеспечения. ....	10
Тема 4. Основы метода конечных элементов.....	11
Тема 5. Программная реализация метода конечных элементов. ....	12
Тема 6. Решение задач методом конечных элементов .....	13
Тема 7. Основы работы с пакетом FEniCS.....	13
Тема 8. Решение задач гидродинамики с помощью пакета FEniCS.....	14
Тема 9. Решение задач теории упругости с помощью пакета FEniCS. ....	15
2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ .....	16
Тема 1. Введение. Научное программное обеспечение. ....	16
Тема 2. Модели жизненного цикла разработки ПО. ....	17
Тема 3. Специфика разработки научного программного обеспечения. ...	17
Тема 4. Основы метода конечных элементов.....	17
Тема 5. Программная реализация метода конечных элементов. ....	17
Тема 6. Решение задач методом конечных элементов.....	18
Тема 7. Основы работы с пакетом FEniCS.....	18
Тема 8. Решение задач гидродинамики с помощью пакета FEniCS.....	18
Тема 9. Решение задач теории упругости с помощью пакета FEniCS. ....	19
3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	19
Библиографический список.....	20

## Введение

Дисциплина «Проектирование и разработка наукоемкого программного обеспечения» является частью образовательной программы магистратуры по направлению подготовки 09.04.01 Информатика и вычислительная техника.

Целью освоения дисциплины «Проектирование и разработка наукоемкого программного обеспечения» является способность разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач.

В результате освоения дисциплины обучающийся должен:

знать: основные принципы построения современного программного обеспечения, типичные формы применения шаблонов проектирования;

уметь: строить модель программного обеспечения, применять основные паттерны проектирования, создавать эффективные сетевые и многопоточные приложения;

владеть: навыками применения современных методов проектирования программного обеспечения, современных методов оценки качества программного обеспечения.

Дисциплина опирается на компетенции, полученные при изучении дисциплин «Программирование», «Программная инженерия», «Высокоуровневые технологии программирования».

Промежуточная аттестация по дисциплине проводится в форме экзамена и защиты расчетно-графической работы.

К оценочным средствам для промежуточной аттестации, проводимой в форме экзамена во втором семестре, относятся экзаменационные вопросы.

Условия допуска для экзамена студентов:

Экзаменационные вопросы и задания. Экзаменационный билет содержит два теоретических вопроса и одно практическое задание. Список экзаменационных вопросов представлен в приложении А.

Экзаменационная оценка («отлично», «хорошо», «удовлетворительно» или «неудовлетворительно») является экспертной, зависит от уровня освоения студентом тем дисциплины (наличия и сущности ошибок, допущенных студентом при ответе на экзаменационные вопросы и решении экзаменационных заданий) и выставляется в соответствии с критериями, указанными в таблице 1.

Таблица 1 – Система оценок и критерии выставления оценки

Система оценок Критерий	2	3	4	5
	0-40 %	41-60 %	61-80 %	81-100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	«не зачтено»	«зачтено»		
<b>1 Системность и полнота знаний в отношении изучаемых объектов</b>	Обладает частичными и разрозненными знаниями, которые не может научно корректно связывать между собой (только некоторые из них может связывать между собой)	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полнотой знаний и системным взглядом на изучаемый объект
<b>2 Работа с информацией</b>	Не в состоянии находить необходимую информацию либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпретировать и систематизировать необходимую информацию в рамках поставленной задачи	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
<b>3 Научное осмысление изучаемого явления, процесса, объекта</b>	Не может делать научно-корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно-корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые, релевантные задаче данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые, релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи

Система оценок  Критерий	2	3	4	5
	0-40 %	41-60 %	61-80 %	81-100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	«не зачтено»	«зачтено»		
<b>4 Освоение стандартных алгоритмов решения профессиональных задач</b>	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

В данном учебно-методическом пособии представлены методические материалы по изучению дисциплины, включающие тематический план занятий с перечнем ключевых вопросов для каждой лекции, рекомендуемой литературой, методическими указаниями и вопросами для самоконтроля. Изложены методические рекомендации по подготовке к лабораторным занятиям. В Приложении А приведены экзаменационные вопросы.

## 1. ТЕМАТИЧЕСКИЙ ПЛАН

Тематический план для студентов очной формы обучения приведен в таблице 2.

Таблица 2 – Трудоёмкость освоения дисциплины во втором семестре по очной форме обучения

№ п/п	Раздел (модуль) дисциплины	Контактная работа с преподавателем					СРС	Подготовка и аттестация в период сессии
		ЛК	ЛР	ПР	РЭ	КА		
1	Введение. Научное программное обеспечение	3	3					
2	Модели жизненного цикла разработки ПО	3	3					
3	Специфика разработки научного программного обеспечения	3	3					
4	Основы метода конечных элементов	3	3					
5	Программная реализация метода конечных элементов	3	3					
6	Решение задач методом конечных элементов	3	3					
7	Основы работы с пакетом FEniCS	4	4					
8	Решение задач гидродинамики с помощью пакета FEniCS	4	4					
9	Решение задач теории упругости с помощью пакета FEniCS	4	4					
<b>ИТОГО:</b>		<b>30</b>	<b>30</b>		<b>4</b>	<b>3,25</b>	<b>79</b>	<b>33,75</b>
<b>Всего</b>		<b>180</b>						

## 2. СОДЕРЖАНИЕ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Приступая к изучению данного курса, рекомендуется придерживаться последовательности, соответствующей тематическому плану, приведенному в данном учебно-методическом пособии.

Приходя на очередную лекцию, необходимо обязательно предварительно просмотреть конспект предыдущей лекции. В случае вынужденного пропуска лекции нужно изучить ее самостоятельно, используя список рекомендованных литературных источников, приведенный в данном учебно-методическом пособии. Ответы на возникшие при этом вопросы можно решить с помощью рекомендованной литературы или на консультации у преподавателя.

Ниже приведен тематический план лекционных занятий.

### Тематический план лекционных занятий

Таблица 3 – План лекционных занятий

Тема 1	Введение. Научное программное обеспечение
Тема 2	Модели жизненного цикла разработки ПО
Тема 3	Специфика разработки научного программного обеспечения
Тема 4	Основы метода конечных элементов
Тема 5	Программная реализация метода конечных элементов
Тема 6	Решение задач методом конечных элементов
Тема 7	Основы работы с пакетом FEniCS
Тема 8	Решение задач гидродинамики с помощью пакета FEniCS
Тема 9	Решение задач теории упругости с помощью пакета FEniCS

#### **Тема 1. Введение. Научное программное обеспечение**

##### *Ключевые вопросы темы*

1. Цель и задачи дисциплины.
2. Место дисциплины в структуре образовательной программы.
3. Планируемые результаты освоения дисциплины.
4. Предмет курса проектирования и разработки ПО.
5. Основные понятия и определения методики разработки ПО. Ее разделы.
6. Отличие научного и научного ПО.
7. Краткое описание моделей жизненного цикла разработки ПО.
8. Примеры научного ПО.

##### *Источники*

[1, с. 7-11].



### *Методические рекомендации*

В начале изучения дисциплины «Проектирование и разработка наукоемкого ПО» необходимо понять цели и задачи изучения этой дисциплины, ее место в структуре образовательной программы, планируемые результаты освоения дисциплины.

Дисциплина состоит из трех разделов: «Модели жизненного цикла разработки ПО», «Метод конечных элементов» и «Программный пакет FEniCS». Изучение начинается с раздела «Модели жизненного цикла разработки ПО», а именно с основных понятий и определений.

Далее следует ознакомиться с основами метода конечных элементов. Изучение данного метода и опыт разработки ПО на его основе необходим для того, чтобы сведения о разработке ПО из первой части курса не выглядели как чисто теоретические и оторванные от практики, а, напротив, могли бы быть немедленно применены на практике. Далее в третьей части студенты познакомятся с пакетом прикладного ПО FEniCS, представляющим собой свободно распространяемый пакет на базе метода конечных элементов, предназначенный для решения научных и инженерных задач. Таким образом, студенты получают опыт использования современного наукоемкого ПО.

### *Вопросы для самоконтроля*

1. Что изучает дисциплина «Проектирование и разработка наукоемкого ПО»?
2. Из каких разделов она состоит?
3. Что изучает раздел «Модели разработки ПО»?
4. Что принято называть моделью разработки ПО?
5. В чем отличие научного и наукоемкого ПО?
6. В чем специфика разработки наукоемкого ПО?
7. Назовите основные модели разработки ПО.

## **Тема 2. Модели жизненного цикла разработки ПО**

### *Ключевые вопросы темы*

1. Жизненный цикл разработки ПО.
2. Водопадная модель жизненного цикла разработки ПО.
3. Итеративная водопадная модель жизненного цикла разработки ПО.
4. Спиральная модель жизненного цикла разработки ПО.
5. V-модель жизненного цикла разработки ПО.

### *Источники*

[1, с. 7-11, 12-27].

### *Методические рекомендации*

В рамках данной темы студентам необходимо изучить, что такое жизненный цикл разработки ПО, чем он характеризуется, как определяется его продолжительность и направление развития.

Следует разобраться, в чем причина возникновения различных моделей, как одна следует из другой, и для каких вариантов работы оптимальна та или другая модель, и в частности, какую из моделей следует использовать для разработки наукоемкого ПО.

### *Вопросы для самоконтроля*

1. Перечислите основные этапы классического жизненного цикла проекта и кратко охарактеризуйте их.
2. Классическая водопадная модель, фазы модели, её достоинства и недостатки.
3. Итеративная водопадная модель, фазы модели. Сравнение с классической водопадной моделью.
4. Дайте определение спиральной модели, опишите основные фазы этой модели.
5. Достоинства и недостатки спиральной модели.
6. Понятие рисков в рамках процесса разработки программного обеспечения.
7. Опишите работу с рисками в спиральной модели.
8. Общая характеристика V-модели, её этапы и связи между ними.
9. Область применения V-модели.

## **Тема 3. Специфика разработки наукоемкого программного обеспечения**

### *Ключевые вопросы темы*

1. Доля НИОКР в стоимости разработки.
2. Особенности жизненного цикла наукоемкого ПО.
3. Характерные черты проектов по разработке наукоемкого ПО.
4. Особенности управления разработкой наукоемкого ПО.
5. Выбор модели жизненного цикла для разработки наукоемкого ПО.

### *Источники*

[3, с. 4-9, 17-31, 90-103].

### *Методические рекомендации*

Перед началом изучения данной темы необходимо рассмотреть вопрос о затратах на разработку ПО и таким образом провести грань между наукоемким и ненаукоемким, а также научным ПО.

Далее следует рассмотреть жизненный цикл наукоемкого ПО, обращая внимание на его отличия от жизненного цикла обычного ПО.

Завершить изучение темы целесообразно рассмотрением особенностей управления разработкой наукоемкого ПО.

#### *Вопросы для самоконтроля*

1. Какая доля НИОКР в стоимости характерна для наукоемкого ПО?
2. Какая модель жизненного цикла разработки ПО оптимальна для наукоемкого ПО?
3. Что является основным фактором создания наукоемкого ПО?
4. Характерна ли процессная модель управления для проектов по разработке наукоемкого ПО?

### **Тема 4. Основы метода конечных элементов**

#### *Ключевые вопросы темы*

1. Постановка задачи в слабом смысле.
2. Матрица жесткости и вектор нагрузок.
3. Пробные функции.
4. Метод Галеркина.
5. Вариационный принцип.

#### *Источники*

[3, с. 31-46].

#### *Методические рекомендации*

Начинать изучение данной темы с рассмотрения основной идеи метода конечных элементов, а именно с постановки задачи в слабом смысле, изучить формирование, затем рассмотреть формирование матрицы жесткости и вектора нагрузок с учетом граничных условий.

Далее рассмотреть базисные и пробные функции, проанализировать метод Галеркина, сравнив его с методом Рэлея – Ритца. После этого следует перейти к изучению вариационного принципа и нахождению решения задачи, поставленной в слабом смысле.

#### *Вопросы для самоконтроля*

1. Что означает постановка задачи в слабом смысле?
2. Как перейти к постановке задачи в слабом смысле?
3. В чем преимущества постановки задачи в слабом смысле?
4. Какими свойствами должны обладать пробные функции?
5. Чем пробные функции отличаются от базисных функций?
6. Что такое вариационный принцип?
7. Как находится численное решение вариационной задачи?

## **Тема 5. Программная реализация метода конечных элементов**

### *Ключевые вопросы темы*

1. Общая структура программы.
2. Входные данные.
3. Сборка матриц и векторов конечных элементов.
4. Применение ограничений.

### *Источники*

[3, с. 51-69].

### *Методические рекомендации*

В рамках данной темы происходит освоение метода конечных элементов (МКЭ) через его программную реализацию на языке Python. Поэтому при рассмотрении данной темы следует меньше внимания уделять теоретическим вопросам, в большей степени сосредоточиться на получении конкретного программного результата в каждом случае.

Данный раздел начинается с формата входных данных, которые в дальнейшем сохранят свою структуру в любых вариантах реализации МКЭ, затем изучается структура связей между элементами и сборка матрицы жесткости и вектора нагрузок. После этого рассматривается применение ограничений (задание граничных условий). Заканчивается изучение темы рассмотрением базовых примеров программ для МКЭ.

Начать рекомендуется с изучения поступательного движения.

### *Вопросы для самоконтроля*

1. Какова общая структура программы в методе конечных элементов?
2. Какие данные должны быть поданы на вход программе, реализующей МКЭ?
3. Что такое матрица жесткости?
4. Как выполняется сборка матрицы жесткости?
5. Что такое вектор нагрузок?
6. Какую роль играет вектор нагрузок в задачах, не связанных с прочностью?
7. Что такое степени свободы?
8. Какую роль играют ограничения?
9. Как выполняется применение ограничений?
10. Какую структуру обычно имеет матрица жесткости?

## **Тема 6. Решение задач методом конечных элементов**

### *Ключевые вопросы темы*

1. Уравнение Пуассона.
2. Линейный треугольный элемент.
3. Билинейный прямоугольный элемент.
4. Формула Гаусса – Кристоффеля.
5. Интеграл по границам.
6. Нестационарные задачи.
7. Методы интегрирования по времени.
8. Матрица масс.
9. Уравнение диффузии.

### *Источники*

[3, с. 83-103, 170-176].

### *Методические рекомендации*

Рассмотрение данной темы необходимо начинать с уравнения Пуассона, которое будет основой для изучения отдельных форм конечных элементов, а также формулы Гаусса - Кристоффеля для вычисления интегралов по конечным элементам там, где их не удастся взять явно.

Далее рассматривается использование МКЭ в задачах с зависимостью от времени, расчет матрицы масс и начальных условий. Этот материал изучается на примере уравнения диффузии, метод решения которого также строится на основе рассмотренного выше уравнения Пуассона.

### *Вопросы для самоконтроля*

1. Как выглядит постановка задачи в слабом смысле для уравнения Пуассона?
2. Как устроен линейный треугольный элемент?
3. Формула Гаусса – Кристоффеля.
4. Как вычисляется матрица масс?
5. Как выполняется решение нестационарных задач на основе МКЭ?
6. Как определяется правильность решения уравнения диффузии?

## **Тема 7. Основы работы с пакетом FEniCS**

### *Ключевые вопросы темы*

1. Основы.
2. Нелинейные задачи.
3. Уравнение Пуассона.
4. Задачи, зависящие от времени.
5. Уравнение диффузии.

6. Создание более сложных областей.
7. Решение задач с различными материалами.
8. Дополнительные примеры.

#### *Источники*

[4, с. 1-66].

#### *Методические рекомендации*

Изучение данной темы необходимо начинать с изучения выполнения средствами пакета FEniCS тех же действий, которые студенты выполняли вручную в ходе изучения двух предшествующих тем, а именно задания сетки, базисных функций, постановки задачи в слабом смысле понятий и интегрирования по времени относительного, переносного и абсолютного движения.

#### *Вопросы для самоконтроля*

1. Как в пакете FEniCS задается сетка?
2. Как задать постановку задачи в слабом смысле?
3. Что такое граничные условия Робина?
4. Как быстро визуализировать сетку, решение, поток и т. д.?
5. Как работать с переменными коэффициентами?

### **Тема 8. Решение задач гидродинамики с помощью пакета FEniCS**

#### *Ключевые вопросы темы*

1. Физический смысл уравнений Навье – Стокса.
2. Краевая задача для уравнений Навье – Стокса.
3. Константы, характеризующие течение жидкости или газа.
4. Постановка задачи в слабом смысле.
5. Реализация на FEniCS.

#### *Источники*

[6, с. 56-73].

#### *Методические рекомендации*

Начать погружение в данную тему необходимо с краткого изучения уравнений Навье – Стокса.

Далее следует разобрать граничные условия и основные коэффициенты этих уравнений. Особое внимание обратить на безразмерные гидродинамические параметры – число Маха, Рейнольдса и отличие сжимаемого и несжимаемого течения.

### *Вопросы для самоконтроля*

1. В чем особенности краевой задачи для уравнений Навье – Стокса?
2. На какие коэффициенты нужно обращать особое внимание?
3. Что характеризует число Маха? Как оно применимо к проверке качества расчетов?
4. Что характеризует число Рейнольдса? Как оно применимо к проверке качества расчетов?
5. Как в FEniCS создается функциональное пространство?
6. Как в FEniCS создается набор пробных функций?

## **Тема 9. Решение задач теории упругости с помощью пакета FEniCS**

### *Ключевые вопросы темы*

1. Физический смысл уравнений упругости
2. Краевая задача для уравнений упругости
3. Константы, характеризующие течение, прочность и упругость твердого тела?
4. Постановка задачи в слабом смысле.
5. Реализация на FEniCS.

### *Источники*

[с. 50-55].

### *Методические рекомендации*

Начать погружение в данную тему необходимо с краткого изучения уравнений теории упругости.

Далее следует разобрать граничные условия и основные коэффициенты этих уравнений. Особое внимание обратить на параметры – модуль Юнга, коэффициент Пуассона и эволюцию тензора напряжений.

### *Вопросы для самоконтроля*

1. В чем особенности краевой задачи для уравнений линейной упругости?
2. На какие коэффициенты нужно обращать особое внимание?
3. Что характеризует модуль Юнга?
4. Как визуализировать эволюцию тензора напряжений?
5. Как в FEniCS создается сетка в прямоугольной области?
6. Как в FEniCS создается и вычисляется оператор дивергенции и градиента?

## 2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

Практические занятия проводятся с целью формирования у студентов знаний, умений и навыков, а также компетенций в соответствии с требованиями ФГОС ВО.

Еще одной важной задачей практических занятий является подготовка к выполнению требуемой расчетно-графической работы. В ходе практических занятий студент поэтапно решает задачи, похожие на предлагаемые в расчетно-графической работе.

Ниже представлен краткий план практических занятий с основными вопросами. Подготовку к практическим занятиям можно осуществлять с помощью рекомендованных литературных источников. Более подробно практические занятия рассмотрены в соответствующем методическом пособии.

### Тематический план практических занятий

Таблица 4 – План практических занятий

Тема 1	Введение. Научное программное обеспечение
Тема 2	Модели жизненного цикла разработки ПО
Тема 3	Специфика разработки научного программного обеспечения
Тема 4	Основы метода конечных элементов
Тема 5	Программная реализация метода конечных элементов
Тема 6	Решение задач методом конечных элементов
Тема 7	Основы работы с пакетом FEniCS
Тема 8	Решение задач гидродинамики с помощью пакета FEniCS
Тема 9	Решение задач теории упругости с помощью пакета FEniCS

### Тема 1. Введение. Научное программное обеспечение

#### *Ключевые вопросы темы*

1. Цель и задачи дисциплины.
2. Место дисциплины в структуре образовательной программы.
3. Планируемые результаты освоения дисциплины.
4. Предмет курса проектирования и разработки ПО.
5. Основные понятия и определения методики разработки ПО. Ее разделы.
6. Отличие научного и научного ПО.
7. Краткое описание моделей жизненного цикла разработки ПО.
8. Примеры научного ПО.



## *Источники*

[6, с. 27, 28].

### **Тема 2. Модели жизненного цикла разработки ПО**

#### *Ключевые вопросы темы*

1. Жизненный цикл разработки ПО.
2. Водопадная модель жизненного цикла разработки ПО.
3. Итеративная водопадная модель жизненного цикла разработки ПО.
4. Спиральная модель жизненного цикла разработки ПО.
5. V-модель жизненного цикла разработки ПО.

#### *Источники*

[1, с. 7-11, 12-27].

### **Тема 3. Специфика разработки наукоемкого программного обеспечения**

#### *Ключевые вопросы темы*

1. Доля НИОКР в стоимости разработки.
2. Особенности жизненного цикла наукоемкого ПО.
3. Характерные черты проектов по разработке наукоемкого ПО.
4. Особенности управления разработкой наукоемкого ПО.
5. Выбор модели жизненного цикла для разработки наукоемкого ПО.

#### *Источники*

[3, с. 4-9, 17-31, 90-103].

### **Тема 4. Основы метода конечных элементов**

#### *Ключевые вопросы темы*

1. Постановка задачи в слабом смысле.
2. Матрица жесткости и вектор нагрузок.
3. Пробные функции.
4. Метод Галеркина.
5. Вариационный принцип

#### *Источники*

[3, с. 31-46].

### **Тема 5. Программная реализация метода конечных элементов**

#### *Ключевые вопросы темы*

1. Общая структура программы.
2. Входные данные.
3. Сборка матриц и векторов конечных элементов.
4. Применение ограничений.

### *Источники*

[3, с. 51-69].

## **Тема 6. Решение задач методом конечных элементов**

### *Ключевые вопросы темы*

1. Уравнение Пуассона.
2. Линейный треугольный элемент.
3. Билинейный прямоугольный элемент.
4. Формула Гаусса – Кристоффеля.
5. Интеграл по границам.
6. Нестационарные задачи.
7. Методы интегрирования по времени.
8. Матрица масс.
9. Уравнение диффузии.

### *Источники*

[3, с. 83-103, 170-176].

## **Тема 7. Основы работы с пакетом FEniCS**

### *Ключевые вопросы темы*

1. Основы
2. Нелинейные задачи
3. Уравнение Пуассона
4. Задачи, зависящие от времени
5. Уравнение диффузии
6. Создание более сложных областей
7. Решение задач с различными материалами
8. Дополнительные примеры

### *Источники*

[4, с. 1-66].

## **Тема 8. Решение задач гидродинамики с помощью пакета FEniCS**

### *Ключевые вопросы темы*

1. Физический смысл уравнений Навье – Стокса.
2. Краевая задача для уравнений Навье – Стокса.
3. Константы, характеризующие течение жидкости или газа.
4. Постановка задачи в слабом смысле.
5. Реализация на FEniCS.

[5, с. 56-73].

## **Тема 9. Решение задач теории упругости с помощью пакета FEniCS**

### *Ключевые вопросы темы*

1. Физический смысл уравнений упругости.
2. Краевая задача для уравнений упругости.
3. Константы, характеризующие течение, прочность и упругость твердого тела.
4. Постановка задачи в слабом смысле.
5. Реализация на FEniCS.

### **3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

Содержание самостоятельной работы студентов очной и заочной форм обучения определяется рабочей учебной программой.

Наряду с проработкой лекционного материала и подготовкой к практическим занятиям студент обязан выполнить расчетно-графическую работу.

Расчетно-графическая работа по курсу выполняется с целью закрепления полученных в ходе обучения знаний и приобретения практических навыков решения типовых задач. Выполнение работы требует предварительного изучения соответствующего теоретического материала [1, 2].

При выполнении расчетно-графической работы следует придерживаться следующих правил:

1. Условия задачи должны полностью соответствовать варианту, выданному преподавателем.
2. Условия задач должны быть переписаны полностью.
3. Решения задач необходимо сопровождать пояснениями и подробными вычислениями.
4. При вычислении каждой величины надо указывать ее название, формулу, по которой она определяется. После записи формулы с буквенными обозначениями подставляются в таком же порядке их численные значения.

Расчетно-графическую работу можно начинать выполнять сразу же после прослушивания необходимого теоретического материала на лекциях. Настоятельно рекомендуется показывать преподавателю поэтапное решение задач вместо того, чтобы сдавать сразу полностью выполненную работу, в целях сокращения возможных исправлений.

Защитить РГР необходимо до начала экзаменационной сессии. Осуществляется путем собеседования или решения задачи, аналогичной выполненной в РГР, но в сокращенном и облегченном объеме. Для подготовки к защите можно использовать литературу, приведенную в данном учебно-методическом пособии [1, 2].

### **Библиографический список**

1. Перл, И. А. Введение в методологию программной инженерии: учебное пособие / И. А. Перл, О. В. Калёнова. – Санкт-Петербург: Университет ИТМО, 2019. – 53 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&i d=566776> (дата обращения: 24.02.2023). – Текст: электронный.

2. Доррер, Г. А. Методология программной инженерии: учебное пособие / Г. А. Доррер. – Красноярск: СибГУ им. академика М. Ф. Решетнёва, 2021. – 190 с. – Режим доступа: для авториз. пользователей. – Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/195097> (дата обращения: 24.02.2023). – Текст: электронный.

3. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие / Т. М. Зубкова. – Санкт-Петербург: Лань, 2022. – 324 с. – Режим доступа: для авториз. пользователей. – Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/206882> (дата обращения: 24.03.2023). – ISBN 978-5-8114-3842-6. – Текст: электронный.

4. Дукельский, К. В. Управление качеством программного обеспечения: учебное пособие / К. В. Дукельский, И. Б. Бондаренко. – Санкт-Петербург: СПбГУТ им. М. А. Бонч-Бруевича, 2021. – 52 с. – Режим доступа: для авториз. пользователей. – Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/279632> (дата обращения: 24.03.2023). – Текст: электронный.

5. Романов, Е. Л. Программная инженерия: учебное пособие / Е. Л. Романов. – Новосибирск: НГТУ, 2017. – 395 с. – Режим доступа: для авториз. пользователей. – Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/118221> (дата обращения: 24.02.2023). – ISBN 978- 5-7782-3455-0. – Текст: электронный.

Локальный электронный методический материал

Алексей Владимирович Снытников

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА НАУКОЕМКОГО  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Редактор М. А. Дмитриева*

Уч.-изд. л. 1,0. Печ. л. 1,3.

Издательства федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Калининградский государственный технический университет».  
236022, Калининград, Советский проспект, 1