ГОУ ВПО «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
«22» апреля 2020 г.
МП

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И СТАНДАРТНАЯ БИБЛИОТЕКА С++»

Направление подготовки:	01.03.02 Прикладная математика и информатика
Образовательная программа:	бакалавриат
Квалификация:	Академический бакалавр
Форма обучения:	очная, очно-заочная, заочная, в том числе с ускоренным сроком обучения

Донецк 2020

УТВЕРЖДАЮ:
Декан факультета математики
и информационных технологий
_____ И. А. Моисеенко
«16» апреля 2020
МП

Программа учебной дисциплины «Объектно-ориентированное программирование и стандартная библиотека C++» составлена на основании Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) Донецкой Народной Республики (ДНР) по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки ДНР от «04» апреля 2016 г. № 280;
Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки ДНР № 1171 от «10» ноября 2017 г.;
учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 01.03.02 Прикладная математика и информатика, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:
Профессор кафедры теории упругости и
вычислительной математики имени
академика А.С. Космодамианского _____ И. А. Моисеенко

Программа учебной дисциплины утверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского

Протокол № 11 от «9» апреля 2020 г.
Заведующий кафедрой _____ В.И. Сторожев

Программа учебной дисциплины одобрена учебно-методической комиссией факультета математики и информационных технологий
Протокол № 8 от «15» апреля 2020 г.

Председатель учебно-методической
комиссии факультета _____ Л.И. Селякова

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

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.

Освоение курсов «Основы информатики», «Языки и методы программирования».

2. СТРУКТУРА ДИСЦИПЛИНЫ

Характеристика учебной дисциплины		
Направление подготовки	01.03.02 Прикладная математика и информатика	
Профиль	Общий	
Образовательная программа	бакалавриат	
Квалификация	Академический бакалавр	
Количество содержательных модулей	3 (15)	
Дисциплина базовой / вариативной части образовательной программы	Вариативная часть, Профессиональный блок	
Формы контроля (МК, экзамен, зачет)	модульный контроль, экзамен	
Показатели	очная форма обучения	заочная форма обучения

	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачетных единиц (кредитов)	5	5		
Год подготовки	2	2		
Семестр	3	3		
Количество часов	180	180		
- лекционных	54	54		
- практических, семинарских	-	-		
- лабораторных	36	36		
- самостоятельной работы	90	90		
в т.ч. индивидуальное задание	0	0		
Недельное количество часов,	10	10		
в т.ч. аудиторных	5	5		

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цели и задачи

Цели учебной дисциплины.

1. 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.
2. 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.
3. Изучение средств объектно-ориентированного и обобщенного программирования языка C++.
4. 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.

1. 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 направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО по направлению подготовки 01.03.02 «Прикладная математика и информатика»:

а) общекультурных (ОК):

- способностью к самоорганизации и самообразованию (ОК-7);

б) общепрофессиональных (ОПК):

- способностью использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой (ОПК-1);
- способностью приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии (ОПК-2);
- способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3);
- способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-4);

в) профессиональных (ПК):

научно-исследовательская деятельность:

- способностью понимать, совершенствовать и применять современный математический аппарат (ПК-2);

проектная и производственно-технологическая деятельность:

- способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7);

организационно-управленческая деятельность:

способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы (ПК-9);

В результате изучения учебной дисциплины студент должен:

знать:

- основные теоретические понятия объектно-ориентированного подхода к программированию;
- синтаксические и семантические аспекты реализации объектно-ориентированного подхода в языке программирования C++;
- основные этапы проектирования объектно-ориентированных приложений;
- достоинства и недостатки объектной технологии программирования;
- тенденции и перспективы развития объектно-ориентированного подхода в программировании;
- основные структурные элементы STL;
- синтаксические и семантические аспекты реализации программ на C++ с использованием STL;
- достоинства и недостатки использования основных структурных элементов STL;

- стандартные контейнерные классы, реализующие наиболее распространенные структуры для хранения данных - вектора, деки, списки, множества, мультимножества, отображения, мультиотображения;
- стандартные универсальные алгоритмы, использующие эти контейнеры;

итераторы, предназначенные для организации унифицированного доступа к элементам этих контейнеров;

уметь:

- анализировать предметную область решаемых задач с целью использования объектно-ориентированного подхода для их реализации, разрабатывать объектную модель программы;
- выбирать методы и средства для реализации программных проектов с использованием объектно-ориентированного программирования;
- разрабатывать алгоритмы применительно к методу объектно-ориентированного программирования;
- 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;
- анализировать предметную область решаемых задач с целью использования доступных структурных элементов STL в разрабатываемой объектной модели программы;
- выбирать методы и средства для реализации программных проектов с использованием структурных элементов STL;
- на основе объектно-ориентированного подхода решать практические задачи программирования на языке C++ с использованием:
  - стандартных контейнеров STL;
  - стандартных универсальных алгоритмов, обрабатывающих эти контейнеры;

владеть:

- одним из современных языков программирования применительно к технологии объектно-ориентированного программирования;
- навыками использования STL при разработке приложений на C++ для моделей реальных и концептуальных систем, соответствующих направлению подготовки «Прикладная математика и информатика».

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

В рамках изучения дисциплины предусмотрены следующие формы организации учебного процесса: лекции, лабораторные занятия, самостоятельную работу студента.

Лекционные занятия направлены на овладение теоретическими основами дисциплины, лабораторные – на овладения методами (алгоритмами) практического решения задач профессиональной деятельности.

Самостоятельная работа студентов предусматривает выполнение домашних индивидуальных заданий, подготовку к лекционным и лабораторным занятиям, изучение учебно-методической литературы, составление конспектов, подготовку презентаций и докладов.

Текущий контроль осуществляется путем защиты индивидуальных заданий, модульной контрольной работы по проверке знаний теоретических положений (понятий, определений, алгоритмов и т.д.).

В учебном процессе применяются активные и интерактивные формы проведения занятий, внеаудиторная самостоятельная работа, балльно-рейтинговая система оценки успеваемости, личностно-ориентированное обучение, проблемное обучение.

Материал излагается с использованием объяснительно-иллюстративных, эвристических и исследовательских методов преподавания. При проведении лекций для обсуждения материала широко используются методы математического и компьютерного моделирования. Также проводятся лекции проблемные, с заранее запланированными ошибками.

Порядковый номер и тема	**Краткое содержание 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 *explicit* для конструкторов. Перегрузка операторов *new* и *delete*.
Тема 4. **Потоки**	Файловые и строковые потоки. Средства ввода-вывода и работа с потоками. Перегрузка операторов потокового ввода/вывода. Библиотека потока С++. 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. Перекрытие 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 *try*. Оператор *catch*. 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**	Итераторы. Обратные итераторы. Итераторы вставки. Потоковые 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.

Тематический план

Названия содержательных модулей и тем	Количество часов																						
	Очная форма обучения												Заочная форма обучения										
	Нормативный срок обучения						Ускоренный срок обучения						Нормативный срок обучения						Ускоренный срок обучения				
	всего	в т.ч.					всего	в т.ч.					всего	в т.ч.					всего	в т.ч.			
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	самостоятельная работа	индивидуальная работа
Содержательный модуль 1.1. Инкапсуляция.																							
Тема 1. **Объектно-ориентированная парадигма**	**8**	4	0	0	4	0	**8**	4	0	0	4	0											
Тема 2. **Базовые классы**	**28**	8	0	6	14	0	**28**	8	0	6	14	0											
Итого по содержательному модулю 1.1.	**36**	**12**	**0**	**6**	**18**	**0**	**36**	**12**	**0**	**6**	**18**	**0**											
Содержательный модуль 1.2. Статический полиморфизм.																							
Тема 3. **Перегрузка операторов**	**13**	4	0	3	6	0	**13**	4	0	3	6	0											
Тема 4. **Потоки**	**5**	2	0	1	2	0	**5**	2	0	1	2	0											
Итого по содержательному модулю 1.2.	**18**	**6**	**0**	**4**	**8**	**0**	**18**	**6**	**0**	**4**	**8**	**0**											

Содержательный модуль 2.1. Наследование и динамический полиморфизм.																							
Тема 5. **Наследование**	**12**	4	0	1	7	0	**12**	4	0	1	7	0											
Тема 6. **Динамический полиморфизм**	**24**	8	0	3	13	0	**24**	8	0	3	13	0											
Итого по содержательному модулю 2.1	**36**	**12**	**0**	**4**	**20**	**0**	**36**	**12**	**0**	**4**	**20**	**0**											
Содержательный модуль 2.2. Шаблоны и параметрический полиморфизм.																							
Тема 7. **Параметрический полиморфизм для функций**	**4**	2	0	0	2	0	**4**	2	0	0	2	0											
Тема 8. **Параметрический полиморфизм для классов**	**8**	2	0	3	3	0	**8**	2	0	3	3	0											
Итого по содержательному модулю 2.2.	**12**	**4**	**0**	**3**	**5**	**0**	**12**	**4**	**0**	**3**	**5**	**0**											
Содержательный модуль 3.1. Архитектура стандартной библиотеки C++.																							
Тема 9. **Архитектура библиотеки**	**6**	2	0	1	3	0	**6**	2	0	1	3	0											
Итого по содержательному модулю 3.1.	**6**	**2**	**0**	**1**	**3**	**0**	**6**	**2**	**0**	**1**	**3**	**0**											
Содержательный модуль 3.2. Итераторы и функциональные объекты.																							
Тема 10. **Итераторы**	**6**	2	0	0	4	0	**6**	2	0	0	4	0											
Тема 11. **Функциональные объекты**	**12**	2	0	2	8	0	**12**	2	0	2	8	0											
Итого по содержательному модулю 3.2.	**18**	**4**	**0**	**2**	**12**	**0**	**18**	**4**	**0**	**2**	**12**	**0**											

Содержательный модуль 3.3. Контейнеры.																							
Тема 12. **Последовательные контейнеры**	**19**	4	0	6	9	0	**19**	4	0	6	9	0											
Тема 13. **Ассоциативные контейнеры**	**17**	4	0	6	7	0	**17**	4	0	6	7	0											
Итого по содержательному модулю 3.3.	**36**	**8**	**0**	**12**	**16**	**0**	**36**	**8**	**0**	**12**	**16**	**0**											
Содержательный модуль 3.4. Алгоритмы.																							
Тема 14. **Операции с последовательностями**	**8**	2	0	2	4	0	**8**	2	0	2	4	0											
Тема 15. **Специальные алгоритмы**	**10**	4	0	2	4	0	**10**	4	0	2	4	0											
Итого по содержательному модулю 3.4.	**18**	**6**	**0**	**4**	**8**	**0**	**18**	**6**	**0**	**4**	**8**	**0**											
Всего по дисциплине	**180**	**54**	**0**	**36**	**90**	**0**	**180**	**54**	**0**	**36**	**90**	**0**											

5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Темы лекционных занятий

№ п/п	***Название темы***	***Количество часов***
1.	*Объектно-ориентированная парадигма*	4
2.	*Базовые классы*	8
3.	*Перегрузка операторов*	4
4.	*Потоки*	2
5.	*Наследование*	4
6.	*Динамический полиморфизм*	8
7.	*Параметрический полиморфизм для функций*	2
8.	*Параметрический полиморфизм для классов*	2
9.	*Архитектура библиотеки*	2
10.	*Итераторы*	2
11.	*Функциональные объекты*	2
12.	*Последовательные контейнеры*	4
13.	*Ассоциативные контейнеры*	4
14.	*Операции с последовательностями*	2
15.	*Специальные алгоритмы*	4
	ВСЕГО	**54**

Темы лабораторных занятий

№ п/п	***Название темы***	***Количество часов***
1.	*Базовые классы*	6
2.	*Перегрузка операторов*	3
3.	*Потоки*	1
4.	*Наследование*	1
5.	*Динамический полиморфизм*	3
6.	*Параметрический полиморфизм для классов*	3
7.	*Архитектура библиотеки*	1
8.	*Функциональные объекты*	2
9.	*Последовательные контейнеры*	6
10.	*Ассоциативные контейнеры*	6
11.	*Операции с последовательностями*	2
12.	*Специальные алгоритмы*	2
	ВСЕГО	**36**

6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Организация самостоятельной работы студентов

№ п/п	*Название темы*	*Количество часов*
1.	*Объектно-ориентированная парадигма*	4
2.	*Базовые классы*	14
3.	*Перегрузка операторов*	6
4.	*Потоки*	2
5.	*Наследование*	7
6.	*Динамический полиморфизм*	13
7.	*Параметрический полиморфизм для функций*	2
8.	*Параметрический полиморфизм для классов*	3
9.	*Архитектура библиотеки*	3
10.	*Итераторы*	4
11.	*Функциональные объекты*	8
12.	*Последовательные контейнеры*	9
13.	*Ассоциативные контейнеры*	7
14.	*Операции с последовательностями*	4
15.	*Специальные алгоритмы*	4
	ВСЕГО	**90**

7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

(не *предусмотрено программой*)

8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Краткий обзор основных парадигм программирования. 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. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.

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.

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

5. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор.

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

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

8. Перегрузка операторов преобразования типа. Спецификатор *explicit* для

конструкторов. Перегрузка операторов *new* и *delete*.

9. 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.

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

11. Преобразование типов (ссылок и указателей). Правила видимости 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.

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

14. Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании.

15. Объектно-ориентированная концепция обработки исключений. Блок *try*. Оператор *catch*. Классы исключений. Стандартные исключения. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений).

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

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

18. Составные части библиотеки. Организация стандартной библиотеки шаблонов. Стандартные контейнеры.

19. Итераторы. Обратные итераторы. Итераторы вставки. Потоковые итераторы.

20. Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели.

21. Связыватели. Адаптеры указателей на функции. Адаптеры методов.

22. Последовательные контейнеры. Векторы (vector). Списки (list). Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority_queue).

23. Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset). Словари (map). Словари с дубликатами (multimap).

24. Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями.

25. Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами.

26. Другие средства стандартной библиотеки.

9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

(*образец варианта и критерии оценивания***)**

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++***

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА
ВАРИАНТ №1

1. В чем состоит концепция полиформизма? Как виртуальные функции помогают реализовать динамический полиформизм? Механизм реализации динамического полиформизма. Опишите и сравните механизмы «раннего связывания» и «позднего связывания». **Примеры**.
2. 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**.
3. Определить класс **PointOnPlane** (*точка на декартовой плоскости*) и с его использованием класс геометрических фигур **QuadrangleOnPlane** – *четырехугольник на декартовой плоскости* (четырехугольник 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 (трактуется как вычисление периметра); перегруженные операции сравнения **==** и **!=** (проверяется совпадение размеров и расположения), сложение *четырехугольника* и *точки на декартовой плоскости* (трактуется как смещение *четырехугольника* на заданный *точкой* вектор).

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

Заведующий кафедрой Сторожев В. И.
Преподаватель Моисеенко И. А.

Критерии оценивания модульного контроля

Номер задания	***Количество баллов***
1	10
2	10
3	20
Всего баллов	**40**

10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Теоретические вопросы к экзамену

1. Краткий обзор основных парадигм программирования. 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. Принципы объектно-ориентированного подхода. Этапы объектно-ориентированной методологии.

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.

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

5. Специальные методы класса. Назначение конструкторов и деструктора. Конструктор по умолчанию. Параметризованные конструкторы. Конструктор копирования. Деструктор.

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

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

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

9. 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.

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

11. Преобразование типов (ссылок и указателей). Правила видимости 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.

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

14. Динамический полиморфизм (множественное наследование). Интерфейсы. Абстрактные классы. Динамическая информация о типе при простом и множественном наследовании.

15. Объектно-ориентированная концепция обработки исключений. Блок *try*. Оператор *catch*. Классы исключений. Стандартные исключения. Последовательность действий при возникновении исключительной ситуации (генерирование и переброска исключений).

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

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

18. Составные части библиотеки. Организация стандартной библиотеки шаблонов. Стандартные контейнеры.

19. Итераторы. Обратные итераторы. Итераторы вставки. Потоковые итераторы.

20. Функциональные объекты. Арифметические функциональные объекты. Предикаты. Отрицатели.

21. Связыватели. Адаптеры указателей на функции. Адаптеры методов.

22. Последовательные контейнеры. Векторы (vector). Списки (list). Стеки (stack). Двусторонние очереди (deque). Очереди (queue). Очереди с приоритетами (priority_queue).

23. Ассоциативные контейнеры. Множества (set). Множества с дубликатами (multiset). Битовые множества (bitset). Словари (map). Словари с дубликатами (multimap).

24. Алгоритмы. Немодифицирующие операции с последовательностями. Модифицирующие операции с последовательностями.

25. Алгоритмы, связанные с сортировкой. Алгоритмы работы с множествами и пирамидами.

26. Другие средства стандартной библиотеки.

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++***

БИЛЕТ №1

1. В чем состоит концепция полиформизма? Как виртуальные функции помогают реализовать динамический полиформизм? Механизм реализации динамического полиформизма. Опишите и сравните механизмы «раннего связывания» и «позднего связывания». **Примеры.**

2. 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.**

3. Определить класс **PointOnPlane** (*точка на декартовой плоскости*) и с его использованием класс геометрических фигур **QuadrangleOnPlane** – *четырехугольник на декартовой плоскости* (четырехугольник 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 (трактуется как вычисление периметра); перегруженные операции сравнения == и **!=** (проверяется совпадение размеров и расположения), сложение *четырехугольника* и *точки на декартовой плоскости* (трактуется как смещение *четырехугольника* на заданный *точкой* вектор).

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

Заведующий кафедрой Сторожев В. И.
Экзаменатор Моисеенко И. А.

Критерии оценивания экзамена

Номер задания	***Количество баллов***
1	10
2	10
3	20
Всего баллов	**40**

11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ – *не предусмотрено программой-*

12. КРИТЕРИИ ОЦЕНИВАНИЯ

По курсу предполагается проведение промежуточной аттестации в виде модульного контроля, выполнения индивидуальной творческой работы и экзамена. Экзамен сдают студенты с целью повышения рейтинга.

Распределение баллов, которые могут получить студенты в процессе изучения дисциплины

Организационно-учебная работа студента	**СРС**		**Всего**
	Модульный контроль	**Индивидуальная творческая работа**	
Max 100 баллов	max 40 баллов	max 60 баллов	100 баллов
		разработка классов	

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

13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Лекционные занятия проводятся в аудитории, оснащенной мультимедийной техникой и доской.

Лабораторные занятия проводятся в компьютерном классе, оборудованном компьютерами с лицензионным программным обеспечением, доступом к сети Интернет, столами, доской.

14. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	**Наименование**	**Кол-во экземпляров в библиотеке ДонНУ**	**Наличие электронной 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. Андреева. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. – 560 с.	2	

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 С++ : Учеб. пособие / С. А. Калоеров ; Донец. нац. ун-т. - Донецк : Юго-Восток, 2002. - 224 с.	3	
4.	Калоеров, С. А. Программирование на языке С++ : 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 С++ : 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 С++ [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. С/С++ 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. С/С++. 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. С/С++. 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 С++ / Айра Пол; Пер. с англ. Д. Ковальчука. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. - 464 с.	2	
12.	Страуструп, Б. Язык программирования Cu ++ / Б. Страуструп ; пер. с англ. М. Г. Пиголкина, В. А. Яницкого. - Москва : Радио и связь, 1991. - 348 с.	4	
13.	Страуструп, Б. Язык программирования С++ / 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++ /	2	

	Гради Буч ; Пер. с англ. под ред. И. Романовского, Ф. Андреева. - 2-е изд. - М. : БИНОМ ; СПб. : Невский диалект, 1999. – 560 с.		
Дополнительная литература			
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	

15. 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%80C%2B%2BA4.pdf
5. Вестник Донецкого национального университета. Серия А: Естественные науки http://donnu.ru/vestnikA/archive

16. 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 кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20____ год.

Протокол № ___ от «__ ___» _____ 20__ __ г.

Заведующий кафедрой Сторожев В. И.

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

Протокол № ___ от «_____» __________ 20____ г.

Заведующий кафедрой Сторожев В. И.

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

Протокол № ___ от «_____» __________ 20____ г.

Заведующий кафедрой Сторожев В. И.

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

Протокол № ___ от «_____» __________ 20____ г.

Заведующий кафедрой Сторожев В. И.

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

Протокол № ___ от «_____» __________ 20____ г.

Заведующий кафедрой Сторожев В. И.