

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра теории упругости и вычислительной математики
имени академика А.С. Космодамианского

УТВЕРЖДАЮ:

проректор по научно-методической
и учебной работе

Е.И. Скафа

«21» апреля 2021 г.

МП



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И
СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ
ИССЛЕДОВАНИЯ МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ»
практико-ориентированная дисциплина

Направление подготовки:	01.03.02 Прикладная математика и информатика
Профиль подготовки:	Статистика
Образовательная программа:	Бакалавриат
Квалификация:	Академический бакалавр
Форма обучения:	очная

Донецк 2021

УТВЕРЖДАЮ:

Декан факультета математики
и информационных технологий

И. А. Моисеенко



«20» апреля 2021 г.

Рабочая программа учебной дисциплины «Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования» составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 01.03.02 Прикладная математика и информатика, утверждённого приказом Министерства образования и науки Российской Федерации от «10» января 2018 г. № 9; Государственного образовательного стандарта высшего образования (ГОС ВО) Донецкой Народной Республики (ДНР) (проекта) по направлению подготовки 01.03.02 Прикладная математика и информатика; Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки Донецкой Народной Республики от 10.11.2017 г. № 1171 (с изменениями и дополнениями); учебного плана и основной профессиональной образовательной программы высшего образования направления подготовки 01.03.02 Прикладная математика и информатика профиль Прикладная математика и информатика, разработанных в Государственном образовательном учреждении высшего профессионального образования «Донецкий национальный университет».

Разработчик:

проф. кафедры теории упругости и вычислительной математики
им. академика А.С. Космодамианского,
д-р физ.-мат. наук, доцент

И.А. Моисеенко

Рабочая программа учебной дисциплины утверждена на заседании кафедры теории упругости и вычислительной математики им. академика А.С. Космодамианского

Протокол № 15 от «12» апреля 2021 г.

Зав. кафедрой теории упругости и вычислительной математики
им. академика А.С. Космодамианского

В.И. Сторожев

Рабочая программа учебной дисциплины одобрена учебно-методической комиссией факультета математики и информационных технологий

Протокол № 4 от «14» апреля 2021 г.

Председатель учебно-методической комиссии
факультета математики и информационных технологий

Л.И. Селякова

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования» относится к базовой части образовательной программы. Для изучения данной учебной дисциплины необходимы знания и умения, формируемые *предшествующими дисциплинами* – "Языки и методы программирования", "Операционные системы", "Архитектура компьютеров", "Дискретная математика". Знания и умения, полученные в ходе изучения дисциплины «Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования» являются основой для изучения *последующих* дисциплин: «Языки разработки специализированных программных приложений», «Теория автоматов и формальных языков». Учебная дисциплина «Объектно-ориентированное программирование и стандартная библиотека C++» может рассматриваться как естественное продолжение фундаментальных курсов «Основы информатики» и «Языки и методы программирования». На основе базовых курсов, в которых изучается язык программирования C++ прежде всего с точки зрения методов процедурного и структурного программирования, в данном курсе на основе C++ изучается объектно-ориентированная технология (парадигма) программирования – наиболее распространенная и востребованная в настоящее время, концентрирующаяся больше на связях между объектами, функциональные возможности и структуру которых задают классы (типы данных, определенные пользователем), чем на деталях реализации. Изучение ООП на примере именно объектно-ориентированного языка программирования C++ ([9], [14]) целесообразно, в первую очередь, по причине его наиболее теоретически выдержанности в этой части. Другие языки, поддерживающие идеи ООП, такие, как Object Pascal [4], Java [7], разрабатывались в первую очередь с учетом удобства программирования задач в соответствующих предметных областях. В данной дисциплине изучается также стандартная библиотека языка C++, включающая стандартизированную надстройку STL, предоставляющую высокоуровневые структуры: строки, вектора, деки, списки, множества, мультимножества, отображения, мультиотображения, которые могут обрабатываться универсальными алгоритмами. STL – уникальная библиотека, за счет вложенного высшего искусства C++ программистов, принципиально изменившая как внешнюю сторону так и внутреннее содержание процесса разработки приложений на C++. Использование контейнеров позволяет значительно повысить надежность программ, их переносимость и универсальность, а также уменьшить сроки их разработки. Основным достоинством библиотеки STL C++ является то, что она действительно стандартная, т.е. входит в стандарт и обязательна для реализации любым компилятором, удовлетворяющем стандарту C++. Эту библиотеку поддерживает и легендарный UNIX-компилятор GCC, и Visual Studio C++.NET, и C++ Builder, и Dev-C++, в общем, любой "живой" компилятор. Поэтому вполне закономерным можно считать утверждение, что, не зная STL нельзя считать себя профессиональным программистом C++.

Предварительные требования к студентам.

1. Знание одного из классических процедурно-ориентированных языков, предпочтительно языка C.
2. Знания в области алгоритмической декомпозиции, основных структур данных и технологий работы с ним.
3. Знание основ теории множеств.
4. Освоение курсов «Основы информатики», «Языки и методы программирования».

2. СТРУКТУРА ДИСЦИПЛИНЫ

Характеристика учебной дисциплины	Форма обучения
-----------------------------------	----------------

	Очная
Направление подготовки	01.03.02 Прикладная математика и информатика
Профиль	Статистика
Образовательная программа	Бакалавриат
Квалификация	Академический бакалавр
Количество содержательных модулей и тем	3 (15)
Дисциплина базовой / вариативной части образовательной программы	Базовой части
Формы контроля	1 модульный контроль, экзамен в 3-м семестре
Год подготовки	2
Семестр	3
Количество зачетных единиц	5
Количество часов всего	180
в т.ч.:	
- лекционных	54
- практических или семинарских	
- лабораторных	36
- самостоятельной работы	90
в т.ч. индивидуальное задание	–
Недельное количество часов	5
в т. ч.: - аудиторных	5
- самостоятельной работы студента	5

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цели изучения дисциплины «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ»:

Цели учебной дисциплины.

- Изучение основ классической теории объектно-ориентированного программирования, в том числе:
 - пути эволюции технологий программирования от алгоритмического к объектно-ориентированному;
 - основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Наследование, Полиморфизм);
 - понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG.
- Формирование понимания идеологии и ключевых аспектов объектно-ориентированного программирования на языке C++, достаточного для практического использования в процессе дальнейшего обучения и в профессиональной сфере, получение навыков разработки программ в среде Microsoft Visual Studio.
- Изучение средств объектно-ориентированного и обобщенного программирования языка C++.
- Формирование понимания идеологии организации, структуры и методов использования стандартной библиотеки языка C++ в объеме, достаточном для практического применения в процессе дальнейшего обучения и в профессиональной сфере.

Задачи учебной дисциплины.

- Познакомить с теоретическими основами и принципами объектно-ориентированного программирования.

2. Изучить особенности реализации объектно-ориентированного подхода в языке программирования C++.
3. Сформировать практические умения и навыки использования объектно-ориентированного подхода к программированию.
4. Научить студентов разрабатывать в соответствии с парадигмой объектно-ориентированного программирования компьютерные модели реальных и концептуальных систем, соответствующих направлению подготовки 01.03.02 «Прикладная математика и информатика».
5. Познакомить с основными структурными элементами стандартной библиотеки языка C++ включая STL:
 - стандартными контейнерными классами, реализующими наиболее распространенные структуры для хранения данных - векторами, деками, списками, множествами, мультимножествами, отображениями, мультиотображениями;
 - стандартными универсальными алгоритмами, использующими эти контейнеры;
 - итераторами, предназначенными для организации унифицированного доступа к элементам этих контейнеров.
6. Сформировать практические умения и навыки использования STL при разработке приложений на C++ для моделей реальных и концептуальных систем, соответствующих направлению подготовки 01.03.02 «Прикладная математика и информатика».

Требования к результатам освоения дисциплины. Процесс изучения дисциплины «Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования» направлен на формирование элементов следующих **компетенций** в соответствии с ФГОС ВО РФ, ГОС ВО ДНР (проект) по направлению подготовки 01.03.02 Прикладная математика и информатика и основной профессиональной образовательной программы высшего образования направления подготовки 01.03.02 Прикладная математика и информатика, профиля: «Статистика»:

Общепрофессиональные компетенции (ОПК):	
ОПК-1	Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности
Профессиональные компетенции (ПК):¹	
ПК-2	Способен проводить обработку и анализ научной информации и результатов исследований

Индикаторы достижения компетенций и результаты обучения². Достижение компетенций оценивается на основе таких индикаторов и соответствующих им результатов обучения:

Общепрофессиональные компетенции	Индикаторы	Результаты обучения
ОПК-1. Способен применять фундаментальные знания,	ОПК-1. И-1. Применяет основные	Знает основные теоретические понятия объектно-ориентированного подхода к программированию

¹ Если ПК взята из профессионального стандарта – можно указать название профстандарта, кем и когда утвержден, регистрационный номер профстандарта

² Количество индикаторов по каждой компетенции может варьироваться (от одного и более).

полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности	положения и концепции в области математических наук и основную терминологию при решении задач профессиональной деятельности	Знает синтаксические и семантические аспекты реализации объектно-ориентированного подхода в языке программирования C++
		Умеет анализировать предметную область решаемых задач с целью использования объектно-ориентированного подхода для их реализации, разрабатывать объектную модель программы
		Умеет разрабатывать алгоритмы применительно к методу объектно-ориентированного программирования
	ОПК-1. И-2. Применяет основную терминологию математических наук и математические объекты при решении задач профессиональной деятельности	Знает основные этапы проектирования объектно-ориентированных приложений
		Знает стандартные контейнерные классы, реализующие наиболее распространенные структуры для хранения данных – вектора, деки, списки, множества, мультимножества, отображения, мультиотображения
		Умеет на основе объектно-ориентированного подхода решать практические задачи программирования на языке C++ с использованием: инкапсуляции различных типов данных и методов в классы; простого и множественного дерева наследования; полиморфизма; обработки исключений; шаблонов классов;
		Умеет анализировать предметную область решаемых задач с целью использования доступных структурных элементов STL в разрабатываемой объектной модели программы
		Умеет выбирать методы и средства для реализации программных проектов с использованием структурных элементов STL
		Владеет навыками использования STL при разработке приложений на C++ для моделей реальных и концептуальных систем, соответствующих направлению подготовки «Прикладная математика и информатика»

Профессиональные	Индикаторы	Результаты обучения
------------------	------------	---------------------

компетенции		
ПК-2. Способен проводить обработку и анализ научной информации и результатов исследований	ПК-2. И-1. Осуществляет сбор, обработку и обобщение результатов научных исследований в области компьютерно-математического моделирования	Знает роль и место «Объектно-ориентированного программирования и стандартной библиотеки C++ в численных методах исследования моделей деформирования» в задачах из сферы профессиональной деятельности и в общей естественнонаучной картине мира
		Знает методы и приемы решения профессионально-ориентированных задач
		Умеет реализовать на практике методы и приемы решения задач «Объектно-ориентированного программирования и стандартной библиотеки C++»
		Умеет систематизировать и обобщать инструментальные средства «Объектно-ориентированного программирования и стандартной библиотеки C++» для исследования объектов профессиональной деятельности
		Владеет навыками поиска и выбора инструментальных средств «Объектно-ориентированного программирования и стандартной библиотеки C++» при исследовании объектов профессиональной деятельности
	ПК-2. И-2. . Оформляет результаты научно-исследовательских работ и вычислительных экспериментов в соответствии с актуальными стандартами	Знает тенденции и перспективы развития объектно-ориентированного подхода в программировании
		Знает методы описания, конструктивного анализа, формализованного описания проблемных ситуаций
		Умеет критически переосмысливать накопленные знания в соответствии с характером своей профессиональной деятельности
		Умеет на основе объектно-ориентированного подхода решать практические задачи программирования на языке C++ с использованием: стандартных контейнеров STL; стандартных универсальных алгоритмов, обрабатывающих эти контейнеры
		Владеет одним из современных языков программирования применительно к технологии объектно-ориентированного программирования

4. ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Дисциплина «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ» предусматривает следующие формы организации учебного процесса: лекции, лабораторные занятия, самостоятельную работу студентов.

Материал излагается с использованием объяснительно-иллюстративных и эвристических методов преподавания. При проведении лекций и лабораторных занятий используются мультимедийные презентации, раздаточные материалы, специальное оборудование.

В учебном процессе широко применяются активные и интерактивные формы проведения занятий (разбор конкретных ситуаций, дискуссия, полемика), внеаудиторная самостоятельная работа, балльно-рейтинговая система оценки успеваемости, личностно-ориентированное обучение, проблемное обучение. В учебном процессе используются интернет-ресурсы по данному курсу; рассматриваются задачи, максимально приближенные к конкретным практическим ситуациям, тесты, самостоятельная работа; контрольные работы.

Самостоятельная работа студентов предусматривает подготовку к лабораторным занятиям, подготовку конспектов по отдельным вопросам изучаемых тем, изучение учебной и методической литературы, выполнение индивидуальных заданий.

Тематический план «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ»

Темы	Вопросы темы
Модуль 1. Объектно-ориентированная парадигма и базовые классы.	
Содержательный модуль 1.1. Инкапсуляция.	
Тема 1. Объектно-ориентированная парадигма	Краткий обзор основных парадигм программирования. Процессно-ориентированная парадигма и язык С. Краткий обзор ключевых конструкций С. Аппликативная (функциональная) парадигма. Парадигма логического программирования. Объектно-ориентированная парадигма. Методология разработки объектно-ориентированного программного обеспечения. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.
Тема 2. Базовые классы	Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации. Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции. Пространства имен. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор. Функциональное замыкание как механизм работы специальных методов

	класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов.
Содержательный модуль 1.2. Статический полиморфизм.	
Тема 3. Перегрузка операторов	Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции. Перегрузка операторов преобразования типа. Спецификатор <i>explicit</i> для конструкторов. Перегрузка операторов <i>new</i> и <i>delete</i> .
Тема 4. Потоки	Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода. Библиотека потока C++. Общие функции ввода-вывода в поток.
Модуль 2. Динамический и параметрический полиморфизм.	
Содержательный модуль 2.1. Наследование и динамический полиморфизм.	
Тема 5. Наследование	Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрывание имен и функциональное замыкание в случае простого наследования.
Тема 6. Динамический полиморфизм	Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы. Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования. Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании. Объектно-ориентированная концепция обработки исключений. Блок <i>try</i> . Оператор <i>catch</i> . Классы исключений. Стандартные исключения. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений).
Содержательный модуль 2.2. Шаблоны и параметрический полиморфизм.	
Тема 7. Параметрический полиморфизм для функций	Параметрический полиморфизм для функций. Синтаксис определения шаблона функции. Механизмы генерирования и идентификации при функциональных обращениях.
Тема 8. Параметрический полиморфизм для классов	Синтаксис определения шаблона класса. Параметры шаблона. Специализация шаблонов классов. Реализация шаблона класса.
Модуль 3. Стандартная библиотека C++	
Содержательный модуль 3.1. Архитектура стандартной библиотеки C++.	
Тема 9. Архитектура библиотеки	Составные части библиотеки. Организация стандартной библиотеки шаблонов. Стандартные контейнеры.
Содержательный модуль 3.2. Итераторы и функциональные объекты.	
Тема 10.	Итераторы. Обратные итераторы. Итераторы вставки. Потоквые

Итераторы	итераторы.
Тема 11. Функциональные объекты	Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели. Связыватели. Адаптеры указателей на функции. Адаптеры методов.
Содержательный модуль 3.3. Контейнеры.	
Тема 12. Последовательные контейнеры	Последовательные контейнеры. Векторы (vector). Списки (list). Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority_queue).
Тема 13. Ассоциативные контейнеры	Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset). Словари (map). Словари с дубликатами (multimap).
Содержательный модуль 3.4. Алгоритмы.	
Тема 14. Операции с последовательностями	Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями.
Тема 15. Специальные алгоритмы	Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами. Другие средства стандартной библиотеки.

**Структура дисциплины «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА C++ В
ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ МОДЕЛЕЙ
ДЕФОРМИРОВАНИЯ» по видам учебной деятельности**

Названия содержательных модулей и тем	Количество часов									
	Очная форма обучения					Заочная форма обучения				
	Всего	В Т.Ч.				Всего	В Т.Ч.			
		Лекции	Практические	Лабораторные	Самостоятельная работа		Лекции	Практические	Лабораторные	Самостоятельная работа
Содержательный модуль 1. <i>Объектно-ориентированная парадигма и базовые классы</i>										
Тема 1. Объектно-ориентированная парадигма	12	3		3	6					
Тема 2. Базовые классы	12	3		3	6					
Тема 3. Перегрузка операторов	11	3		2	6					
Тема 4. Потоки	11	3		2	6					
Итого по содержательному модулю 1	46	12		10	24					
Содержательный модуль 2. <i>Динамический и параметрический полиморфизм</i>										
Тема 5. Наследование	12	3		3	6					
Тема 6. Динамический полиморфизм	12	3		3	6					
Тема 7. Параметрический полиморфизм для функций	11	3		2	6					
Тема 8. Параметрический полиморфизм для классов	11	3		2	6					
Итого по содержательному модулю 2	46	12		10	24					
Содержательный модуль 3. <i>Стандартная библиотека C++</i>										
Тема 9. Архитектура библиотеки	12	4		2	6					
Тема 10. Итераторы	12	4		2	6					
Тема 11. Функциональные объекты	12	4		2	6					
Тема 12. Последовательные контейнеры	12	4		2	6					
Тема 13. Ассоциативные контейнеры	12	4		2	6					
Тема 14. Операции с последовательностями	12	4		2	6					

Тема 15.										
Специальные алгоритмы	16	6		4	6					
Итого по содержательному модулю 3	88	30		16	42					
Всего часов	180	54		36	90					

5. ТЕМАТИКА ЛЕКЦИОННЫХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Темы лекционных занятий

№ п/п	Название темы	Количество часов
1	Краткий обзор основных парадигм программирования. Процессно-ориентированная парадигма и язык С. Краткий обзор ключевых конструкций С. Аппликативная (функциональная) парадигма. Парадигма логического программирования.	2
2	Объектно-ориентированная парадигма. Методология разработки объектно ориентированного программного обеспечения. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.	2
3	Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации	2
4	Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции. Пространства имен. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор	2
5	Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов	2
6	Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции	2
7	Перегрузка операторов преобразования типа. Спецификатор <i>explicit</i> для конструкторов. Перегрузка операторов <i>new</i> и <i>delete</i>	2
8	Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода	2
9	Перегрузка операторов потокового ввода/вывода. Библиотека потока C++. Общие функции ввода-вывода в поток	2
10	Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрывание имен и функциональное замыкание в случае простого наследования	2
11	Преобразование типов (ссылок и указателей). Правила видимости	2

	при простом наследовании. Перекрытие имен и функциональное замыкание в случае простого наследования	
12	Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы	2
13	Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования	2
14	Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании	2
15	Объектно-ориентированная концепция обработки исключений. Блок <i>try</i> . Оператор <i>catch</i> . Классы исключений. Стандартные исключения. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений)	2
16	Параметрический полиморфизм для функций. Синтаксис определения шаблона функции	2
17	Механизмы генерирования и идентификации при функциональных обращениях	2
18	Синтаксис определения шаблона класса. Параметры шаблона. Специализация шаблонов классов. Реализация шаблона класса	2
19	Итераторы. Обратные итераторы. Итераторы вставки. Поточные итераторы	2
20	Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели	2
21	Связыватели. Адаптеры указателей на функции. Адаптеры методов	2
22	Последовательные контейнеры. Векторы (vector). Списки (list).	2
23	Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority_queue)	2
24	Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset)	2
25	Словари (map). Словари с дубликатами (multimap)	2
26	Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями	2
27	Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами. Другие средства стандартной библиотеки	2
Всего		54

Тексты лекций приведены в: электронном УМКД на кафедре ТУиВМ и в дистанционном курсе ООПИСБ C++ ПМИИ на платформе Moodle ДонНУ.

Темы лабораторных занятий

№ п/п	Название темы	Количество часов
----------	---------------	---------------------

1	Процессно-ориентированная парадигма и язык С. Краткий обзор ключевых конструкций С. Аппликативная (функциональная) парадигма. Парадигма логического программирования.	2
2	Методология разработки объектно ориентированного программного обеспечения. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.	2
3	Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации	2
4	Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции. Пространства имен. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор	2
5	Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов	2
6	Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции	2
7	Перегрузка операторов преобразования типа. Спецификатор <i>explicit</i> для конструкторов. Перегрузка операторов <i>new</i> и <i>delete</i>	2
8	Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода	2
9	Перегрузка операторов потокового ввода/вывода. Библиотека потока C++. Общие функции ввода-вывода в поток	2
10	Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования	2
11	Итераторы. Обратные итераторы. Итераторы вставки. Потоковые итераторы. Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели	2
12	Связыватели. Адаптеры указателей на функции. Адаптеры методов	2
13	Последовательные контейнеры. Векторы (vector). Списки (list).	2
14	Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority_queue)	2
15	Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset)	2
16	Словари (map). Словари с дубликатами (multimap)	2
17	Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями	2
18	Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами. Другие средства стандартной библиотеки	2

Всего	36
--------------	-----------

Материалы для проведения лабораторных работ и методические рекомендации к выполнению лабораторных работ приведены в: электронном УМКД на кафедре ТУиВМ и в дистанционном курсе ООПИСБ С++ ПМИИ на платформе Moodle ДонНУ.

6. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

№ п/п	Название темы	Количество часов
1	роцессно-ориентированная парадигма и язык С. Краткий обзор ключевых конструкций С. Аппликативная (функциональная) парадигма	3
2	Методология разработки объектно ориентированного программного обеспечения. Принципы объектно-ориентированного подхода	3
3	Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации	3
4	Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции. Пространства имен. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор	3
5	Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов	3
6	Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции	3
7	Перегрузка операторов преобразования типа. Спецификатор <i>explicit</i> для конструкторов. Перегрузка операторов <i>new</i> и <i>delete</i>	3
8	Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода	3
9	Перегрузка операторов потокового ввода/вывода. Библиотека потока С++. Общие функции ввода-вывода в поток	3
10	Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрытие имен и функциональное замыкание в случае простого наследования	3
11	Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрытие имен и функциональное замыкание в случае простого наследования	3

12	Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы	3
13	Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования	3
14	Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании	3
15	Объектно-ориентированная концепция обработки исключений. Блок <i>try</i> . Оператор <i>catch</i> . Классы исключений. Стандартные исключения. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений)	3
16	Параметрический полиморфизм для функций. Синтаксис определения шаблона функции	3
17	Механизмы генерирования и идентификации при функциональных обращениях	3
18	Синтаксис определения шаблона класса. Параметры шаблона. Специализация шаблонов классов. Реализация шаблона класса	3
19	Итераторы. Обратные итераторы. Итераторы вставки. Поточные итераторы	3
20	Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели	3
21	Связыватели. Адаптеры указателей на функции. Адаптеры методов	3
22	Последовательные контейнеры. Векторы (vector). Списки (list).	3
23	Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority_queue)	3
24	Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset)	3
25	Словари (map).	3
26	Словари с дубликатами (multimap)	3
27	Немодифицирующие операции с последовательностями	3
28	Модифицирующие операции с последовательностями	3
29	Алгоритмы работы с множествами и пирамидами	3
30	Другие средства стандартной библиотеки	3
Всего		90

Содержание самостоятельной (в т.ч. индивидуальной) работы по темам и методические рекомендации по ее выполнению приведены в: электронном УМКД на кафедре ТУиВМ и в дистанционном курсе ООПИСБ C++ ПМИИ на платформе Moodle ДонНУ.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Контрольные вопросы:

1. Процессно-ориентированная парадигма и язык С. Краткий обзор ключевых

конструкций С. Аппликативная (функциональная) парадигма. Парадигма логического программирования.

2. Объектно-ориентированная парадигма. Методология разработки объектно-ориентированного программного обеспечения.

3. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.

4. Понятие класса, объекта, поля, метода. Определение базового класса. Секции доступа. Область видимости класса. Инкапсуляция – объектно-ориентированная характеристика модульности.

5. Внешний интерфейс и внутренняя реализация инкапсулированного программного объекта.

6. Характерные признаки эффективной инкапсуляции: абстракция, общедоступный интерфейс и сокрытие реализации.

7. Доступ к элементам класса. Статические элементы класса. Дружественные классы и функции. Пространства имен.

8. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы.

9. Конструктор копирования. Деструктор.

10. Функциональное замыкание как механизм работы специальных методов класса по созданию и уничтожению объектов. Ограничения на использование автоматически сгенерированных компилятором специальных методов.

11. Статический полиморфизм. Перегруженные функции-члены. Концепция перегрузки операторов. Перегруженные операторы как методы класса и как дружественные функции.

12. Перегрузка операторов преобразования типа. Спецификатор *explicit* для конструкторов. Перегрузка операторов *new* и *delete*.

13. Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода. Библиотека потока C++. Общие функции ввода-вывода в поток.

14. Наследование – базовое понятие объектно-ориентированного программирования. Наследование и повторное использование кода. Простое наследование. Синтаксис определения класса-потомка. Правила наследования.

15. Преобразование типов (ссылок и указателей). Правила видимости при простом наследовании. Перекрытие имен и функциональное замыкание в случае простого наследования.

16. Динамический полиморфизм (простое наследование). Реализация виртуальных функций. Перегруженные и переопределённые методы.

17. Раннее и позднее связывание. Пустые и чистые виртуальные функции. Абстрактные классы.

18. Множественное наследование. Синтаксис определения класса-потомка. Видимость при множественном наследовании. Виртуальные базовые классы. Функциональное замыкание в случае множественного наследования.

19. Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы.

20. Динамическая информация о типе при простом и множественном наследовании.

21. Объектно-ориентированная концепция обработки исключений. Блок *try*. Оператор *catch*. Классы исключений. Стандартные исключения.

22. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений).

23. Параметрический полиморфизм для функций. Синтаксис определения шаблона функции. Механизмы генерирования и идентификации при функциональных обращениях.

24. Синтаксис определения шаблона класса. Параметры шаблона. Специализация шаблонов классов. Реализация шаблона класса.

25. Составные части библиотеки. Организация стандартной библиотеки шаблонов. Стандартные контейнеры.
26. Итераторы. Обратные итераторы. Итераторы вставки. Поточковые итераторы.
27. Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели.
28. Связыватели. Адаптеры указателей на функции. Адаптеры методов.
29. Последовательные контейнеры. Векторы (vector). Списки (list). Стеки (stack). Двусторонние очереди (deque).
30. Очереди (queue). Очереди с приоритетами (priority_queue).
31. Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset). Словари (map). Словари с дубликатами (multimap).
32. Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями.
33. Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами.

8. ОБРАЗЕЦ ЗАДАНИЯ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»
Факультет математики и информационных технологий

Направление подготовки: 01.03.02 Прикладная математика и информатика

Программа подготовки: **бакалавриат**

Семестр 3

Учебная дисциплина: Объектно-ориентированное программирование и стандартная библиотека C++
в численных методах исследования моделей деформирования

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА

ВАРИАНТ №1

1. В чем состоит концепция полиморфизма? Как виртуальные функции помогают реализовать динамический полиморфизм?
2. Механизм реализации динамического полиморфизма. Опишите и сравните механизмы «раннего связывания» и «позднего связывания». **Примеры.**
3. Статические и нестатические члены класса, в чем различие? Особенности доступа к ним? Особенности создания и размещения в памяти статических и нестатических членов данных класса? Чем отличается интерфейс статического и нестатического метода класса? **Примеры.**
4. Определить класс **PointOnPlane** (*точка на декартовой плоскости*) и с его использованием класс геометрических фигур **QuadrangleOnPlane** – *четырёхугольник на декартовой плоскости* (четырёхугольник задается координатами четырех его вершин в порядке против часовой стрелки) в котором определить методы: необходимые конструкторы (с проверкой корректности геометрической фигуры) и деструктор; преобразование в тип double (трактуются как вычисление периметра); перегруженные операции сравнения == и != (проверяется совпадение размеров и расположения), сложение *четырёхугольника* и *точки на декартовой плоскости* (трактуются как смещение *четырёхугольника* на заданный *точкой* вектор).

Утверждено на заседании кафедры теории упругости и вычислительной математики
имени академика А.С. Космодамианского, протокол №____ от «__» _____ 20__ г.

Заведующий кафедрой
Преподаватель

Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	10
2	10
3	10
4	10
Всего	40

10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Образовательная программа: бакалавриат

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Статистика

Очная форма обучения. Семестр: 3

Учебная дисциплина: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И
СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ
МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

БИЛЕТ №1

1. Статические и нестатические члены класса, в чем различие? Особенности доступа к ним? Особенности создания и размещения в памяти статических и нестатических членов данных класса? Чем отличается интерфейс статического и нестатического метода класса? **Примеры.**

2. Определить класс **PointOnPlane** (*точка на декартовой плоскости*) и с его использованием класс геометрических фигур **QuadrangleOnPlane** – *четырёхугольник на декартовой плоскости* (четырёхугольник задается координатами четырех его вершин в порядке против часовой стрелки) в котором определить методы: необходимые конструкторы (с проверкой корректности геометрической фигуры) и деструктор; преобразование в тип double (трактуются как вычисление периметра); перегруженные операции сравнения == и != (проверяется совпадение размеров и расположения), сложение *четырёхугольника* и *точки на декартовой плоскости* (трактуются как смещение *четырёхугольника* на заданный *точкой* вектор).

Утверждено на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского

протокол №___ от «__» _____ 20__ г.

Заведующий кафедрой
Экзаменатор

ФИО
ФИО

11. КРИТЕРИИ ОЦЕНИВАНИЯ ЭКЗАМЕНАЦИОННОГО ЗАДАНИЯ

<i>Номер задания</i>	<i>Количество баллов</i>
1	10
2	10
Всего	20

12. КРИТЕРИИ ОЦЕНИВАНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа (включая выполнение СРС) оценивается в 30 баллов.

13. КРИТЕРИИ ОЦЕНИВАНИЯ ОБЩЕЙ УСПЕВАЕМОСТИ

Общая оценка знаний студентов по дисциплине проводится по 100-балльной шкале согласно таким критериям, приведенным в таблице ниже. Согласно модульному принципу организации учебного процесса знания студентов по учебной дисциплине **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА C++ В ЧИСЛЕННЫХ МЕТОДАХ ИССЛЕДОВАНИЯ МОДЕЛЕЙ ДЕФОРМИРОВАНИЯ** оцениваются в соответствии со следующей **СИСТЕМОЙ ОЦЕНИВАНИЯ АКАДЕМИЧЕСКИХ ДОСТИЖЕНИЙ СТУДЕНТОВ**.

Формы контроля по курсу:

- Теоретический опрос по четырем содержательным модулям (СМ) и защита четырех индивидуальных заданий (ИЗ) – промежуточный контроль;
- проверка теоретических знаний и практических умений в режиме тестового модульного контроля по заданиям комплексной контрольной работы (ККР);
- письменный экзамен.

РАСПРЕДЕЛЕНИЕ БАЛЛОВ

МОДУЛЬ 1				МОДУЛЬ 2				ВСЕГО		
Промежуточный контроль		МКР		Промежуточный контроль		МКР				
СМ 1.1		СМ 1.2		Тест		ККР				
12		18		0	15		15		40	100
ТО	ЗП	ТО	ЗП		ТО	ЗП	ТО	ЗП		
12	0	14	4		11	4	11	4		

Сокращения: ТО – теоретический опрос; ЗП – защита программы; МКР – модульная контрольная работа.

Шкала соответствия баллов национальной шкале

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (зачет)
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено

C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено
FX	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

Знание теоретической части курса оценивается с точностью до 5 баллов по следующим критериям:

1. Студент получает 76-100% баллов от максимального, если показал
 - глубокие и полные ответы на теоретические вопросы; глубокое понимание возможности применения теоретических положений в практических задачах;
 - умение проводить логические рассуждения и обобщения и сопровождать их соответствующими доказательствами;
2. Студент получает 51-75% баллов от максимального, если показал
 - глубокие и полные ответы на теоретические вопросы с незначительными погрешностями, затем исправленными самим студентом; понимание сущности рассматриваемых проблем;
 - умение логически рассуждать и проводить доказательства;
3. Студент получает 26-50% баллов от максимального, если показал
 - при ответе на теоретические вопросы ряд неточностей, которые он не в состоянии самостоятельно исправить;
4. Студент получает 0-25% баллов от максимального, если
 - не выполнены требования, изложенные в предыдущих пунктах;
 - нет ответов на теоретические вопросы, не решены практические задачи.

14. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Учебные занятия проводятся в главном (г. Донецк, пр. Гурова, д. 6) и двенадцатом (г. Донецк, ул. Университетская, 24-а, УПВЦ) учебных корпусах университета. Лекционные занятия проводятся в аудитории, оснащенной доской. Лабораторные занятия проводятся в специализированном компьютерном классе, оборудованном компьютерами с лицензионным программным обеспечением, доступом к сети Интернет, столами, доской. Выход в Интернет проводной или с использованием Wi-Fi.

Для самостоятельной работы используются текстовые и электронные ресурсы Научной библиотеки университета и других электронных библиотечных баз данных, учебно-методическое обеспечение, представленное в учебно-методическом кабинете главного корпуса (ауд.605 и 610).

15. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электрон ной версии в ЭБС
Основная литература			
1.	Буч Гради. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Гради Буч ; Пер. с англ. под ред. И. Романовского, Ф.	2	

	Андреева. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. – 560 с.		
2.	Иванова Г.С.. Объектно-ориентированное программирование : Учеб. для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / Г.С. Иванова, Т.Н. Ничушкина, Е.К. Пугачев; Под ред. Г.С. Ивановой. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2001. - 317 с.	2	
3.	Калоеров, С. А. Программирование на языке C++ : Учеб. пособие / С. А. Калоеров ; Донец. нац. ун-т. - Донецк : Юго-Восток, 2002. - 224 с.	3	
4.	Калоеров, С. А. Программирование на языке C++ : Учеб. пособие / С. А. Калоеров ; Донец. нац. ун-т. - 2-е изд. - Донецк : Юго-Восток, 2004. - 237 с.	65	
5.	Калоеров, С. А. Программирование на языке C++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. - Донецк : Юго-Восток, 2009. - 298 с.	92	
6.	Лафоре, Р. Объектно-ориентированное программирование в C++ [Текст] / Р. Лафоре; [пер. с англ. А. Кузнецова]. - 4-е изд. - М. [и др.] : Питер, 2008. - 923 с.	2	
7.	Леоненко, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учеб. пособие / А. В. Леоненков. - М. : Интернет-ун-т информ. технологий : Бином. Лаб. знаний, 2006. - 320 с.	4	
8.	Павловская, Т. А. C/C++ Программирование на языке высокого уровня : учебник для вузов по направлению "Информатика и вычисл. техника" / Т. А. Павловская. - М. и др. : Питер, 2008. - 461 с.	2	
9.	Павловская, Т. А. C/C++. Программирование на языке высокого уровня : учеб. для студентов вузов, обучающихся по направлению "Информатика и вычислит. техника" / Т. А. Павловская. - Москва [и др.] : Питер, 2009. - 460 с.	22	
10.	Павловская, Т. А. C/C++. Программирование на языке высокого уровня : учеб. для студентов вузов, обучающихся по направлению "Информатика и вычислит. техника" / Т. А. Павловская. - Москва [и др.] : Питер, 2010. - 460 с.	32	
11.	Пол, Айра. Объектно-ориентированное программирование на C++ / Айра Пол; Пер. с англ. Д. Ковальчука. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. - 464 с.	2	
12.	Страуструп, Б. Язык программирования Си ++ / Б. Страуструп ; пер. с англ. М. Г. Пиголкина, В. А. Яницкого. - Москва : Радио и связь, 1991. - 348 с.	4	
13.	Страуструп, Б. Язык программирования C++ / Бьерн Страуструп ; пер. с англ. С. Анисимова, М. Кононова ; под ред. Ф. Андреева, А. Ушакова. - спец. изд. - М. : Бином-Пресс, 2008. - 1098 с.	1	

14.	Буч, Гради. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Гради Буч ; Пер. с англ. под ред. И. Романовского, Ф. Андреева. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. – 560 с.	2	
<i>Дополнительная литература</i>			
15.	Кнут, Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 1 : Основные алгоритмы / Пер. с англ. Г. П.Бабенко, Ю. М.Баяковского ; Под ред. Г. П. Бабенко, В. С. Штаркмана. - М. : Мир, 1976. - 735 с.	16	
16.	Кнут, Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 2 : Получисленные алгоритмы / Пер. с англ. Г. П. Бабенко и др. ; Под ред. Г. П. Бабенко. - М. : Мир, 1977. - 724 с.	18	
17.	Кнут, Д. Э. Искусство программирования для ЭВМ : [В 7 т.] : Пер. с англ. Т. 3 : Сортировка и поиск / Пер. с англ. Н. И. Вьюковой и др. ; Под ред. Ю. М. Баяковского. - М. : Мир, 1978. - 844 с.	22	

1. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

(с указанием названия и полного электронного адреса)

1. Основы программирования на языке С++: Учебное пособие
http://tk.ulstu.ru/lib/books/lang_c_1.pdf
2. Использование визуальных компонент в С++ Builder: методические указания к лабораторным работам по программированию
http://pnu.edu.ru/media/filer_public/2013/03/04/mu_builder.pdf
3. Структуры данных и алгоритмы: программирование на языке С++: Учеб, пособие в 2 ч. Часть 1 <https://studfiles.net/preview/6324253/>
4. Краткий справочник по языку программирования с++
<http://dspace.univer.kharkov.ua/bitstream/123456789/1356/2/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%A3%D0%BA%D0%B0%D0%B7%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%2B%D0%A4.pdf>
5. Вестник Донецкого национального университета. Серия А: Естественные науки
<http://donnu.ru/vestnikA/archive>

2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Windows 7 PRO (корпоративная лицензия ДОННУ № 46484614);
2. Microsoft Office (корпоративная лицензия ДОННУ лицензия № 46472919);

Рабочая программа рассмотрена и переутверждена на заседании теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20_____год.

Протокол №____от «_____»_____20____г.

Заведующий кафедрой _____