

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Луганский национальный университет имени Владимира Даля»

Институт технологий и инженерной механики  
Кафедра технологии машиностроения и инженерного консалтинга

УТВЕРЖДАЮ

Директор института технологий и  
инженерной механики

Могильная Е.П.

«12» 09 2023 г.



## РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

### «РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

По направлению подготовки 15.03.01 Машиностроение

Профиль: «Технологии прототипирования машиностроительных объектов»

## Лист согласования РПУД

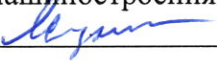
Рабочая программа учебной дисциплины «Разработка прикладного программного обеспечения» по направлению подготовки 15.03.01 Машиностроение. – 23 с.

Рабочая программа учебной дисциплины «Разработка прикладного программного обеспечения» составлена с учетом Федерального государственного образовательного стандарта высшего образования по направлению подготовки 15.03.01 Машиностроение, утвержденного приказом Министерства науки и высшего образования Российской Федерации от «9» августа 2021 года № 727.

### СОСТАВИТЕЛЬ:

канд. техн. наук, доцент Киреев И.Ю.

Рабочая программа дисциплины утверждена на заседании кафедры технологии машиностроения и инженерного консалтинга « 5 » 09 2023 года, протокол № 1

Заведующий кафедрой технологии машиностроения  
и инженерного консалтинга  Ясуник С.Н.

Переутверждена: «     »     20    г., протокол №    

Рекомендована на заседании учебно-методической комиссии института технологий и инженерной механики « 12 » 09 2023 года, протокол № 1

Председатель учебно-методической комиссии  
института технологий и инженерной механики  Ясуник С.Н.

© Киреев И.Ю., 2023 год

© ФГБОУ ВО «ЛГУ им. В. Даля», 2023 год

## Структура и содержание дисциплины

### 1. Цели и задачи дисциплины, ее место в учебном процессе

Цель изучения дисциплины – формирование у будущих инженеров-машиностроителей комплекса знаний, умений и навыков для разработки высокопроизводительного и системно-зависимого программного обеспечения на языках С и С++, решающего задачи расчета, моделирования и управления в области прототипирования и автоматизированного машиностроения.

Задачи: Изучить фундаментальные принципы процедурного и объектно-ориентированного программирования на языках С и С++, освоить работу с памятью, указателями, низкоуровневыми структурами данных, сформировать навыки создания программ для инженерных расчетов, включая численные методы и обработку данных.

### 2. Место дисциплины в структуре ООП ВО.

Дисциплина «Разработка прикладного программного обеспечения» входит в часть, формируемую участниками образовательных отношений.

Содержание дисциплины является логическим продолжением содержания дисциплин «Информатики и информационные технологии», «Компьютерная и инженерная графика», «Основы CAD/CAM систем», «Компьютерное моделирование объектов профессиональной деятельности» и служит основой для написания выпускной квалификационной работы.

### 3. Требования к результатам освоения содержания дисциплины

| Код и наименование компетенции                                                                            | Индикаторы достижений компетенции (по реализуемой дисциплине)                           | Перечень планируемых результатов                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| ОПК-14. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ОПК-14.1. Разрабатывает алгоритмы, пригодные для практического применения.              | Знать: Принципы объектно-ориентированного программирования                                                                                       |
|                                                                                                           |                                                                                         | Уметь: Разрабатывать алгоритмы решения инженерных задач                                                                                          |
|                                                                                                           |                                                                                         | Владеть методами объектно-ориентированного проектирования                                                                                        |
|                                                                                                           | ОПК-14.2. Разрабатывает компьютерные программы, пригодные для практического применения. | Знать: Структуру и возможности стандартной библиотеки шаблонов, методы создания графических интерфейсов, основы работы с файлами и базами данных |
|                                                                                                           |                                                                                         | Уметь: Создавать классы для описания объектов машиностроения, использовать STL для работы с данными, проводить отладку и тестирование программ   |
|                                                                                                           |                                                                                         | Владеть: Навыками программирования на С++, технологиями создания прикладного ПО, навыками работы со средами разработки.                          |

## 4. Структура и содержание дисциплины

### 4.1. Объем учебной дисциплины и виды учебной работы

| Вид учебной работы                                               | Объем часов (зач. ед.) |                      |
|------------------------------------------------------------------|------------------------|----------------------|
|                                                                  | Очная форма            | Заочная форма        |
| Общая учебная нагрузка (всего)                                   | 144<br>(4,0 зач. ед)   | 144<br>(4,0 зач. ед) |
| Обязательная аудиторная учебная нагрузка (всего)<br>в том числе: | 48                     | 8                    |
| Лекции                                                           | 24                     | 4                    |
| Семинарские занятия                                              | -                      | -                    |
| Практические занятия                                             | 24                     | 4                    |
| Лабораторные работы                                              | -                      | -                    |
| Курсовой проект                                                  | -                      | -                    |
| Другие формы и методы организации образовательного процесса      | -                      | -                    |
| Самостоятельная работа студента (всего)                          | 96                     | 136                  |
| Форма аттестации                                                 | экзамен                | экзамен              |

### 4.2. Содержание разделов дисциплины

**Тема 1. Введение в C++. Особенности языка. Среда разработки.**

Основные особенности. Популярные среды разработки (IDE).

**Тема 2. Базовые конструкции: типы данных, операторы, функции.**

Основные типы данных. Операторы. Функции.

**Тема 3. Указатели и ссылки. Динамическая память.**

Указатели. Ссылки. Динамическая память. Умные указатели (modern C++).

**Тема 4. Пространства имен. Препроцессор.**

Пространства имен. Препроцессор.

**Тема 5. Работа с файлами и потоками ввода-вывода.**

Работа с файлами. Потоки ввода-вывода. Базовый ввод-вывод.

**Тема 6. Классы и объекты. Инкапсуляция.**

Базовый класс. Специальные методы класса.

**Тема 7. Наследование и полиморфизм.**

Наследование. Полиморфизм. Абстрактные классы.

**Тема 8. Шаблоны функций и классов.**

Шаблоны функций. Шаблоны классов.

**Тема 9. Стандартная библиотека шаблонов (STL).**

Контейнеры. Алгоритмы и итераторы.

**Тема 10. Обработка исключений.**

Базовые исключения. Исключения в конструкторах.

**Тема 11. Создание графических интерфейсов (Qt Framework.)**

Базовая структура Qt приложения.

**Тема 12. Работа с базами данных в C++.**

Использование SQLite.

**Тема 13. Многопоточное программирование.**

Потоки и мьютексы. Асинхронное программирование.

#### **Тема 14. Оптимизация и профилирование кода.**

Основные принципы оптимизации.

#### **4.3. Лекции**

| №<br>п/п      | Название темы                                        | Объем часов    |                  |
|---------------|------------------------------------------------------|----------------|------------------|
|               |                                                      | Очная<br>форма | Заочная<br>форма |
| 1             | Введение в C++. Особенности языка. Среда разработки  | 2              | 0,5              |
| 2             | Базовые конструкции: типы данных, операторы, функции | 2              | 0,5              |
| 3             | Указатели и ссылки. Динамическая память              | 2              | 0,5              |
| 4             | Пространства имен. Препроцессор                      | 2              | 0,5              |
| 5             | Работа с файлами и потоками ввода-вывода             | 2              | 0,5              |
| 6             | Классы и объекты. Инкапсуляция                       | 2              | 0,5              |
| 7             | Наследование и полиморфизм                           | 2              | 0,5              |
| 8             | Шаблоны функций и классов                            | 2              | 0,5              |
| 9             | Стандартная библиотека шаблонов (STL)                | 2              | 0,5              |
| 10            | Обработка исключений                                 | 2              | 0,5              |
| 11            | Создание графических интерфейсов (Qt Framework)      | 1              | 0,5              |
| 12            | Работа с базами данных в C++                         | 1              | 0,5              |
| 13            | Многопоточное программирование                       | 1              | 1                |
| 14            | Оптимизация и профилирование кода                    | 1              | 1                |
| <b>Итого:</b> |                                                      | <b>24</b>      | <b>8</b>         |

#### **4.4. Практические работы**

| №<br>п/п      | Название темы                                     | Объем часов    |                  |
|---------------|---------------------------------------------------|----------------|------------------|
|               |                                                   | Очная<br>форма | Заочная<br>форма |
| 1             | Система расчета параметров режущего инструмента   | 2              | 1                |
| 2             | Система проектирования литейной оснастки          | 2              | 1                |
| 3             | Программа анализа допусков и посадок              | 2              | 1                |
| 4             | Система планирования раскроя листового материала  | 2              | 1                |
| 5             | Модуль анализа технологичности конструкций        | 2              | 1                |
| 6             | Система управления проектами прототипирования     | 2              | 1                |
| 7             | Программа расчета режимов 3D-печати               | 2              |                  |
| 8             | Система контроля качества деталей                 | 2              | 1                |
| 9             | Модуль симуляции литейных процессов               | 4              | 0,5              |
| 10            | Интегрированная система быстрого прототипирования | 4              | 0,5              |
| <b>Итого:</b> |                                                   | <b>24</b>      | <b>8</b>         |

#### **4.5. Лабораторные работы**

Лабораторные работы учебным планом не предусмотрены.

#### **4.6. Самостоятельная работа студентов**

| №<br>п/п | Название темы                                        | Вид СРС                         | Объем часов    |                  |
|----------|------------------------------------------------------|---------------------------------|----------------|------------------|
|          |                                                      |                                 | Очная<br>форма | Заочная<br>форма |
| 1        | Введение в C++. Особенности языка. Среда разработки. | Поиск, анализ, структурирование | 4              | 9                |

|        |                                                       |                                                         |    |     |
|--------|-------------------------------------------------------|---------------------------------------------------------|----|-----|
| 2      | Базовые конструкции: типы данных, операторы, функции. | и изучение информации по темам.<br>Подготовка к зачету. | 4  | 9   |
| 3      | Указатели и ссылки. Динамическая память.              |                                                         | 4  | 9   |
| 4      | Пространства имен. Препроцессор.                      |                                                         | 6  | 9   |
| 5      | Работа с файлами и потоками ввода-вывода.             |                                                         | 6  | 9   |
| 6      | Классы и объекты. Инкапсуляция.                       |                                                         | 6  | 9   |
| 7      | Наследование и полиморфизм.                           |                                                         | 6  | 9   |
| 8      | Шаблоны функций и классов.                            |                                                         | 6  | 9   |
| 9      | Стандартная библиотека шаблонов (STL).                |                                                         | 6  | 9   |
| 10     | Обработка исключений.                                 |                                                         | 6  | 9   |
| 11     | Создание графических интерфейсов (Qt Framework).      |                                                         | 6  | 9   |
| 12     | Работа с базами данных в C++.                         |                                                         | 6  | 11  |
| 13     | Многопоточное программирование.                       |                                                         | 6  | 11  |
| 14     | Оптимизация и профилирование кода.                    |                                                         | 6  | 11  |
| Итого: |                                                       |                                                         | 76 | 132 |

**4.7. Курсовые проекты.** Учебным планом не предусмотрено выполнение курсового проекта.

## 5. Образовательные технологии

Для достижения планируемых результатов освоения дисциплины «Разработка прикладного программного обеспечения» используются следующие образовательные технологии:

- традиционные объяснительно-иллюстративные технологии, которые обеспечивают доступность учебного материала для большинства студентов, системность, отработанность организационных форм и привычных методов, относительно малые затраты времени;
- информационно-коммуникационная технология, в том числе визуализация, создание электронных учебных материалов;
- использование электронных образовательных ресурсов при подготовке к лекциям, практическим и лабораторным занятиям;
- технология проблемного обучения, в том числе в рамках разбора проблемных ситуаций;
- технология развивающего обучения, в том числе постановка и решение задач от менее сложных к более сложным, развивающих компетенции студентов.

В рамках перечисленных технологий основными методами обучения являются: работа в команде; самостоятельная работа; проблемное обучение.

## 6. Учебно-методическое и информационное обеспечение дисциплины:

### а) основная литература:

1. Страуструп Б. Язык программирования C++. - М.: Вильямс, 2022
2. Прата С. Язык программирования C++. Лекции и упражнения. - М.: Вильямс, 2021

3. Шлее М. Qt 5.10. Профессиональное программирование на C++. - СПб.: БХВ-Петербург, 2020

**б) дополнительная литература:**

1. Липпман С. Язык программирования C++. Базовый курс. - М.: Вильямс, 2022

2. Эккель Б. Философия C++. Введение в стандартный C++. - СПб.: Питер, 2021

**в) Интернет-ресурсы:**

Министерство образования и науки Российской Федерации – <http://минобрнауки.рф/>

Федеральная служба по надзору в сфере образования и науки – <http://obrnadzor.gov.ru/>

Портал Федеральных государственных образовательных стандартов высшего образования – <http://fgosvo.ru>

Федеральный портал «Российское образование» – <http://www.edu.ru/>

Информационная система «Единое окно доступа к образовательным ресурсам» – <http://window.edu.ru/>

Федеральный центр информационно-образовательных ресурсов – <http://fcior.edu.ru/>

**Электронные библиотечные системы и ресурсы**

Электронно-библиотечная система «Консультант студента» – <http://www.studentlibrary.ru/cgi-bin/mb4x>

Электронно-библиотечная система «StudMed.ru» – <https://www.studmed.ru>

Научная электронная библиотека Elibrary – Режим доступа: URL: <http://elibrary.ru/>

Справочная правовая система «Консультант Плюс» – Режим доступа: URL: <https://www.consultant.ru/sys/>

**Информационный ресурс библиотеки образовательной организации**

Научная библиотека имени А. Н. Коняева – Режим доступа: URL: <http://biblio.dahluniver.ru/>

**7. Материально-техническое и программное обеспечение дисциплины**

Освоение дисциплины «Разработка прикладного программного обеспечения» предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Прочее: рабочее место преподавателя, оснащенное компьютером с доступом в Интернет.

**Программное обеспечение:**

| Функциональное назначение | Бесплатное программное обеспечение | Ссылки                                                                  |
|---------------------------|------------------------------------|-------------------------------------------------------------------------|
| Офисный пакет             | Libre Office 6.3.1                 | <a href="https://www.libreoffice.org/">https://www.libreoffice.org/</a> |

|                      |                                       |                                                                                                                                                                                                                                           |
|----------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      |                                       | <a href="https://ru.wikipedia.org/wiki/LibreOffice">https://ru.wikipedia.org/wiki/LibreOffice</a>                                                                                                                                         |
| Операционная система | UBUNTU 19.04                          | <a href="https://ubuntu.com/">https://ubuntu.com/</a><br><a href="https://ru.wikipedia.org/wiki/Ubuntu">https://ru.wikipedia.org/wiki/Ubuntu</a>                                                                                          |
| Браузер              | Firefox Mozilla                       | <a href="http://www.mozilla.org/ru/firefox/fx">http://www.mozilla.org/ru/firefox/fx</a>                                                                                                                                                   |
| Браузер              | Opera                                 | <a href="http://www.opera.com">http://www.opera.com</a>                                                                                                                                                                                   |
| Почтовый клиент      | Mozilla Thunderbird                   | <a href="http://www.mozilla.org/ru/thunderbird">http://www.mozilla.org/ru/thunderbird</a>                                                                                                                                                 |
| Файл-менеджер        | Far Manager                           | <a href="http://www.farmanager.com/download.php">http://www.farmanager.com/download.php</a>                                                                                                                                               |
| Архиватор            | 7Zip                                  | <a href="http://www.7-zip.org/">http://www.7-zip.org/</a>                                                                                                                                                                                 |
| Графический редактор | GIMP (GNU Image Manipulation Program) | <a href="http://www.gimp.org/">http://www.gimp.org/</a><br><a href="http://gimp.ru/viewpage.php?page_id=8">http://gimp.ru/viewpage.php?page_id=8</a><br><a href="http://ru.wikipedia.org/wiki/GIMP">http://ru.wikipedia.org/wiki/GIMP</a> |
| Редактор PDF         | PDFCreator                            | <a href="http://www.pdfforge.org/pdfcreator">http://www.pdfforge.org/pdfcreator</a>                                                                                                                                                       |
| Аудиоплеер           | VLC                                   | <a href="http://www.videolan.org/vlc/">http://www.videolan.org/vlc/</a>                                                                                                                                                                   |

## 8. Оценочные средства по дисциплине

### Паспорт фонда оценочных средств по учебной дисциплине «Разработка прикладного программного обеспечения»

#### Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины (модуля) или практики

| № п / п | Код контролируемой компетенции | Формулировка контролируемой компетенции                                                           | Индикаторы достижений компетенции (по реализуемой дисциплине)            | Контролируемые темы учебной дисциплины, практики                                                                                                                                                                                                                                                        | Этапы формирования (семестры изучения) |
|---------|--------------------------------|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| 1       | ОПК-14.                        | Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ПК-14.1. Разрабатывает алгоритмы, пригодные для практического применения | Тема 1. Введение в C++. Особенности языка. Среда разработки.<br>Основные особенности<br>Популярные среды разработки (IDE).<br>Тема 2. Базовые конструкции: типы данных, операторы, функции.<br>Основные типы данных. Операторы. Функции.<br>Тема 3. Указатели и ссылки. Динамическая память. Указатели. | 7                                      |

|  |  |  |                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |   |
|--|--|--|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|  |  |  |                                                                                            | Ссылки. Динамическая память. Умные указатели (modern C++).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |   |
|  |  |  | ОПК-14.2.<br>Разрабатывает компьютерные программы, пригодные для практического применения. | <p>Тема 4. Пространства имен. Препроцессор. Пространства имен. Препроцессор.</p> <p>Тема 5. Работа с файлами и потоками ввода-вывода. Работа с файлами. Потоки ввода-вывода. Базовый ввод-вывод.</p> <p>Тема 6. Классы и объекты. Инкапсуляция. Базовый класс. Специальные методы класса.</p> <p>Тема 7. Наследование и полиморфизм. Наследование. Полиморфизм. Абстрактные классы.</p> <p>Тема 8. Шаблоны функций и классов. Шаблоны функций. Шаблоны классов.</p> <p>Тема 9. Стандартная библиотека шаблонов (STL). Контейнеры. Алгоритмы и итераторы.</p> <p>Тема 10 Обработка исключений Базовые исключения. Исключения в конструкторах.</p> <p>Тема 11. Создание графических интерфейсов (Qt Framework). Базовая структура Qt приложения.</p> <p>Тема 12. Работа с базами данных в C++. Использование SQLite.</p> <p>Тема 13. Многопоточное программирование. Потоки и мьютексы. Асинхронное программирование.</p> <p>Тема 14. Оптимизация и профилирование кода.</p> | 7 |

**Показатели и критерии оценивания компетенций,  
описание шкал оценивания**

| № п/п | Код контролируемой компетенции                | Индикаторы достижений компетенции (по реализуемой дисциплине) | Перечень планируемых результатов                           | Контролируемые темы учебной дисциплины | Наименование оценочного средства      |
|-------|-----------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------|----------------------------------------|---------------------------------------|
| 1     | ОПК-14.<br>Способен разрабатывать алгоритмы и | ОПК-14.1.<br>Разрабатывает алгоритмы, пригодные для           | Знать: Принципы объектно-ориентированного программирования | Тема 1<br>Тема 2<br>Тема 3<br>Тема 4   | Вопросы для комбинированного контроля |

|  |                                                                |                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                       |                                                                                                                                                |
|--|----------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
|  | компьютерные программы, пригодные для практического применения | практического применения.                                                               | Уметь: Разрабатывать алгоритмы решения инженерных задач<br>Владеть методами объектно-ориентированного проектирования                                                                                                                                                                                                                                                                                                          |                                                                                                       | усвоения теоретического материала (устно или письменно), задания по практическим занятиям, тест, экзамен                                       |
|  |                                                                | ОПК-14.2. Разрабатывает компьютерные программы, пригодные для практического применения. | Знать: Структуру и возможности стандартной библиотеки шаблонов, методы создания графических интерфейсов, Основы работы с файлами и базами данных<br>Уметь: Создавать классы для описания объектов машиностроения, использовать STL для работы с данными, проводить отладку и тестирование программ<br>Владеть: Навыками программирования на C++, Технологиями создания прикладного ПО, Навыками работы со средами разработки. | Тема 5<br>Тема 6<br>Тема 7<br>Тема 8<br>Тема 9<br>Тема 10<br>Тема 11<br>Тема 12<br>Тема 13<br>Тема 14 | Вопросы для комбинированного контроля усвоения теоретического материала (устно или письменно), задания по практическим занятиям, тест, экзамен |

### **Фонды оценочных средств по дисциплине «Разработка прикладного программного обеспечения»**

#### **Вопросы для комбинированного контроля усвоения теоретического материала (устно или письменно)**

1. Объясните преимущества использования C++ для разработки инженерного ПО в сравнении с другими языками программирования.
2. Какие принципы ООП наиболее важны при моделировании машиностроительных объектов? Приведите примеры из области прототипирования.
3. В чем разница между композицией и наследованием? Какой подход предпочтительнее при моделировании сборочных единиц?
4. Как эффективно управлять памятью при работе с большими CAD-моделями? Объясните на примере.

5. В чем преимущества умных указателей при разработке систем длительного выполнения (например, симуляторов литейных процессов)?
6. Какие алгоритмы STL наиболее полезны для обработки массивов данных измерений в системах контроля качества?
7. Как организовать вычисления с плавающей точкой для обеспечения точности в инженерных расчетах?
8. Опишите оптимальную структуру классов для представления технологического процесса прототипирования.
9. Как эффективно организовать хранение и доступ к справочным данным (материалы, допуски, стандарты)?
10. Разработайте архитектуру системы расчета режимов резания. Какие классы и модули необходимы?
11. Как организовать взаимодействие между модулями в системе управления проектами прототипирования?
12. Какие требования к GUI предъявляются к инженерному ПО? Чем они отличаются от обычных приложений?
13. Как организовать ввод/вывод данных для системы обмена с CAD-системами?
14. Опишите способы интеграции ПО для прототипирования с оборудованием (3D-принтеры, станки ЧПУ).
15. Какие протоколы передачи данных наиболее подходят для систем сбора данных с измерительного оборудования?
16. Какие методы оптимизации кода критически важны для систем инженерного анализа?
17. Как организовать многопоточные вычисления в системе симуляции литейных процессов?
18. Проанализируйте, какие структуры данных оптимальны для хранения информации о дереве сборки машиностроительного изделия.
19. Предложите архитектуру системы визуализации допусков и посадок. Какие библиотеки и подходы следует использовать?
20. Разработайте концепцию системы поддержки принятия решений при выборе технологии прототипирования
21. Какие стратегии тестирования наиболее эффективны для проверки корректности инженерных расчетов?
22. Как организовать обработку исключений в системе управления критически важными производственными процессами?
23. Какие меры обеспечения надежности необходимы в ПО для управления оборудованием прототипирования?
24. Как применение шаблонов проектирования (design patterns) улучшает сопровождаемость инженерного ПО?
25. Какие современные технологии C++ (C++17/20) наиболее перспективны для машиностроительных приложений?

Критерии и шкала оценивания по оценочному средству –

*комбинированный контроль усвоения теоретического материала*

| Шкала оценивания<br>(интервал баллов) | Критерий оценивания                                                                                                                                                                                   |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5                                     | Ответ дан на высоком уровне (студент в полном объеме осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, владеет профильным понятийным (категориальным) аппаратом и т.п.) |
| 4                                     | Ответ дан на среднем уровне (студент в целом осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, допустив некоторые неточности и т.п.)                                    |
| 3                                     | Ответ дан на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)                       |
| 2                                     | Ответ дан на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)                                                                                            |

### **Практические работы**

#### **Практическая работа №1**

Тема: Система расчета параметров режущего инструмента

Цель: Разработать на C# программу с графическим интерфейсом (Windows Forms/WPF) для определения оптимальных геометрических параметров резца в зависимости от обрабатываемого материала и типа обработки.

Используемое ПО: Visual Studio, C#, .NET Framework, Windows Forms/WPF

Краткие сведения из теории: Основные геометрические параметры резца: главный задний угол ( $\alpha$ ), передний угол ( $\gamma$ ), угол в плане ( $\phi$ ). Влияние обрабатываемого материала (сталь, чугун, алюминий) и типа обработки (черновая, чистовая) на выбор углов. Использование справочных таблиц стандартных значений.

Порядок выполнения:

1. Создать проект Windows Forms/WPF.
2. Разработать интерфейс: ComboBox для выбора материала и типа обработки, NumericUpDown для ввода глубины резания, Button для расчета, DataGridView/ListBox для вывода результатов.
3. Реализовать класс 'CuttingToolCalculator' с методами для загрузки данных и расчета углов.
4. Использовать структуру 'Dictionary' или XML-файл для хранения справочных данных о рекомендуемых углах.
5. Реализовать логику расчета, вывод результатов в таблицу и функцию сохранения в текстовый файл.

Контрольные вопросы:

1. Как выбор материала заготовки влияет на значение переднего угла?
2. Объясните, как в вашей программе организована работа со справочными данными.

3. Какие классы и методы .NET Framework вы использовали для создания GUI?

Содержание отчета: Титульный лист, цель, листинг кода, схема классов, скриншоты интерфейса и результатов расчета для разных условий, выводы.

#### Практическая работа №2

Тема: Система проектирования литейной оснастки

Цель: Разработать на C++ с использованием библиотеки Open CASCADE или на C# с использованием API Kompas-3D параметрический скрипт для генерации 3D-модели литниковой системы.

Используемое ПО: Visual Studio, C++ (Open CASCADE) или C# (API Kompas-3D/SolidWorks).

Краткие сведения из теории: Элементы литниковой системы: стояк, шлакоуловитель, питатели. Параметрические зависимости: диаметр стояка рассчитывается от массы отливки и скорости заливки.

Порядок выполнения:

1. Реализовать ввод основных параметров (масса отливки, материал).
2. Написать алгоритм расчета диаметра стояка и других элементов системы по эмпирическим формулам.
3. Используя API выбранной системы, создать 3D-модели элементов оснастки (цилиндр для стояка, усеченный конус для шлакоуловителя).
4. Обеспечить сборку модели литниковой системы в единый узел.

Контрольные вопросы:

1. Какие функции API вы использовали для создания тел вращения?
2. Как реализована параметрическая связь между расчетной частью и геометрией?
3. В чем преимущества использования API для решения таких задач?

Содержание отчета: Титульный лист, цель, алгоритм расчета, листинг кода, скриншоты полученной 3D-модели, выводы.

#### Практическая работа №3

Тема: Программа анализа допусков и посадок.

Цель: Создать на C++ консольное приложение или на C# приложение с GUI для автоматического определения типа посадки, зазоров и натягов по заданным полям допусков.

Используемое ПО: Visual Studio, C++ (STL) или C# (Windows Forms).

Краткие сведения из теории: Понятия системы отверстия и вала. Формулы для расчета предельных отклонений, зазоров и натягов. Квалитеты точности и основные отклонения.

Порядок выполнения:

1. Реализовать структуру или класс для хранения данных о поле допуска (обозначение, верхнее и нижнее отклонения).
2. Загрузить данные из файла (JSON, XML) или ввести вручную для нескольких полей допусков.

3. Реализовать функции для расчета предельных размеров, зазоров ( $S_{\max}$ ,  $S_{\min}$ ) и натягов ( $N_{\max}$ ,  $N_{\min}$ ).

4. Написать логику для определения типа посадки на основе расчетных значений.

5. Вывести результаты в консоль или графический интерфейс в виде таблицы.

Контрольные вопросы:

1. Как программно отличить посадку с натягом от посадки с зазором?

2. Какие структуры данных в C++/C# наиболее удобны для хранения табличных значений допусков?

3. Как организован ввод-вывод данных в вашей программе?

Содержание отчета: Титульный лист, цель, листинг программы, примеры расчетов для разных посадок, скриншоты работы программы.

#### Практическая работа №4

Тема: Система планирования раскроя листового материала

Цель: Реализовать на C++/C# алгоритм размещения прямоугольных деталей на листе для минимизации отходов.

Используемое ПО Visual Studio, C++/C#, библиотека для визуализации (например, SFML для C++ или Windows.Forms + GDI+ для C#).

Краткие сведения из теории: Задача двумерной упаковки. Эвристические алгоритмы (например, "First Fit Decreasing Height"). Понятие коэффициента использования материала (КИМ).

Порядок выполнения:

1. Реализовать классы `Sheet` и `Detail`.

2. Написать алгоритм размещения деталей на листе в соответствии с выбранной эвристикой.

3. Реализовать визуализацию результата раскроя.

4. Рассчитать и вывести значение КИМ.

Контрольные вопросы:

1. Какова вычислительная сложность реализованного алгоритма?

2. Каким образом можно модифицировать алгоритм для учета направления волокон материала?

3. Как в коде реализовано предотвращение пересечения деталей?

Содержание отчета: Титульный лист, цель, описание алгоритма, листинг программы, скриншоты визуализации для разных наборов данных, анализ эффективности.

#### Практическая работа №5

Тема: Модуль анализа технологичности конструкций.

Цель: Разработать на C++/C# модуль для проверки CAD-модели (на основе данных из STL-файла) на соблюдение правил литья.

Используемое ПО: Visual Studio, C++/C#, библиотеки для работы с STL (например, собственный парсер).

Краткие сведения из теории: Основные требования технологичности: равномерность толщин стенок, наличие литейных уклонов, минимальные радиусы закруглений.

Порядок выполнения:

1. Реализовать загрузку и парсинг STL-файла.
2. Реализовать анализ модели (например, проверку толщины стенок через анализ соседних треугольников).
3. Выявлять конструктивные элементы, не соответствующие критериям технологичности.
4. Формировать отчет о найденных ошибках.

Контрольные вопросы:

1. Как в вашем коде представлена модель из STL-файла?
2. Какие критерии вы использовали для определения "резкого перепада толщины"?
3. Какие ограничения имеет анализ на основе STL-файла по сравнению с анализом в CAD-системе?

Содержание отчета: Титульный лист, цель, листинг программы, скриншоты модели и отчета о проверке, выводы.

#### Практическая работа №6

Тема: Система управления проектами прототипирования

Цель: Создать на C# (WPF или ASP.NET Core) приложение для управления задачами, сроками и ресурсами в проекте прототипирования.

Используемое ПО: Visual Studio, C#, WPF/ASP.NET Core, Entity Framework Core, SQL Server LocalDB

Краткие сведения из теории: Основы управления проектами. Жизненный цикл проекта прототипирования. Методы Agile/Waterfall.

Порядок выполнения:

1. Спроектировать базу данных (Проекты, Задачи, Исполнители, Ресурсы).
2. Реализовать модель данных с помощью Entity Framework Code First.
3. Разработать пользовательский интерфейс для CRUD-операций с проектами и задачами.
4. Реализовать функцию построения диаграммы Ганта (упрощенной).

Контрольные вопросы:

1. Как организована связь между сущностями в вашей модели данных?
2. Какие компоненты WPF/ASP.NET Core вы использовали для отображения данных?
3. Как в приложении реализована бизнес-логика (например, проверка на пересечение сроков задач)?

Содержание отчета: Титульный лист, цель, ER-диаграмма базы данных, листинги ключевых классов, скриншоты интерфейса.

#### Практическая работа №7

Тема: Программа расчета режимов 3D-печати.

Цель: Разработать на C++/C# программу для расчета времени печати, расхода материала и оптимальных температурных режимов FDM 3D-принтера.

Используемое ПО: Visual Studio, C++/C#.

Краткие сведения из теории: Основные параметры FDM-печати: скорость печати, температура сопла и стола, высота слоя, заполнение. Зависимость качества и прочности от параметров.

Порядок выполнения:

1. Реализовать класс `PrintProfile` с параметрами печати.
2. Написать парсер G-кода для анализа траекторий и расчета длины экструзии.
3. Реализовать расчет времени печати на основе длин траекторий и скоростей.
4. Реализовать расчет массы материала на основе объема экструзии и плотности пластика.

Контрольные вопросы:

1. Каким образом рассчитывается объем экструдированного материала?
2. Как в программе учитываются разные скорости печати для различных типов перемещений?
3. Какие параметры печати оказывают наибольшее влияние на ее продолжительность?

Содержание отчета: Титульный лист, цель, листинг программы, примеры расчетов для разных моделей и профилей, выводы.

## Практическая работа №8

Тема: Система контроля качества деталей.

Цель: Создать на C++ с использованием библиотек компьютерного зрения (OpenCV) модуль для сравнения эталонного изображения детали с изображением полученного прототипа.

Используемое ПО: Visual Studio, C++, OpenCV

Краткие сведения из теории: Методы обработки изображений: бинаризация, морфологические операции, поиск контуров. Сравнение форм и размеров.

Порядок выполнения:

1. Реализовать загрузку и предобработку изображений (перевод в grayscale, фильтрация).
2. Найти контуры на эталонном и тестовом изображениях.
3. Реализовать сравнение контуров с помощью методов OpenCV (например, `matchShapes`).
4. Визуализировать различия и вывести количественную оценку отклонения.

Контрольные вопросы:

1. Какой метод сравнения контуров вы использовали и в чем его принцип?

2. Какое влияние на результат оказывает этап предобработки изображения?

3. Каковы ограничения данного подхода к контролю качества?

Содержание отчета: Титульный лист, цель, листинг программы, скриншоты этапов обработки изображений, результаты сравнения для тестовых примеров.

#### Практическая работа №9

Тема: Модуль симуляции литейных процессов.

Цель: Реализовать на C++ упрощенную 2D-модель заливки расплава в форму для анализа заполняемости.

Используемое ПО: Visual Studio, C++, библиотеки для визуализации (например, SFML).

Краткие сведения из теории: Уравнение неразрывности. Упрощенные модели течения вязкой жидкости. Понятие о методе конечных разностей.

Порядок выполнения:

1. Создать сеточную модель полости формы.
2. Реализовать алгоритм, моделирующий распространение жидкости от точки заливки.
3. Учесть основные факторы (вязкость, гравитация).
4. Визуализировать процесс заполнения и определить возможные места недоливов.

Контрольные вопросы:

1. Какие упрощения физики процесса были сделаны в вашей модели?
2. Как в алгоритме учитывается геометрия литниковой системы?
3. Как можно повысить точность симуляции?

Содержание отчета: Титульный лист, цель, математическая модель, листинг программы, скриншоты ключевых этапов симуляции, анализ результатов.

#### Практическая работа №10

Тема: Интегрированная система быстрого прототипирования.

Цель: Разработать на C# (WPF) многооконное приложение, объединяющее функционал нескольких модулей (например, анализ технологичности, расчет стоимости, управление проектом) через единый интерфейс.

Используемое ПО: Visual Studio, C#, WPF, паттерн MVVM.

Краткие сведения из теории: Принципы модульности и слабого зацепления. Паттерн проектирования Model-View-ViewModel (MVVM). Межмодульное взаимодействие.

Порядок выполнения:

1. Спроектировать архитектуру приложения по паттерну MVVM.
2. Выделить общие сервисы (например, для работы с данными проекта).

3. Интегрировать 2-3 модуля из предыдущих работ в виде отдельных компонентов (UserControl или отдельных View).

4. Реализовать механизм обмена данными между модулями (через общую Model или сервис сообщений).

Контрольные вопросы:

1. Как в архитектуре MVVM реализовано разделение ответственности?
2. Каким способом организовано взаимодействие между вашими модулями?
3. Каковы преимущества и недостатки созданной интеграции?

Содержание отчета: Титульный лист, цель, схема архитектуры приложения, листинги ключевых компонентов, скриншоты интерфейса, описание workflow.

Критерии и шкала оценивания по оценочному средству –  
*практические работы*

| Шкала оценивания<br>(интервал баллов) | Критерий оценивания                                                                                                                                                                                           |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5                                     | Задание выполнено на высоком уровне (студент в полном объеме осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, владеет профильным понятийным (категориальным) аппаратом и т.п.) |
| 4                                     | Задание выполнено на среднем уровне (студент в целом осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, допустив некоторые неточности и т.п.)                                    |
| 3                                     | Задание выполнено на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)                       |
| 2                                     | Задание выполнено на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)                                                                                            |

**Тест**

1. Какая пара принципов ООП наиболее важна для создания модуля анализа технологичности, где класс 'Деталь' наследует от базового класса 'Геометрическое Тело' и использует приватные методы для внутренних расчетов?

- а) Полиморфизм и инкапсуляция
- б) Наследование и инкапсуляция
- в) Абстракция и полиморфизм
- г) Инкапсуляция и композиция

2. При разработке системы расчета параметров режущего инструмента необходимо хранить и быстро выбирать значения углов резца для различных материалов. Какой контейнер STL будет наиболее эффективен?

- а) `std::vector`
- б) `std::list`

- в) ``std::map`` (или ``std::unordered_map``)
- г) ``std::array``

3. Для системы планирования раскроя листового материала требуется динамически управлять памятью под массив координат деталей. Какая операция в C++ является корректной и безопасной для этого?

- а) Использование ``malloc`` и ``free``
- б) Использование операторов ``new`` и ``delete``
- в) Использование умного указателя ``std::unique_ptr`` с массивом (``std::unique_ptr<int[]>``)
- г) Создание массива на стеке с помощью ``int arr[1000]``

4. При проектировании системы управления проектами прототипирования создан класс ``Project`` (Проект). Как правильно запретить создание копий объекта этого класса?

- а) Объявить конструктор копирования и оператор присваивания и сделать их приватными
- б) Использовать ключевое слово ``const``
- в) Объявить деструктор приватным
- г) Использовать ключевое слово ``mutable``

5. В модуле симуляции литейных процессов используется функция, которая должна модифицировать переданный в нее объект класса ``Form`` (Форма). Как правильно передать этот объект в функцию?

- а) По значению: ``void simulate(Form f)``
- б) По константной ссылке: ``void simulate(const Form& f)``
- в) По ссылке: ``void simulate(Form& f)``
- г) Через указатель: ``void simulate(Form *f)``

6. При обработке данных от 3D-сканера в системе контроля качества может произойти ошибка чтения файла. Какой механизм C++ является наиболее предпочтительным для обработки таких ошибок?

- а) Возврат кода ошибки из функции
- б) Использование глобальной переменной для хранения ошибки
- в) Использование механизма исключений (``try`/`catch``)
- г) Вывод сообщения в стандартный поток ошибок ``std::cerr``

7. Для программы расчета режимов 3D-печати необходимо обеспечить работу с типами данных для чисел с плавающей точкой двойной точности и гарантировать переносимость. Какой тип следует использовать?

- а) ``float``
- б) ``double``
- в) ``long double``
- г) ``auto``

8. В интегрированной системе прототипирования несколько модулей должны использовать единый объект для ведения лога событий. Какой шаблон проектирования лучше всего подходит для реализации этого объекта?

- а) Фабричный метод (Factory Method)
- б) Одиночка (Singleton)
- в) Наблюдатель (Observer)
- г) Стратегия (Strategy)

9. При разработке программы анализа допусков и посадок создан шаблонный класс `Tolerance` (Допуск). Что означает следующая строка кода: `Tolerance<double> tol;`?

- а) Создание специализации класса `Tolerance` для типа `double`
- б) Объявление функции `tol` с возвращаемым типом `Tolerance<double>`
- в) Создание объекта `tol` класса `Tolerance`, параметризованного типом `double`
- г) Явное инстанцирование шаблона

10. Какая функция является точкой входа в консольное приложение на C++?

- а) `main()`
- б) `start()`
- в) `init()`
- г) `run()`

Ключ к тесту:

1. б) Наследование и инкапсуляция
2. в) `std::map` (или `std::unordered_map`)
3. в) Использование умного указателя `std::unique_ptr` с массивом
4. а) Объявить конструктор копирования и оператор присваивания и сделать их приватными (или использовать `= delete`)
5. в) По ссылке: `void simulate(Form& f)`
6. в) Использование механизма исключений (`try/catch`)
7. б) `double`
8. б) Одиночка (Singleton)
9. в) Создание объекта `tol` класса `Tolerance`, параметризованного типом `double`
10. а) `main()`

Критерии и шкала оценивания по оценочному средству – *тест*

| Шкала оценивания | Критерий оценивания                                                          |
|------------------|------------------------------------------------------------------------------|
| 5                | Тесты выполнены на высоком уровне (правильные ответы даны на 90-100% тестов) |
| 4                | Тесты выполнены на среднем уровне (правильные ответы даны на 75-89% тестов)  |
| 3                | Тесты выполнены на низком уровне (правильные ответы даны на 50-74% тестов)   |

|   |                                                                                                 |
|---|-------------------------------------------------------------------------------------------------|
| 2 | Тесты выполнены на неудовлетворительном уровне (правильные ответы даны менее чем на 50% тестов) |
|---|-------------------------------------------------------------------------------------------------|

### **Критерии и шкала оценивания по оценочному средству итоговый контроль (экзамен)**

1. Отличия языков C и C++. Основные парадигмы (процедурное vs. объектно-ориентированное программирование). Области применения каждого языка в разработке ПО для машиностроения (низкоуровневые драйверы vs. высокоуровневые CAD-системы).

2. Управление памятью в C и C++. Понятие стека и кучи. Работа с операторами `'new'/'delete'` в C++ и функциями `'malloc()'/'free()'` в C. Проблемы утечек памяти и их влияние на надежность инженерных приложений.

3. Объектно-ориентированное программирование в C++. Инкапсуляция, наследование, полиморфизм. Приведите пример иерархии классов для описания геометрических примитивов (например, `'Фигура'` -> `'Вектор3D'` -> `'Многогранник'`).

4. Шаблоны (templates) в C++. Их назначение и преимущества. Как шаблоны могут быть использованы для создания универсальных библиотек математических вычислений (например, шаблонный класс `'Matrix<T>'`)?

5. Структуры данных в C/C++. Структуры (`'struct'`) и классы (`'class'`). Разработайте структуру для хранения данных о вершине 3D-модели (координаты x, y, z, нормаль).

6. Работа с файлами в C и C++. Стандартные библиотеки `'stdio.h'` (C) и `'fstream'` (C++). Особенности чтения и записи бинарных данных. Пример чтения заголовка STL-файла.

7. Контейнеры стандартной библиотеки шаблонов (STL). Преимущества использования `'std::vector'`, `'std::list'`, `'std::map'` по сравнению с массивами C. Пример использования `'std::vector'` для хранения массива вершин треугольной сетки.

8. Указатели и ссылки в C/C++. Их сходства и различия. Использование указателей для работы с динамическими массивами данных (например, массив точек облака точек, полученного с 3D-сканера).

9. Обработка исключений в C++. Механизм `'try'/'catch'/'throw'`. Зачем он нужен в критическом ПО, например, при расчете прочностных характеристик?

10. Оптимизация кода на C/C++. Роль `'const'`, передача параметров по ссылке, инлайнинг функций. Почему это критически важно в циклах обработки миллионов полигонов?

11. Математические библиотеки для научных вычислений. Использование стандартной библиотеки `'<cmath>'`. Обзор специализированных библиотек (например, Eigen для линейной алгебры). Пример решения системы линейных уравнений методом Гаусса.

12. Создание библиотек (DLL / .so) на C/C++. Цели и процесс создания. Какую роль играют библиотеки в модульной архитектуре CAD/CAE систем?

13. Разработка алгоритмов для работы с геометрией. Реализация на C++ базовых операций вычислительной геометрии: расчет расстояния между точкой и плоскостью, проверка принадлежности точки объему.

14. Интеграция кода на C/C++ с другими языками и средами. Примеры вызова функций C/C++ из Python для высокопроизводительных расчетов в скриптах для САЕ-систем.

15. Многопоточное программирование в C++. Использование библиотеки `<thread>`. Как распараллелить расчет напряжений в различных элементах конечно-элементной сетки?

16. Разработка простого графического интерфейса пользователя (GUI). Обзор библиотек (Qt, wxWidgets). Спроектируйте интерфейс для приложения, управляющего параметрами 3D-печати (поле ввода скорости, температуры, кнопка "Старт").

17. Работа с аппаратурой. Использование C/C++ для написания низкоуровневых драйверов и систем управления оборудованием для прототипирования (например, управление шаговыми двигателями 3D-принтера через COM-порт).

18. Численные методы в C++. Реализация методов интерполяции (например, сплайны) или аппроксимации для обработки данных измерений с датчиков.

19. Понятие "Zero Overhead Principle" в C++. Как этот принцип соотносится с требованиями к производительности в реальном времени для систем управления технологическим оборудованием?

20. Инструментарий разработчика. Назначение и базовое использование компилятора (gcc/clang), отладчика (gdb), системы сборки (Makefile, CMake). Почему понимание этого инструментария необходимо для создания сложных прикладных программ?

#### Критерии и шкала оценивания по оценочному средству итоговый контроль (экзамен)

| Шкала оценивания      | Характеристика знания предмета и ответов                                                                                                                                                                                                                                                                                                                    |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| отлично (5)           | Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач. |
| хорошо (4)            | Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.                                      |
| удовлетворительно (3) | Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или                                                                                                                                                                                         |

|                         |                                                                                                                                                                                                                                                                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.                                                                                                                                                                              |
| неудовлетворительно (2) | Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы. |

### Лист изменений и дополнений

| №<br>п/п | Виды дополнений и<br>изменений | Дата и номер протокола<br>заседания кафедры<br>(кафедр), на котором были<br>рассмотрены и одобрены<br>изменения и дополнения | Подпись (с<br>расшифровкой)<br>заведующего кафедрой<br>(заведующих кафедрами) |
|----------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
|          |                                |                                                                                                                              |                                                                               |
|          |                                |                                                                                                                              |                                                                               |
|          |                                |                                                                                                                              |                                                                               |
|          |                                |                                                                                                                              |                                                                               |