

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ, ФИЗИКИ И ИНФОРМАТИКИ
(Наименование института)

УТВЕРЖДАЮ:

Директор института

Хорьков К.С.

08 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ
(наименование дисциплины)

направление подготовки / специальность

01.03.02. «Прикладная математика и информатика»
(код и наименование направления подготовки (специальности))

направленность (профиль) подготовки

Математическое и компьютерное моделирование, программирование и системный анализ
(направленность (профиль) подготовки))

г. Владимир

2021 год

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ** является изучение объектно-ориентированного языка программирования C++, базовых принципов построения объектно-ориентированного программного кода.

Задачи:

- изучение теоретической части (объектно-ориентированные возможности языка программирования Си++, основные свойства объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм)
- закрепление на практике полученных теоретических знаний (разработка небольших программ и полноценных клиентских приложений на языке Си++)

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ** относится к обязательным дисциплинам блока Блок 1. Дисциплины (модули) учебного плана.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1. Знает математические основы, основные положения и концепции в области программирования.	<p>Знает:</p> <ul style="list-style-type: none"> - расширенные знания в области математики; - математические основы, основные положения и концепции в области программирования; - архитектура языков программирования; - основная терминология в области программного обеспечения. <p>Умеет:</p> <ul style="list-style-type: none"> - умеет осуществлять обоснованный выбор математических и компьютерных методов, а также необходимого программного обеспечения при решении задач профессиональной деятельности. <p>Владеет:</p> <ul style="list-style-type: none"> - имеет навыки применения данных методов и программного обеспечения при решении конкретных задач. 	<p>-- вопросы к текущему контролю знаний и промежуточной аттестации</p> <p>-- отчеты по лабораторным работам</p> <p>-- отчет по заданию в рамках самостоятельной работы в семестре</p>
	ОПК-2.1. Умеет осуществлять обоснованный выбор математических и компьютерных методов, а также необходимого программного обеспечения при решении задач профессиональной деятельности.		
ОПК-4. Способен	ОПК-4.1. Знает принципы	<p>Знает:</p> <ul style="list-style-type: none"> - 	<p>-- вопросы к</p>

<p>понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</p>	<p>работы и использования современных информационных технологий профессиональной деятельности, основные требования информационной безопасности. ОПК-4.2. Умеет осуществлять обоснованный выбор необходимых информационных технологий при решении задач профессиональной деятельности. ОПК-4.3. Владеет практическими навыками использования информационно-коммуникационных технологий при решении задач профессиональной деятельности.</p>	<p>– основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей; – современные языки программирования; – технологии создания и эксплуатации программных продуктов и программных комплексов – основные требования информационной безопасности. Умеет: – умеет осуществлять обоснованный выбор необходимых информационных технологий при решении задач профессиональной деятельности. Владеет: – имеет практические навыки использования информационно-коммуникационных технологий при решении задач профессиональной деятельности.</p>	<p>– текущему контролю знаний и промежуточной аттестации – отчеты по лабораторным работам – отчет по заданию в рамках самостоятельной работы в семестре</p>
<p>ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.</p>	<p>ОПК-5.1. Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей. ОПК-5.2. Умеет разрабатывать алгоритмы и компьютерные программы с использованием современных языков и инструментов программирования. ОПК-5.3. Владеет технологиями создания и эксплуатации программных продуктов и программных комплексов.</p>	<p>Знает: – основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей; Умеет: – разрабатывать алгоритмы и компьютерные программы с использованием современных языков и инструментов программирования. Владеет: – технологиями создания и эксплуатации программных продуктов и программных комплексов.</p>	<p>– вопросы к текущему контролю знаний и промежуточной аттестации – отчеты по лабораторным работам – отчет по заданию в рамках самостоятельной работы в семестре</p>
<p>ПК-4. Способен анализировать требования к программному средству</p>	<p>ПК-4.1. Знает методы анализа и тестирования требований к программному средству, методы анализа, проектирования и разработки программного</p>	<p>Знает: – методы анализа, проектирования и разработки программного обеспечения; Умеет: – тестировать требования к программному средству.</p>	<p>– вопросы к текущему контролю знаний и промежуточной аттестации – отчеты по лабораторным</p>

	<p>обеспечения. ПК-4.2. Умеет тестировать требования к программному средству, оформлять документацию по тестированию, анализировать требования на соответствие принятым стандартам и методам проектирования, использовать современные CASE-средства.</p> <p>ПК-4.3. Владеет навыками проверки осуществимости функционирования и сопровождения программного средства, определения возможности введения изменений и дополнений требований к программному средству.</p>	<p>Владеет:</p> <p>– навыками сопровождения программного средства.</p>	<p>– работам отчет по заданию в рамках самостоятельной работы в семестре</p>
<p>ПК-1. Способен проектировать и реализовывать программное обеспечение в соответствии с требованиями</p>	<p>ПК-1.1. Знает методологии разработки программного обеспечения и технологии программирования, языки формализации функциональных спецификаций, принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, основные концепции и атрибуты качества программного обеспечения.</p> <p>ПК-1.2. Умеет проводить анализ исполнения требований, вырабатывать варианты их реализации, проводить оценку и обоснование рекомендуемых решений, выбирать и использовать средства и варианты реализации программного обеспечения.</p> <p>ПК-1.3. Владеть навыками оценки возможностей, времени и трудоемкости реализации</p>	<p>Знает:</p> <p>– возможности современных и перспективных средств разработки программных продуктов, технических средств;</p> <p>– методологии разработки программного обеспечения и технологии программирования;</p> <p>– методологии и технологии проектирования и использования баз данных;</p> <p>– методы и средства проектирования программного обеспечения и баз данных;</p> <p>– языки формализации функциональных спецификаций;</p> <p>– принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения;</p> <p>– типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения;</p> <p>– основные концепции и</p>	<p>– вопросы текущего контролю знаний и промежуточной аттестации отчеты по лабораторным работам отчет по заданию в рамках самостоятельной работы в семестре</p>

		написания кода.	
--	--	-----------------	--

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

Тематический план форма обучения – очная

№ п/п	Наименование тем и/или разделов/тем дисциплины	Семестр	Неделя семестра	Контактная работа обучающихся с педагогическим работником				Самостоятельная работа	Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	в форме практической подготовки		
1	Раздел 1. Многомодульное программирование. Классы памяти	3	1-4	4	-	8	10	2	Рейтинг-контроль №1
2	Раздел 2. Основы объектно-ориентированного проектирования. Основные объектно-ориентированные возможности языка Си++.	3	3-8	12	-	8	10	2	
3	Раздел 3. Работа с потоками данных (stream) в Си++	3	9-10	2	-	4	13	9	Рейтинг-контроль №2
4	Раздел 4. Вмещающие классы. Шаблоны классов (родовые классы). Стандартная библиотека шаблонов С++ (STL).	3	10-14	10	-	8	10	2	
5	Раздел 5. Наследование. Раннее и позднее связывание. Полиморфизм.	3	15-18	8	-	8	10	2	Рейтинг-контроль №3
6	Раздел 6. Разработка пользовательских приложений с оконным интерфейсом	3	1-18	-	-	-	100	100	
Всего за 3 семестр		-	-	36	-	36	153	117	
Наличие в дисциплине КП/КР		-	-	-	-	-	-	-	
Итого по дисциплине		-	-	36	-	36	-	117	Экзамен (27ч)

Содержание лекционных занятий по дисциплине

Раздел 1 Многомодульное программирование, Классы памяти

Тема 1 Многомодульное программирование

Понятие модуля. Структура многомодульного проекта. Препроцессинг, компиляция, компоновка (линковка). Способы компиляции (совместная, отдельная).

Тема 2 Классы памяти

Классы памяти `auto`, `register`, `static`, `extern`

Раздел 2 Основы объектно-ориентированного проектирования. Основные объектно-ориентированные возможности языка Си++.

Тема 1 Основы объектно-ориентированного проектирования

Понятие объекта и класса. Понятие объектной модели задачи. Инкапсуляция.

Тема 2 Классы в Си++

Синтаксис объявления классов, элементы класса (поля и методы). Конструкторы (с параметрами; по умолчанию; с параметрами, заданными по умолчанию; конструкторы копирования). Деструкторы. Модификаторы видимости. Доступ к элементам класса. Указатель `this`.

Тема 2 Объекты и массивы объектов в Си++.

Создание статических и динамических объектов. Способы инициализации объектов. Создание статических и динамических массивов объектов. Способы инициализации статических и динамических массивов объектов.

Тема 2 Дружественные функции, дружественные классы в Си++.

Синтаксис объявления дружественных функций и дружественных классов. Специфика использования дружественных функций и дружественных классов в объектно-ориентированном программном коде.

Тема 4 Перегрузка операторов в Си++.

Синтаксис. Способы вызова перегруженного оператора. Переопределяемые и непереопределяемые операторы. Перегрузка унарных и бинарных операторов методами класса и дружественными функциями.

Тема 5 Статические элементы класса в Си++.

Понятие статических элементов класса (полей, методов). Синтаксис объявления и использования статических элементов класса.

Раздел 3 Работа с потоками данных (`stream`) в Си++

Тема 1 Потоки данных (`stream`)

Понятие потока данных. Текстовые и бинарные потоки данных.

Тема 2 Библиотека стандартных классов для работы с потоками данных в Си++

Структура библиотеки. Классы для работы с входными, выходными и двунаправленными потоками данных. Предопределенные объекты для работы со стандартными потоками ввода\вывода.

Раздел 4 Вмещающие классы. Шаблоны классов (родовые классы). Стандартная библиотека шаблонов С++ (STL)

Тема 1 Вмещающие классы

Контейнерные структуры данных: классификация и принципы организации. Перегрузка операторов `new` и `delete` для реализации вмещающего класса.

Тема 2 Шаблоны классов (родовые классы) в Си++

Назначение родовых классов, синтаксис объявления, особенности компиляции и особенности использования шаблонов классов в многомодульных проектах.

Тема 3 Стандартная библиотека шаблонов С++ (STL)

Структура библиотеки. Контейнеры и итераторы библиотеки STL

Раздел 5 Наследование. Раннее и позднее связывание. Полиморфизм.

Тема 1 Одиночное наследование

Объявление производного класса. Управление видимостью унаследованными элементами. Список инициализаторов. Порядок вызова конструкторов при создании объекта производного класса. Особенности работы с объектами

классов, находящимися в одной иерархии наследования, в том числе через указатели. Конструкторы копирования в производных классах.

Тема 2 Множественное наследование.

Виды множественного наследования (вертикальное, горизонтальное, смешанное). Проблемы множественного наследования. Виртуальные базовые классы.

Тема 3 Раннее и позднее связывание, полиморфизм.

Понятие связывания. Проблемы раннего связывания. Позднее связывание; реализация механизма позднего связывания в Си++ (виртуальные функции, ТВМ, поле ТВМ в объекте). Проблемы множественного наследования для классов с виртуальными методами. Полиморфизм. Виртуальные деструкторы. Абстрактные классы. Указатели на компоненты класса.

Содержание лабораторных занятий по дисциплине

Раздел 1 Многомодульное программирование, классы памяти

Тема 1 Многомодульное программирование.

Тема 2 Классы памяти

Лабораторная работа № 1. Разработка многомодульного проекта. Работа с классами памяти auto, static, extern. Реализация совместной и отдельной компиляции многомодульного проекта.

Раздел 2 Основы объектно-ориентированного проектирования. Основные объектно-ориентированные возможности языка Си++.

Тема 1 Основы объектно-ориентированного проектирования

Тема 2 Классы в Си++

Тема 3 Объекты и массивы объектов в Си++.

Тема 4 Дружественные функции, дружественные классы в Си++.

Тема 5 Перегрузка операторов в Си++.

Лабораторная работа № 2. Разработка класса и перегрузка для объектов этого класса унарных и бинарных операторов методами класса и дружественными функциями. Создание статических и динамических объектов разработанного класса, создание статических и динамических массивов объектов разработанного класса.

Раздел 3 Работа с потоками данных (stream) в Си++

Тема 1 Потоки данных (stream)

Тема 2 Библиотека стандартных классов для работы с потоками данных в Си++

Лабораторная работа № 3. Реализация чтения и записи данных в файл\ из файла через библиотеку стандартных классов потокового ввода\вывода. Реализация ввода\вывода данных с клавиатуры\ на консоль через предопределенные объекты классов потокового ввода\вывода. Использование для ввода\вывода данных перегруженных операторов '>>', '<<'.

Раздел 4 Вмещающие классы. Шаблоны классов (родовые классы). Стандартная библиотека шаблонов С++ (STL)

Тема 1 Вмещающие классы

Тема 2 Шаблоны классов (родовые классы) в Си++

Тема 3 Стандартная библиотека шаблонов С++ (STL)

Лабораторная работа № 4. Разработка шаблонов вмещающих классов. Использование библиотеки STL (встроенных контейнеров и итераторов)

Раздел 5 Наследование. Раннее и позднее связывание. Полиморфизм.

Тема 1 Одиночное наследование

Тема 2 Множественное наследование.

Лабораторная работа № 5. Разработка иерархии классов (базовые и производные классы). Создание объектов производных классов. Работа с объектами производных классов.

Тема 3 Раннее и позднее связывание, полиморфизм.

Лабораторная работа № 6. Разработка иерархии классов с обычными и виртуальными методами. Работа с объектами классов, находящимися в одной иерархии наследования, в том числе через указатели.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

5.1. Текущий контроль успеваемости

Рейтинг-контроль №1 (контроль освоения разделов 1, 2)

Примерные вопросы и задания:

- 1) Что такое модуль?
- 2) В чем преимущества многомодульного проекта?
- 3) В чем заключается совместная и раздельная компиляция?
- 4) Как в Си++ реализовать совместную и раздельную компиляцию многомодульного проекта?
- 5) Какая область видимости и время жизни у переменных, объявленных с классами памяти `auto` \ `register` \ `static` \ `extern`?
- 6) В чем отличие функций, объявленных с классами памяти `static` \ `extern`?
- 7) Что такое класс? Привести пример объявления класса.
- 8) Что такое объект? Привести пример создания статического объекта. Привести пример работы с элементами класса через статический объект.
- 9) Привести пример создания и разрушения динамического объекта. Привести пример работы с элементами класса через динамический объект.
- 10) Привести пример создания и инициализации статического массива объектов.
- 11) Привести пример создания и инициализации динамического массива объектов.
- 12) Что такое конструктор?
- 13) Какие бывают конструкторы?
- 14) Что такое конструктор по умолчанию?
- 15) Что такое конструктор копирования?
- 16) Когда требуется явное определение в классе конструктора копирования?
- 17) Привести пример объявления и вызова конструктора с параметрами.
- 18) Привести пример объявления и вызова конструктора по умолчанию.
- 19) Привести пример объявления и вызова конструктора копирования.
- 20) Что такое деструктор?
- 21) Что такое `this` и для чего он нужен?
- 22) Привести пример явного и неявного использования `this`.
- 23) Что такое инкапсуляция?
- 24) Что такое модификаторы видимости?
- 25) Какие бывают модификаторы видимости?
- 26) Какой модификатор видимости будет установлен по умолчанию для элемента класса \ конструктора или деструктора, если класс объявлен с помощью ключевого слова `class`?
- 27) Какой модификатор видимости будет установлен по умолчанию для элемента класса \ конструктора или деструктора, если класс объявлен с помощью ключевого слова `struct`?
- 28) Что такое дружественные функции?
- 29) Чем дружественные функции отличаются от методов класса?

- 30) Чем дружественные функции отличаются от обычных функций?
- 31) Может ли функция быть дружественной к нескольким классам?
- 32) Привести пример дружественной функции.
- 33) Что такое дружественные классы?
- 34) Привести пример дружественного класса.
- 35) Привести пример перегрузки бинарного оператора для объектов класса методом класса.
- 36) Привести пример перегрузки бинарного оператора для объектов класса дружественной функцией
- 37) Привести пример перегрузки унарного оператора для объектов класса методом класса.
- 38) Привести пример перегрузки унарного оператора для объектов класса дружественной функцией
- 39) Можно ли в классе одновременно перегрузить один и тот же оператор методом класса и дружественной функцией? Если да, то привести пример перегрузки и вызова такой функции.
- 40) Можно ли передать параметр по значению\ по указателю\ по ссылке при перегрузке унарного оператора дружественной функцией? Пояснить на примере.
- 41) Какие операторы нельзя перегрузить?
- 42) Что такое статические элементы класса?
- 43) Привести пример работы со статическим полем.
- 44) Привести пример работы со статическим методом.

Рейтинг-контроль №2 (контроль освоения разделов 3, 4)

Примерные вопросы и задания:

- 1) Что такое поток данных?
- 2) Какие основные классы для работы с потоками данных представлены в библиотеке стандартных классов потокового ввода\ вывода?
- 3) Что такое cin? Привести пример использования.
- 4) Что такое cout? Привести пример использования.
- 5) В чем отличие бинарных файлов от текстовых?
- 6) Привести пример открытия файла для записи.
- 7) Привести пример открытия файла для дозаписи в конец файла.
- 8) Привести пример открытия файла для чтения.
- 9) Что будет при попытке открытия для записи несуществующего файла?
- 10) Что будет при попытке открытия для чтения несуществующего файла?
- 11) Привести пример чтения всех данных из бинарного файла в массив.
- 12) Привести пример записи всех данных из массива в бинарный файл.
- 13) Привести пример чтения всех данных из текстового файла в массив.
- 14) Привести пример записи всех данных из массива в текстовый файл.
- 15) Привести пример перегрузки оператора << для записи в файл объекта пользовательского класса.
- 16) Привести пример перегрузки оператора >> для чтения из файла объекта пользовательского класса
- 17) Привести пример чтения n-ого объекта от начала бинарного файла.
- 18) Привести пример чтения n-ого объекта от конца бинарного файла.
- 19) Привести пример обновления (update) в бинарном файле n-ого объекта от начала файла.
- 20) Привести пример обновления (update) в бинарном файле n-ого объекта от конца файла.
- 21) Что такое вмещающий класс?
- 22) Как классифицируются вмещающие классы?
- 23) Что такое шаблон класса?
- 24) Особенности компиляции шаблонов классов?
- 25) Что такое STL?
- 26) Что такое контейнеры в STL?
- 27) Какие типы (категории) контейнеров поддерживаются в STL?
- 28) Что такое итераторы в STL?

- 29) Какие типы (категории) итераторов поддерживаются в STL?
- 30) Особенности реализации контейнера `std:: vector<T>`? Привести пример использования.
- 31) Особенности реализации контейнера `std:: list<T>`? Привести пример использования.
- 32) Особенности реализации контейнера `std:: stack <T>`? Привести пример использования.
- 33) Особенности реализации контейнера `std:: queue <T>`? Привести пример использования.
- 34) Особенности реализации контейнера `std:: deque <T>`? Привести пример использования.
- 35) Особенности реализации контейнера `std:: set <T>`? Привести пример использования.
- 36) Особенности реализации контейнера `std:: map <T>`? Привести пример использования.

Рейтинг-контроль №3 (контроль освоения раздела 5)

Примерные вопросы и задания:

- 1) Привести пример объявления производного класса.
- 2) Что такое список инициализаторов в производном классе? Привести пример списка инициализаторов в производном классе.
- 3) Порядок вызова конструкторов при создании объекта производного класса?
- 4) Порядок вызова деструкторов при разрушении объекта производного класса?
- 5) Как регулируется видимость унаследованных элементов базового класса в производном классе?
- 6) Что в программировании понимают под связыванием?
- 7) Что такое раннее связывание?
- 8) Что такое позднее связывание?
- 9) Проблемы раннего связывания? Пояснить на примере.
- 10) Как реализован механизм позднего связывания в Си++?
- 11) Что такое виртуальная функция?
- 12) Что такое ТВМ?
- 13) Что такое поле ТВМ, где и когда оно создается и инициализируется, для чего оно нужно?
- 14) Преимущества и недостатки раннего и позднего связывания?
- 15) Проблемы множественного наследования для классов с виртуальными методами?
- 16) Могут ли конструкторы быть виртуальными?
- 17) Могут ли деструкторы быть виртуальными?
- 18) Могут ли статические методы класса быть виртуальными?
- 19) Что такое абстрактный класс? Привести пример абстрактного класса.
- 20) Могут ли в абстрактном классе объявляться конструкторы?
- 21) Может ли в абстрактном классе объявляться деструктор?
- 22) Что такое чистая функция?
- 23) Какие модификаторы видимости могут быть у чистой функции?
- 24) Что такое чисто абстрактный класс?
- 25) Для чего нужны чисто абстрактные классы (какую роль чисто абстрактные классы играют при проектировании иерархического семейства классов)?
- 26) Что такое полиморфизм? Привести пример полиморфного кода.
- 27) Что такое указатели на компоненты класса? Привести пример использования.

5.2. Промежуточная аттестация по итогам освоения дисциплины проводится в форме экзамена.

Примерные вопросы к экзамену:

1. Классы памяти
2. Многомодульное программирование: отдельная и совместная компиляция
3. Классы и объекты в Си++, свойство инкапсуляции
4. Конструкторы: назначение конструкторов, синтаксис объявления, виды конструкторов, варианты использования
5. Доступ к элементам класса (полям и методам) из методов и конструкторов класса и извне класса через объект

6. Создание статических и динамических объектов, массивов объектов, размещение последовательности объектов в динамической памяти

7. Дружественные функции

8. Перегрузка операторных функций

9. Статические элементы класса

10. Стандартные классы потокового ввода/вывода в C++

11. Вмещающие классы: назначение, классификация вмещающих классов

12. Перегрузка операторов new\ delete для реализации вмещающего класса

13. Шаблоны классов

14. Библиотека `STL: типы контейнеров, типы итераторов. примеры работы с контейнерами STL.

15. Наследование, создание объектов-потомков, порядок вызова конструкторов базовых классов

16. Модификаторы доступа

17. Множественное наследование: виды наследования, проблемы и пути решения

18. Раннее и позднее связывание, реализация механизма позднего связывания в C++.

19. Полиморфизм

20. Абстрактные классы

21. Указатели на компоненты класса.

5.3. Самостоятельная работа обучающегося.

На самостоятельную работу выносятся:

1) подготовка к лабораторным работам 1-6

2) освоение раздела №6 - разработка клиентских приложений с оконным пользовательским интерфейсом

Задание: изучить графическую библиотеку для разработки оконных приложений (любую на выбор) и по заданной предметной области разработать пользовательское приложение.

Требования к выполнению задания

1. Разработка объектной модели (ОМ)

2. Реализация основной бизнес-логики приложения (представление данных ОМ, изменение состояния ОМ, поиск по заданному критерию и проч.)

3. Организация постоянного хранения данных ОМ в файловой системе

4. Организация размещения данных ОМ во временном хранилище на сеанс работы приложения

5. Реализация оконного пользовательского интерфейса с помощью графической библиотеки

Задача должна быть реализована с соблюдением основных принципов ООП (инкапсуляция, наследование, полиморфизм). Результатом работы должно стать полноценное законченное клиентское приложение с оконным пользовательским интерфейсом. Задачи могут выполняться как индивидуально, так и в небольших группах по 2-3 человека.

Список задач

1. Автосалон: продажа подержанных и новых автомобилей

(2-3 человека)

2. Автосалон: аренда подержанных и новых автомобилей

(2-3 человека)

3. Автосервис - ремонт и техобслуживание автомобилей: перечень услуг, учет клиентов, организация системы скидок для постоянных клиентов

(2-3 человека)

4. Сервисный центр по ремонту компьютеров и оргтехники: классификация услуг, учет клиентов, организация скидок для постоянных клиентов

(2-3 человека)

5. Агентство недвижимости - покупка и продажа: классификация видов недвижимости, перечень услуг, учет клиентов, мониторинг недобросовестных покупателей и продавцов

- (2-3 человека)
6. Агентство недвижимости - аренда: классификация видов недвижимости, перечень услуг, учет клиентов, мониторинг недобросовестных арендаторов и владельцев недвижимости (2-3 человека)
7. Косметический салон: перечень услуг, учет клиентов, организация системы скидок для постоянных клиентов (2-3 человека)
8. Продажа мебели – линейка товаров: классификация по виду товара (мягкая мебель, гостиная, кухни, спальни, прихожие и проч.), стране- и фирме- производителю, материалу (дерево, ДСП, МДФ и проч), цене (1-2 человека)
9. Продажа саженцев и семян декоративных растений – каталог товаров: классификация по виду (дерево, кустарник, цветы, луговая трава и проч.), способу посадки (саженец, рассада, семена), времени посадки, периоду цветения. (1-2 человека)
10. Продажа мобильных телефонов - модельный ряд: классификация по техническим характеристикам, фирме-производителю и цене (2-3 человека)
11. Библиотека: учет книг, классификация по разделам и авторам (1-2 человека)
12. Картотека студентов: личные данные, успеваемость по семестрам (текущая и итоговая) (1-2 человека)
13. Картотека пациентов больницы (частной клиники): личные данные, даты осмотров, карта анализов, заболевания, лечение, процедуры, лечащие врачи (2-3 человека)
14. Кулинария: организация банка готовых блюд и формирование меню на основе имеющихся продуктов (2-3 человека)
15. Кафе: организация банка данных блюд и их классификация по принадлежности к группе (постные, мясные, рыбные, куриные, для диабетиков, для детей до 7 лет, для кремлевской диеты, по количеству калорий и т.д.), автоматический подбор блюд по заданному запросу с ограничениями (посетитель не ест мясо или посетитель поститься и т.д.) (2-3 человека)
16. Клуб знакомств: организация банка данных анкет клиентов, группировка анкет по персональным данным (образование, семейное положение, возраст и т.д) и по критериям поиска партнера (образование, семейное положение, возраст и т.д), автоматический подбор партнеров по заданным критериям (2-3 человека)
17. Томагочи (1-2 человека)
18. Ветеринарная клиника: регистрация животных, мониторинг перенесенных заболеваний и прививок (2-3 человека)
19. Клуб "домашних животных": регистрация животных, подбор пары, регистрация и учет потомства (2-3 человека)

Для выполнения заданий в рамках самостоятельной работы в семестре следует использовать лекционный материал и источники, указанные в разделе 6 рабочей программы. **Контрольными мероприятиями для оценки выполнения студентом самостоятельной работы являются отчеты по выполненным заданиями**

Фонд оценочных материалов (ФОМ) для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ Наличие в электронном каталоге ЭБС
Основная литература		
1. Васильев А.Н. Объектно-ориентированное программирование на C++ [Электронный ресурс]/ Васильев А.Н.— Электрон. текстовые данные.— Санкт-Петербург: Наука и Техника, 2016.— 544 с.	2016	http://www.iprbookshop.ru/60648.html
2. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ Новиков П.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 с.	2017	http://www.iprbookshop.ru/64650.html
3. Букунов С.В. Основы объектно-ориентированного программирования [Электронный ресурс]: учебное пособие/ Букунов С.В., Букунова О.В.— Электрон. текстовые данные.— Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017.— 196 с.	2017	http://www.iprbookshop.ru/74339.html
Дополнительная литература		
1. Герасимов В.П. Объектно-ориентированное программирование в научных исследованиях [Электронный ресурс]: учебное пособие (практикум)/ Герасимов В.П., Ковалев В.Д.— Электрон. текстовые данные.— Ставрополь: Северо-Кавказский федеральный университет, 2018.— 119 с	2018	http://www.iprbookshop.ru/92568.html
1. Объектно-ориентированное программирование [Электронный ресурс]: лабораторный практикум/ — Электрон. текстовые данные.— Ставрополь: Северо-Кавказский федеральный университет, 2018.— 111 с	2018	http://www.iprbookshop.ru/92712.html
2. Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс]: конспект лекций/ Лисицин Д.В.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2010.— 88 с	2010	http://www.iprbookshop.ru/44970.html

6.2. Периодические издания

1. Журнал РАН «Программирование» // Режим доступа: <https://www.ispras.ru/programming/>
2. Журнал «Прикладная информатика» // Режим доступа: <http://appliedinformatics.ru/>
3. Журнал «Информационные технологии» // Режим доступа: <http://novtex.ru/IT/index.htm>

6.3. Интернет-ресурсы

1. Programming Languages — C++, International Standard // Режим доступа: <https://isocpp.org/files/papers/N3690.pdf>
2. MSDN. Возвращение к C++ (современный C++). // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/hh279654.aspx>
3. MSDN. Справочник по языку C++. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/3bstk3k5.aspx>
4. MSDN. Visual Studio. // Режим доступа: <https://www.visualstudio.com/>

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекционные аудитории, оснащённые доской (для мела или маркера), экраном для проекционных систем, проектором и ноутбуком (420-3, 430-3).

Аудитории для проведения лабораторных занятий, оснащённые современными персональными компьютерами, объединёнными в локальную вычислительную сеть и укомплектованными необходимым системным и прикладным программным обеспечением (1226-3, 100-3, 511-3), аудитории вычислительного центра.

Компьютеры должны иметь выход в Интернет для получения доступа к справочной информации.

Среда разработки - Microsoft Visual Studio.

Рабочую программу составил ст. преп. каф. ФиПИМ Воронова Н.М.

(ФИО, должность, подпись)

Рецензент

(представитель работодателя) Генеральный директор ООО «ФС Сервис» Д.С. Квасов

(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПИМ

Протокол №1 от 30.08.2021 года

Заведующий кафедрой

(ФИО, подпись)

С.М. Аракелян

Рабочая программа рассмотрена и одобрена

на заседании учебно-методической комиссии направления 01.03.02.

Протокол №1 от 30.08.2021 года

Председатель комиссии

(ФИО, должность, подпись)

С.М. Аракелян

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на 20 22 / 20 23 учебный года

Протокол заседания кафедры № 1 от 30.08.2021 года

Заведующий кафедрой

Рабочая программа одобрена на 20 ____ / 20 ____ учебный года

Протокол заседания кафедры № ____ от ____ года

Заведующий кафедрой

Рабочая программа одобрена на 20 ____ / 20 ____ учебный года

Протокол заседания кафедры № ____ от ____ года

Заведующий кафедрой