

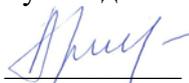


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

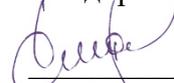
«СОГЛАСОВАНО»

Руководитель ОП

 Артемьева И.Л.

«УТВЕРЖДАЮ»

И.о. директора департамента

 Смагин С.В.

«2» марта 2023 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Объектно-ориентированное программирование
Направление подготовки **09.03.04 Программная инженерия**
(Программная инженерия)

Форма подготовки очная

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия, утвержденный приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 920.

Рабочая программа обсуждена на заседании департамента программной инженерии и искусственного интеллекта, протокол № 3.0 от «2» марта 2023 г.

И.о. директора департамента программной инженерии и искусственного интеллекта к.т.н. Смагин С.В.

Составители: старший преподаватель департамента ПИИИИ А.Е.Чусова

Владивосток
2023

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании кафедры/департамента::

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

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании кафедры/департамента::

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

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании кафедры/департамента::

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

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании кафедры/департамента::

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

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

Аннотация дисциплины

Объектно-ориентированное программирование

Общая трудоемкость дисциплины составляет 7 зачётных единиц /252 академических часа. Является дисциплиной части ОП, формируемой участниками образовательных отношений, изучается на 2 курсе и завершается экзаменом. Учебным планом предусмотрено проведение лекционных занятий в объеме 34 часа, лабораторных 102 часа, а также выделены часы на самостоятельную работу студента – 116 часов (в том числе на подготовку к экзамену 54 часа).

Язык реализации: русский.

Цель дисциплины - усвоение и закрепление методов создания объектно-ориентированных программ на C++, знакомство с основными понятиями в рамках данного подхода.

Задачи:

- Изучить основные концептуальные положения объектно-ориентированного программирования, а также механизмы, методы и средства разработки приложений в рамках данного направления

- Изучить язык программирования C++, научиться грамотно его использовать.

- Научиться использовать методы разработки объектно-ориентированных программ.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы.

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Индикаторы достижения компетенции	Наименование показателя оценивания (результата обучения)
Проектный	ПК-6. Способен использовать навыки моделирования, анализа и формальных методов конструирования программного обеспечения	ПК-6.1 Демонстрирует знание основ моделирования и формальных методов конструирования программного обеспечения	<u>Знает</u> основные модели информационных технологий, используемых при создании программ на языке высокого уровня, процесс создания программ, стандарты языков программирования <u>Умеет</u> использовать основные модели информационных технологий при создании программ на языке высокого уровня, общие принципы построения алгоритмов, основные алгоритмические конструкции <u>Владеет</u> основными моделями информационных технологий, используемых при создании программ на языке высокого уровня

		ПК-6.2 Использует формальные методы конструирования программного обеспечения	<u>Знает</u> методы использования информационных технологий при подготовке документации и текстовых редакторов при разработке программ и документации <u>Умеет</u> использовать информационные технологии при подготовке документации и разработке программ <u>Владеет</u> средствами, предоставляемыми информационными технологиями для подготовки документации и разработке программ
		ПК-6.3 Применяет методы формализации и моделирования программного обеспечения	<u>Знает</u> методы формализации и моделирования программного обеспечения <u>Умеет</u> разрабатывать программы, использующие данные в произвольном формате <u>Владеет</u> навыками разработки формальных моделей программного обеспечения
Проектный	ПК-8. Способен создавать программные интерфейсы	ПК-8.1 Демонстрирует знание способов создания программных интерфейсов	<u>Знает</u> способы создания программных интерфейсов <u>Умеет</u> использовать структуры данных и операторы языка программирования для разработки программных интерфейсов <u>Владеет</u> набором операций над структурами данных и операторами языка программирования и методами их использования
		ПК-8.2 Использует методы создания интуитивно понятных программных интерфейсов	<u>Знает</u> определение интуитивно понятного интерфейса <u>Умеет</u> создавать интуитивно понятные интерфейсы <u>Владеет</u> набором средств для создания интуитивно понятного интерфейса
		ПК-8.3 Создает современные программные интерфейсы	<u>Знает</u> свойства современного программного интерфейса <u>Умеет</u> создавать современные программные интерфейсы <u>Владеет</u> методами обеспечения качества создаваемых интерфейсов
Производственно-технологический	ПК-10. Способен использовать различные технологии разработки программного обеспечения	ПК-10.1 Демонстрирует знание современных технологий разработки программного обеспечения (структурное, объектно-ориентированное)	<u>Знает</u> ключевые понятия структурной и объектно-ориентированной парадигм программирования <u>Умеет</u> производить анализ предметной области в рамках современных технологий разработки ПО (структурной, объектно-ориентированной) <u>Владеет</u> навыками использования современных технологий разработки ПО (структурной, объектно-ориентированной)

		<p>ПК-10.2 Использует структурное и объектно-ориентированное проектирование при разработке программного обеспечения</p>	<p><u>Знает</u> нотацию моделей структурного (DFD) и объектно-ориентированного (диаграмма классов, диаграмма перехода состояний объекта, трасса событий) проектирования <u>Умеет</u> производить функциональную декомпозицию системы, переходить от контекстной диаграммы к спецификациям процессов (при структурном проектировании); производить объектно-ориентированный анализ (переходить от объектов к классам объектов, определять отношения между классами, поведение объектов, состояния объектов и переходы от одного состояния к другому) <u>Владеет</u> навыками построения структурных и объектно-ориентированных моделей проектирования при разработке ПО</p>
		<p>ПК-10.3 Применяет современные технологии разработки программного обеспечения</p>	<p><u>Знает</u> современные технологии разработки ПО, алгоритмы решения прикладных задач на основе типовых структур алгоритмов, методы организации сложных структур данных <u>Умеет</u> применять современные технологии разработки ПО <u>Владеет</u> синтаксисом и семантикой основных конструкций языка программирования высокого уровня, основанного на структурной и объектно-ориентированной технологии</p>
Производственно-технологический	ПК-11. Способен использовать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества	<p>ПК-11.1 Демонстрирует знание концепций и атрибутов качества программного обеспечения</p>	<p><u>Знает</u> концепции и атрибуты качества ПО <u>Умеет</u> использовать концепции и атрибуты качества ПО <u>Владеет</u> методами использования атрибутов качества ПО при разработке программного обеспечения</p>
		<p>ПК-11.2 Определяет необходимые средства для обеспечения качества разрабатываемого программного обеспечения</p>	<p><u>Знает</u> необходимые средства для обеспечения качества разрабатываемого ПО <u>Умеет</u> определять необходимые средства для обеспечения качества разрабатываемого ПО, записанного средствами языков программирования <u>Владеет</u> методами тестирования для определения качества программного обеспечения</p>

		ПК-11.3 Применяет методы, инструменты и технологии обеспечения качества программного обеспечения	<u>Знает</u> методы, инструменты и технологии обеспечения качества ПО <u>Умеет</u> применять методы, инструменты и технологии обеспечения качества ПО <u>Владеет</u> методами, инструментами и технологией обеспечения качества ПО, навыками тестирования разработанных программ для проверки достижения качества
Производственно-технологический	ПК-12. Способен применять стандарты и модели жизненного цикла	ПК-12.1 Демонстрирует знание стандартов и моделей жизненного цикла программного обеспечения	<u>Знает</u> содержание основных стандартов и модели жизненного цикла ПО <u>Умеет</u> работать с документацией, касающейся основных стандартов и моделей жизненного цикла ПО <u>Владеет</u> навыками работы со стандартами и различными моделями жизненного цикла ПО при создании ПО
		ПК-12.2 Использует модели жизненного цикла при создании программного обеспечения	<u>Знает</u> терминологию, нотацию различных групп моделей жизненного цикла при создании ПО <u>Умеет</u> планировать деятельность по созданию ПО в рамках выбранной модели жизненного цикла <u>Владеет</u> навыками создания ПО в рамках выбранной модели жизненного цикла для определенной предметной области
		ПК-12.3. Применяет стандарты и модели жизненного цикла программного обеспечения	<u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологические подходы, принципы структурного и объектно-ориентированного программирования, основные понятия COM-технологий и SOA-технологий; тенденции развития современных технологий программирования <u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта, стабильности требований, сроков получения стабильных версий программного обеспечения, выбрать технологический подход к разработке и соответствующие средства автоматизации разработки <u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу и конкретизации задач по созданию соответствующих технологических артефактов

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины - усвоение и закрепление методов создания объектно-ориентированных программ на С++, знакомство с основными понятиями в рамках данного подхода.

Задачи:

- Изучить основные концептуальные положения объектно-ориентированного программирования, а также механизмы, методы и средства разработки приложений в рамках данного направления

- Изучить язык программирования С++, научиться грамотно его использовать.

- Научиться использовать методы разработки объектно-ориентированных программ.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы.

Профессиональные компетенции выпускников и индикаторы их достижения:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Индикаторы достижения компетенции	Наименование показателя оценивания (результата обучения)
Проектный	ПК-6. Способен использовать навыки моделирования, анализа и формальных методов конструирования программного обеспечения	ПК-6.1 Демонстрирует знание основ моделирования и формальных методов конструирования программного обеспечения	<i>Знает</i> основные модели информационных технологий, используемых при создании программ на языке высокого уровня, процесс создания программ, стандарты языков программирования <i>Умеет</i> использовать основные модели информационных технологий при создании программ на языке высокого уровня, общие принципы построения алгоритмов, основные алгоритмические конструкции <i>Владеет</i> основными моделями информационных технологий, используемых при создании программ на языке высокого уровня

		ПК-6.2 Использует формальные методы конструирования программного обеспечения	<u>Знает</u> методы использования информационных технологий при подготовке документации и текстовых редакторов при разработке программ и документации <u>Умеет</u> использовать информационные технологии при подготовке документации и разработке программ <u>Владеет</u> средствами, предоставляемыми информационными технологиями для подготовки документации и разработке программ
		ПК-6.3 Применяет методы формализации и моделирования программного обеспечения	<u>Знает</u> методы формализации и моделирования программного обеспечения <u>Умеет</u> разрабатывать программы, использующие данные в произвольном формате <u>Владеет</u> навыками разработки формальных моделей программного обеспечения
Проектный	ПК-8. Способен создавать программные интерфейсы	ПК-8.1 Демонстрирует знание способов создания программных интерфейсов	<u>Знает</u> способы создания программных интерфейсов <u>Умеет</u> использовать структуры данных и операторы языка программирования для разработки программных интерфейсов <u>Владеет</u> набором операций над структурами данных и операторами языка программирования и методами их использования
		ПК-8.2 Использует методы создания интуитивно понятных программных интерфейсов	<u>Знает</u> определение интуитивно понятного интерфейса <u>Умеет</u> создавать интуитивно понятные интерфейсы <u>Владеет</u> набором средств для создания интуитивно понятного интерфейса
		ПК-8.3 Создает современные программные интерфейсы	<u>Знает</u> свойства современного программного интерфейса <u>Умеет</u> создавать современные программные интерфейсы <u>Владеет</u> методами обеспечения качества создаваемых интерфейсов
Производственно-технологический	ПК-10. Способен использовать различные технологии разработки программного обеспечения	ПК-10.1 Демонстрирует знание современных технологий разработки программного обеспечения (структурное, объектно-ориентированное)	<u>Знает</u> ключевые понятия структурной и объектно-ориентированной парадигм программирования <u>Умеет</u> производить анализ предметной области в рамках современных технологий разработки ПО (структурной, объектно-ориентированной) <u>Владеет</u> навыками использования современных технологий разработки ПО (структурной, объектно-ориентированной)

		<p>ПК-10.2 Использует структурное и объектно-ориентированное проектирование при разработке программного обеспечения</p>	<p><u>Знает</u> нотацию моделей структурного (DFD) и объектно-ориентированного (диаграмма классов, диаграмма перехода состояний объекта, трасса событий) проектирования <u>Умеет</u> производить функциональную декомпозицию системы, переходить от контекстной диаграммы к спецификациям процессов (при структурном проектировании); производить объектно-ориентированный анализ (переходить от объектов к классам объектов, определять отношения между классами, поведение объектов, состояния объектов и переходы от одного состояния к другому) <u>Владеет</u> навыками построения структурных и объектно-ориентированных моделей проектирования при разработке ПО</p>
		<p>ПК-10.3 Применяет современные технологии разработки программного обеспечения</p>	<p><u>Знает</u> современные технологии разработки ПО, алгоритмы решения прикладных задач на основе типовых структур алгоритмов, методы организации сложных структур данных <u>Умеет</u> применять современные технологии разработки ПО <u>Владеет</u> синтаксисом и семантикой основных конструкций языка программирования высокого уровня, основанного на структурной и объектно-ориентированной технологии</p>
<p>Производственно-технологический</p>	<p>ПК-11. Способен использовать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества</p>	<p>ПК-11.1 Демонстрирует знание концепций и атрибутов качества программного обеспечения</p>	<p><u>Знает</u> концепции и атрибуты качества ПО <u>Умеет</u> использовать концепции и атрибуты качества ПО <u>Владеет</u> методами использования атрибутов качества ПО при разработке программного обеспечения</p>
		<p>ПК-11.2 Определяет необходимые средства для обеспечения качества разрабатываемого программного обеспечения</p>	<p><u>Знает</u> необходимые средства для обеспечения качества разрабатываемого ПО <u>Умеет</u> определять необходимые средства для обеспечения качества разрабатываемого ПО, записанного средствами языков программирования <u>Владеет</u> методами тестирования для определения качества программного обеспечения</p>

		ПК-11.3 Применяет методы, инструменты и технологии обеспечения качества программного обеспечения	<u>Знает</u> методы, инструменты и технологии обеспечения качества ПО <u>Умеет</u> применять методы, инструменты и технологии обеспечения качества ПО <u>Владеет</u> методами, инструментами и технологией обеспечения качества ПО, навыками тестирования разработанных программ для проверки достижения качества
Производственно-технологический	ПК-12. Способен применять стандарты и модели жизненного цикла	ПК-12.1 Демонстрирует знание стандартов и моделей жизненного цикла программного обеспечения	<u>Знает</u> содержание основных стандартов и модели жизненного цикла ПО <u>Умеет</u> работать с документацией, касающейся основных стандартов и моделей жизненного цикла ПО <u>Владеет</u> навыками работы со стандартами и различными моделями жизненного цикла ПО при создании ПО
		ПК-12.2 Использует модели жизненного цикла при создании программного обеспечения	<u>Знает</u> терминологию, нотацию различных групп моделей жизненного цикла при создании ПО <u>Умеет</u> планировать деятельность по созданию ПО в рамках выбранной модели жизненного цикла <u>Владеет</u> навыками создания ПО в рамках выбранной модели жизненного цикла для определенной предметной области
		ПК-12.3. Применяет стандарты и модели жизненного цикла программного обеспечения	<u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологические подходы, принципы структурного и объектно-ориентированного программирования, основные понятия COM-технологий и SOA-технологий; тенденции развития современных технологий программирования <u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта, стабильности требований, сроков получения стабильных версий программного обеспечения, выбрать технологический подход к разработке и соответствующие средства автоматизации разработки <u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу и конкретизации задач по созданию соответствующих технологических артефактов

Для формирования вышеуказанных компетенций в рамках дисциплины «Объектно-ориентированное программирование» применяются следующие методы активного/ интерактивного обучения: метод проектов.

Трудоёмкость дисциплины и видов учебных занятий по дисциплине

Общая трудоёмкость дисциплины составляет 10 зачётных единиц 360 академических часов.

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине являются:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Лаб	Лабораторные работы
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
1	Раздел I. Основные принципы объектно-ориентированного программирования	3	-	50	-	144	22	-	УО-1; УО-3; ПР-6;
2	Раздел 2 Основы программирования на языке C++	3,4	-	72	-		45	27	
	Итого:		-	122	-	144	67	27	

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекционные занятия по дисциплине не предусмотрены учебным планом. Студенты изучают материал самостоятельно.

Раздел I. Основные принципы объектно-ориентированного программирования

Тема 1. Основные понятия объектно-ориентированного подхода в программировании.

Основные принципы ООП: наследование инкапсуляция, полиморфизм. Понятие класс, экземпляр класса, метод класса, интерфейс. История развития концепции ООП, примеры объектно-ориентированных языков.

Тема 2. Основные элементы языка C++

Типы данных, их размерность, реализация управляющих конструкций в C++, организация ввода-вывода данных, манипуляторы потока вывода. Основные библиотеки для работы со стандартными типами данных.

Тема 3. Функции в C++ и реализация простых структур

Синтаксис описания функций, рекурсивные функции, перегрузка. Функции с переменным числом параметров, прототипы функций. Передача параметров функции по значению, по ссылке по указателю. Массив, индексация, организация работы с массивом.

Тема 4. Модульное программирование

Многофайловые программы – принцип организации и доступа к отдельным файлам, файлы исходного кода и заголовочные файлы. Область видимости переменных, описание глобальных переменных.

Тема 5. Работа с памятью – динамическое и статическое выделение памяти. Указатели.

Способы выделения динамической памяти, исключение утечек памяти. Указатели – синтаксис и назначение, арифметика указателей. Работа с массивами в динамической памяти, организация линейных списков, двухсвязных списков и бинарных деревьев.

Тема 6. Пользовательские типы данных

Структура, объединение, перечислимый тип данных – синтаксис, назначение, основные принципы разработки указанных типов.

Тема 7. Классы, наследование, специальные методы классов

Синтаксис описания класса, реализация методов, тип доступа, специфика наследования с различным типом доступа. Одиночное и множественное наследование. Статические и константные методы класса. Конструктор и деструктор.

Тема 8. Перегрузка операторов. Виртуальные функции.

Дружественные функции. Дружественные классы

Специфика перегрузки операторов, наследование перегруженных операций. Абстрактный класс, виртуальная функция и чистая виртуальная функция. Переопределение виртуальных методов.

Тема 9. Шаблоны функций и классов. Обработка исключительных ситуаций.

Синтаксис, назначение и принцип разработки шаблонов функций и классов. Конструкция try...catch с параметрами и без.

Тема 10. Библиотека STL, основные элементы. Итераторы и аллокаторы. Контейнерные классы.

Элементы библиотеки STL с прямым и ассоциативным доступом.

Тема 11. Организация многопоточных программ. Мьютексы.

Параллельная обработка данных с использованием потоков, мьютексы – синтаксис, назначение, принцип работы.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Практические занятия (0 час.)

Практические занятия учебным планом не предусмотрены.

Лабораторные работы (122 час.)

3 семестр (58 час.)

Лабораторная работа 1. Объекты и классы C++. (28 часов)

Получение базовых знаний в области объектно-ориентированного программирования, изучение основных понятий и парадигм: определение класса для заданной задачи, программирование методов класса, типы конструкторов, прототипы функций.

Лабораторная работа 2. Работа с классами в C++. (20 часов).

Получение практических навыков в области перегрузки операторов.

Разработка классов для работы с данными разных типов: стек, список, вектор, комплексные числа и т.д. Перегрузка операций. Дружественные функции.

Лабораторная работа 3. Классы и динамическое выделение памяти. (10 часов)

Получение практических навыков в выделении динамической памяти под объекты классов.

4 семестр (64 часа)

Лабораторная работа 4. Наследование. (25 часов)

Получение практических навыков в области одиночного и множественного наследования классов. Базовый класс. Виртуальные методы.

Лабораторная работа 5. Шаблоны функций и классов. Обработка исключительных ситуаций. (25 часов)

Получение практических навыков работы в области разработки дружественных классов

Лабораторная работа 6. Организация многопоточных программ. Мьютексы. (14 часов)

Получение практических навыков работы с потоками ввода и вывода C++.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Разработка объектно-ориентированных приложений» представлено в разделе VIII и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
3 семестр				
1	1-2 неделя семестра	Изучить основные понятия объектно-ориентированного подхода в программировании	2	Защита индивидуального задания
2	3-6 неделя семестра	Освоить основные элементы языка C++	2	Защита индивидуального задания
3	7-8 неделя семестра	Изучить функции в C++ и реализация простых структур	2	Защита индивидуального задания
4	9-10 неделя семестра	Изучить модульное программирование	2	Защита индивидуального задания
5	11-13 неделя семестра	Изучить работу с памятью – динамическое и статическое выделение памяти. Указатели.	2	Защита индивидуального задания

6	14-18 неделя семестра	Изучить пользовательские типы данных	2	Защита индивидуального задания
	1-18 недели	Онлайн курс	10	
4 семестр				
7	1-3 неделя семестра	Изучить классы, наследование, специальные методы классов	10	Защита индивидуального задания
8	4-7 неделя семестра	Изучить что такое перегрузка операторов. Виртуальные функции. Дружественные функции. Дружественные классы	10	Защита индивидуального задания
9	8-12 неделя семестра	Изучить шаблоны функций и классов. Обработка исключительных ситуаций	10	Защита индивидуального задания
10	13-14 неделя семестра	Изучить библиотеку STL, основные элементы. Итераторы и аллокаторы. Контейнерные классы	10	Защита индивидуального задания
11	15-16 неделя семестра	Изучить организация многопоточных программ. Мьютексы	10	Защита индивидуального задания
	17-18 недели	Подготовка к экзамену	22	
Итого:			94	

Рекомендации по самостоятельной работе студентов

Планирование и организация времени, отведенного на выполнение заданий самостоятельной работы.

Изучив график выполнения самостоятельных работ, следует правильно её организовать. Рекомендуется изучить структуру каждого задания, обратить внимание на график выполнения работ, отчетность по каждому заданию предоставляется в последнюю неделю согласно графику. Обратить внимание, что итоги самостоятельной работы влияют на окончательную оценку по итогам освоения учебной дисциплины.

Работа с литературой.

При выполнении ряда заданий требуется работать с литературой. Рекомендуется использовать различные возможности работы с литературой: фонды научной библиотеки ДВФУ (<http://www.dvfu.ru/library/>) и других ведущих вузов страны, а также доступных для использования научно-библиотечных систем.

В процессе выполнения самостоятельной работы, в том числе при написании эссе рекомендуется работать со следующими видами

изданий:

а) Научные издания, предназначенные для научной работы и содержащие теоретические, экспериментальные сведения об исследованиях. Они могут публиковаться в форме: монографий, научных статей в журналах или в научных сборниках;

б) Учебная литература подразделяется на:

- учебные издания (учебники, учебные пособия, тексты лекций), в которых содержится наиболее полное системное изложение дисциплины или какого-то ее раздела;

- справочники, словари и энциклопедии – издания, содержащие краткие сведения научного или прикладного характера, не предназначенные для сплошного чтения. Их цель – возможность быстрого получения самых общих представлений о предмете.

Существуют два метода работы над источниками:

– сплошное чтение обязательно при изучении учебника, глав монографии или статьи, то есть того, что имеет учебное значение. Как правило, здесь требуется повторное чтение, для того чтобы понять написанное. Старайтесь при сплошном чтении не пропускать комментарии, сноски, справочные материалы, так как они предназначены для пояснений и помощи. Анализируйте рисунки (карты, диаграммы, графики), старайтесь понять, какие тенденции и закономерности они отражают;

– метод выборочного чтения дополняет сплошное чтение; он применяется для поисков дополнительных, уточняющих необходимых сведений в словарях, энциклопедиях, иных справочных изданиях. Этот метод крайне важен для повторения изученного и его закрепления, особенно при подготовке к зачету.

Для того чтобы каждый метод принес наибольший эффект, необходимо фиксировать все важные моменты, связанные с интересующей Вас темой.

Конспект – это способ самостоятельно изложить содержание книги или статьи в логической последовательности. Конспектируя какой-либо источник, надо стремиться к тому, чтобы немногими словами сказать о многом. В тексте конспекта желательно поместить не только выводы или положения, но и их аргументированные доказательства (факты, цифры, цитаты).

Писать конспект можно и по мере изучения произведения, например, если прорабатывается монография или несколько журнальных статей.

Методические рекомендации по выполнению заданий для самостоятельной работы и критерии оценки.

Собеседование (устный опрос) позволяет оценить знания и кругозор

студента, умение логически построить ответ, владение монологической речью и иные коммуникативные навыки.

Опрос – важнейшее средство развития мышления и речи. Обучающая функция опроса состоит в выявлении деталей, которые по каким-то причинам оказались недостаточно осмысленными в ходе учебных занятий и при подготовке задания по самостоятельной работе.

Критерии оценки. Используется зачетная система. Во время опроса допускается не более 1-й ошибки или неточности по названию периода, его времени и длительности.

Оценка	Требования
«зачтено»	Студент владеет навыками самостоятельной работы по теме исследования, реферировать литературные источники; методами анализа теоретических и/или практических аспектов изучаемой области. Эссе характеризуется смысловой цельностью, связностью и последовательностью изложения. Студент умеет обобщать фактический материал, делать самостоятельные выводы. Работа соответствует требованиям и выполнена в установленные сроки.
«не зачтено»	Не раскрыта структура и теоретическая составляющая темы. Студент не умеет обобщать фактический материал, делать самостоятельные выводы, не владеет навыком реферировать литературные источники. Эссе не выполнено.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/ разделы / темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства – наименование	
				текущий контроль	промежуточная аттестация
1	Объекты и классы C++. Работа с классами в C++. Классы и динамическое выделение памяти. Наследование. Шаблоны функций и	ПК-6.1 Демонстрирует знание основ моделирования и формальных методов конструирования программного обеспечения	<u>Знает</u> основные модели информационных технологий, используемых при создании программ на языке высокого уровня, процесс создания программ, стандарты языков программирования <u>Умеет</u> использовать основные модели информационных технологий при создании программ на языке высокого уровня, общие принципы построения алгоритмов, основные алгоритмические конструкции <u>Владеет</u> основными моделями информационных технологий, используемых при создании программ на языке высокого уровня	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 1	Зачет, вопросы 1-9

классов. Обработка исключительн ых ситуаций. Организация многопоточных программ. Мьютексы.	ПК-6.2 Использует формальные методы конструирования программного обеспечения	<i>Знает</i> методы использования информационных технологий при подготовке документации и текстовых редакторов при разработке программ и документации	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 1	
		<i>Умеет</i> использовать информационные технологии при подготовке документации и разработке программ		
		<i>Владеет</i> средствами, предоставляемыми информационными технологиями для подготовки документации и разработке программ		
	ПК-6.3 Применяет методы формализации и моделирования программного обеспечения	<i>Знает</i> методы формализации и моделирования программного обеспечения	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 1	
		<i>Умеет</i> разрабатывать программы, использующие данные в произвольном формате		
		<i>Владеет</i> навыками разработки формальных моделей программного обеспечения		
	ПК-8.1 Демонстрирует знание способов создания программных интерфейсов	<i>Знает</i> способы создания программных интерфейсов	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 2	
		<i>Умеет</i> использовать структуры данных и операторы языка программирования для разработки программных интерфейсов		
		<i>Владеет</i> набором операций над структурами данных и операторами языка программирования и методами их использования		
	ПК-8.2 Использует методы создания интуитивно понятных программных интерфейсов	<i>Знает</i> определение интуитивно понятного интерфейса	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 2	
		<i>Умеет</i> создавать интуитивно понятные интерфейсы		
		<i>Владеет</i> набором средств для создания интуитивно понятного интерфейса		
ПК-8.3 Создает современные программные интерфейсы	<i>Знает</i> свойства современного программного интерфейса	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор		
	<i>Умеет</i> создавать современные программные интерфейсы			
	<i>Владеет</i> методами обеспечения качества создаваемых интерфейсов			

Зачет,
вопросы 1-
4

				ная работа 2	
	ПК-10.1 Демонстрирует знание современных технологий разработки программного обеспечения (структурное, объектно- ориентированно е)	<u>Знает</u> ключевые понятия структурной и объектно- ориентированной парадигм программирования	<u>Умеет</u> производить анализ предметной области в рамках современных технологий разработки ПО (структурной, объектно-ориентированной)	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 3	
	ПК-10.2 Использует структурное и объектно- ориентированно е проектирование при разработке программного обеспечения	<u>Знает</u> нотацию моделей структурного (DFD) и объектно- ориентированного (диаграмма классов, диаграмма перехода состояний объекта, трасса событий) проектирования	<u>Умеет</u> производить функциональную декомпозицию системы, переходить от контекстной диаграммы к спецификациям процессов (при структурном проектировании); производить объектно- ориентированный анализ (переходить от объектов к классам объектов, определять отношения между классами, поведение объектов, состояния объектов и переходы от одного состояния к другому)	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 3	Зачет, вопросы 10-12
	ПК-10.3 Применяет современные технологии разработки программного обеспечения	<u>Знает</u> современные технологии разработки ПО, алгоритмы решения прикладных задач на основе типовых структур алгоритмов, методы организации сложных структур данных	<u>Умеет</u> применять современные технологии разработки ПО	УО1 собеседов ание ПР2 контроль ное задание ПР-6 лаборатор ная работа 3	
	ПК-11.1 Демонстрирует знание концепций и атрибутов	<u>Знает</u> концепции и атрибуты качества ПО	<u>Умеет</u> использовать концепции и атрибуты качества ПО	УО1 собеседов ание ПР2 контроль	Зачет, вопросы 5- 6

	качества программного обеспечения	<i>Владеет</i> методами использования атрибутов качества ПО при разработке программного обеспечения	ное задание ПР-6 лабораторная работа 4	
	ПК-11.2 Определяет необходимые средства для обеспечения качества разрабатываемого программного обеспечения	<i>Знает</i> необходимые средства для обеспечения качества разрабатываемого ПО	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 4	
<i>Умеет</i> определять необходимые средства для обеспечения качества разрабатываемого ПО, записанного средствами языков программирования				
<i>Владеет</i> методами тестирования для определения качества программного обеспечения				
	ПК-11.3 Применяет методы, инструменты и технологии обеспечения качества программного обеспечения	<i>Знает</i> методы, инструменты и технологии обеспечения качества ПО	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 4	
<i>Умеет</i> применять методы, инструменты и технологии обеспечения качества ПО				
<i>Владеет</i> методами, инструментами и технологией обеспечения качества ПО, навыками тестирования разработанных программ для проверки достижения качества				
	ПК-12.1 Демонстрирует знание стандартов и моделей жизненного цикла программного обеспечения	<i>Знает</i> содержание основных стандартов и модели жизненного цикла ПО	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 5	
<i>Умеет</i> работать с документацией, касающейся основных стандартов и моделей жизненного цикла ПО				
<i>Владеет</i> навыками работы со стандартами и различными моделями жизненного цикла ПО при создании ПО				
	ПК-12.2 Использует модели жизненного цикла при создании программного обеспечения	<i>Знает</i> терминологию, нотацию различных групп моделей жизненного цикла при создании ПО	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 5	Зачет, вопросы 11-12
<i>Умеет</i> планировать деятельность по созданию ПО в рамках выбранной модели жизненного цикла				
<i>Владеет</i> навыками создания ПО в рамках выбранной модели жизненного цикла для определенной предметной области				
	ПК-12.3. Применяет стандарты и модели жизненного цикла	<i>Знает</i> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологические подходы,	УО1 собеседование ПР2 контрольное задание	

		программного обеспечения	принципы структурного и объектно-ориентированного программирования, основные понятия СОМ-технологий и SOA-технологий; тенденции развития современных технологий программирования	задание ПР-6 лабораторная работа 5	
			<i>Умеет</i> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта, стабильности требований, сроков получения стабильных версий программного обеспечения, выбирать технологический подход к разработке и соответствующие средства автоматизации разработки		
			<i>Владеет</i> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу и конкретизации задач по созданию соответствующих технологических артефактов		

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в разделе IX.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.— 285 с.— Режим доступа: <http://www.iprbookshop.ru/79706.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.— Режим доступа: <http://www.iprbookshop.ru/64650.html>.
3. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С#: Учебное пособие / Хорев П.Б. - Москва : Форум, НИЦ ИНФРА-М, 2016. - 200 с. (Высшее образование: Бакалавриат) ISBN 978-5-00091-144-0. - Текст

- : электронный. - URL: <https://znanium.com/catalog/product/529350>
4. Павловская, Т. А. С/С++. Процедурное и объектно-ориентированное программирование : учебник для вузов / Татьяна Павловская. - Санкт-Петербург : Питер, 2015. - 495 с. - Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:780731&theme=FEFU>
 5. Язык программирования С++. Специальное издание: пер. с англ. / Бьерн Страуструп. - М.: Вильямс, 2011. — 1135с.
 6. С/С++. Программирование на языке высокого уровня: учебник для вузов для магистров и бакалавров / Т. А. Павловская. СПб.: Питер, 2011.-460 с.
 7. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 512 с. — URL: <https://znanium.com/catalog/product/1000008>
 8. Объектно-ориентированное программирование в С++: лекции и упражнения [Электронный ресурс]: Учебное пособие для вузов / Ашарина И.В. - М.: Горячая линия - Телеком, 2012. - <http://www.studentlibrary.ru/book/ISBN9785991270014.html>

Дополнительная литература

(печатные и электронные издания)

1. С/С++. Структурное и объектно-ориентированное программирование: практикум / Т.А. Павловская, Ю.А. Щупак.- СПб.: Питер, 2011.- 352 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:418970&theme=FEFU>
2. Павловская Т.А. С/С++. Программирование на языке высокого уровня-СПб.: Питер, 2021. - 461 с.
3. Царев, Р. Ю. Программирование на языке Си [Электронный ресурс]: учеб. пособие / Р. Ю. Царев. – Красноярск: Сиб. федер. ун-т, 2014. – 108 с. <http://znanium.com/catalog.php?bookinfo=510946>
4. Комлев, Н. Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей / Комлев Н.Ю. - Москва :СОЛОН-Пр., 2015. - 298 с.:ISBN 978-5-91359-138-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/884394>
5. Керниган Б., Ритчи Д. Язык программирования С. Пер. с англ. — М.: Издательский дом "Вильямс", 2012. — 289 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:666721&theme=FEFU>
6. Программирование на языке С++ в среде Qt Creator/ Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк, О. В. Чеснокова, А. С. Чмыхало - М. : ALT Linux, 2015. — 448 с.
7. Объектно-ориентированное программирование в С++ / Р. Лафоре ; [Пер. с англ. А. Кузнецов и др.]. - 4. изд. - М. [и др.] : Питер, 2018 (ГПП Печ. Двор). - 923 с

Профессиональные базы данных и информационные справочные системы

1. База данных Scopus <http://www.scopus.com/home.url>
2. База данных Web of Science <http://apps.webofknowledge.com/>
3. Электронная библиотека диссертаций Российской государственной библиотеки <http://diss.rsl.ru/>
4. Электронные базы данных EBSCO <http://search.ebscohost.com/>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <https://code-live.ru/tag/cpp-manual/> Портал о программировании
2. <http://cppstudio.com/cat/274/> Язык программирования С++
3. <http://www.c-cpp.ru/> Программирование на С и С++. Онлайн справочник программиста
4. <http://purecodecpp.com/> основы программирования на С++ для начинающих
5. <http://progbook.ru/c/> Книги по С++
6. <http://www.programmersclub.ru/main/> Клуб программистов
7. <http://info-comp.ru/> Информационный портал. Все о компьютере и программировании для начинающих

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Microsoft Office (Access, Excel, PowerPoint, Word и т. д).
2. Open Office.
3. Skype.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека "Консультант студента".
4. Электронно-библиотечная система IPRbooks.
5. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лабораторные занятия проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Объектно-ориентированное программирование» изучается в следующих организационных формах: лабораторное занятие; самостоятельное изучение теоретического материала; индивидуальные и групповые консультации.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы, отраженного в документации и защиты проекта.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

Лабораторные работы

В результате выполнения лабораторных работ студент должен научиться формализовать поставленную задачу, составлять и оформлять программы на языке программирования C++, тестировать и отлаживать программы, использовать для разработки программ современные интегрированные среды разработки программ.

При выполнении лабораторной работы необходимо следовать методическим рекомендациям по ее выполнению. Результатом лабораторной работы является программа и отчет (документация), которые демонстрируются преподавателю в конце работы. Студент должен уметь отвечать на вопросы преподавателя, поясняя процесс создания программ и документа и выполнения работы.

Индивидуальные задания необходимо выполнять согласно рассмотренной технологии:

1. изучить словесную постановку задачи;
2. сформулировать математическую постановку задачи;
3. выбрать метод решения задачи, если это необходимо;
4. разработать схему алгоритма и спроектировать требуемые классы;
5. записать разработанный алгоритм на языке C++;
6. разработать контрольные тесты программы;
7. отладить программу;
8. написать отчет.

Содержание отчета

1. Титульный лист.
2. Неформальная постановка задачи (НПЗ).
3. Формальная постановка задачи (ФПЗ).
4. Проекты классов
5. Текст программы с комментариями.
6. Тестовые условия.

Работа с литературой

В процессе подготовки к лабораторным работам студентам необходимо обратить особое внимание на поиск и на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выполнения индивидуального проекта, выявить широкий спектр мнений по изучаемой проблеме.

Самостоятельная работа студента

Основными формами самостоятельной работы студента являются:

- подготовка к лабораторным занятиям, зачету,
- изучение обязательной и дополнительной литературы,
- поиск информации по изучаемым темам в периодических изданиях и Интернете,
- изучение в рамках программы курса тем, не выносимых на лабораторные работы,
- оформление отчетов по лабораторным работам.

Контроль за выполнением работы студента производится в виде контроля каждого этапа работы.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
---	---	---

<p>690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, Корпус D, аудитория 733/733а Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации</p>	<p>Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CCBA – 1 шт. Доска аудиторная, Моноблок Lenovo C360G- i34164G500UDK с лицензионными программами Microsoft Office 2013(13 шт.) и аудиовизуальными средствами проектор Panasonic DLPPjectorPT- D2110XE</p>	<p>1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12, Alice 3, Anaconda3, Autodesk, CodeBlocks, CorelDRAW X7, Dia ,Directum4.8 ,DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git version 1.9.5-preview20141217, Greenfoot, gsview, Inscapе0.91, Java 8 Update 60 (64-bit), Java development Kit, Kaspersky, Lazarus, LibreOffice4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight, Microsoft Silverlight 5SDK-русский, Microsoft Sistem Center, Microsoft Visial Studio 2012, MikTeX2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Pythonv 3.4, Pyton2.7(3.4,3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity2017.3.1f1, Veusz, Vim8.1, Visual Paradigm CE, Visual Studio 2013, Microsoft Visual C++, Windows Kits, Windows Phone SDK8.1, Xilinx Design Tools, Acrobat Reader DC, Adobe Bridge CS3, Adobe Device Central, CS3,Adobe Extend Script Toolkit 2,Adobe Photoshope CS3,DVD-студия Windows, Google Chrome, Internet Explorer, ITMOproctor, Mozilla Firefox 39.0 (x86 ru), Visual Studio Installer, Windows Media Center, WinSCP, Xming 6.9.0.31, ESET Endpoint Security, Firebird 2.5.3.26780 (x64), 7-Zip 9.20 (x64 edition), Adobe Flash Player, Adobe Shockwave Player, ESET Endpoint Antivirus, FOG Service, Hao Zip</p>
<p>690922, Приморский край, г.Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, Корпус D, аудитория 733/733а. Аудитория для самостоятельной работы</p>	<p>Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CCBA – 1 шт. Доска аудиторная, Моноблок Lenovo C360G- i34164G500UDK с лицензионными программами Microsoft Office 2013(13 шт.) и аудиовизуальными средствами проектор Panasonic DLPPjectorPT- D2110XE</p>	<p>1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12, Alice 3, Anaconda3, Autodesk, CodeBlocks, CorelDRAW X7, Dia ,Directum4.8 ,DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git version 1.9.5-preview20141217, Greenfoot, gsview, Inscapе0.91, Java 8 Update 60 (64-bit), Java development Kit, Kaspersky, Lazarus, LibreOffice4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight, Microsoft Silverlight 5SDK-русский, Microsoft Sistem Center, Microsoft Visial Studio 2012, MikTeX2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Pythonv 3.4, Pyton2.7(3.4,3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity2017.3.1f1, Veusz, Vim8.1, Visual Paradigm CE, Visual Studio 2013, Microsoft Visual C++, Windows Kits, Windows Phone SDK8.1, Xilinx Design Tools, Acrobat Reader DC, Adobe Bridge CS3, Adobe Device Central, CS3,Adobe Extend Script Toolkit 2,Adobe Photoshope CS3,DVD-студия Windows, Google Chrome, Internet Explorer, ITMOproctor, Mozilla Firefox 39.0 (x86 ru), Visual Studio Installer, Windows Media Center, WinSCP, Xming 6.9.0.31, ESET Endpoint Security, Firebird 2.5.3.26780 (x64), 7-Zip 9.20 (x64 edition), Adobe Flash Player, Adobe Shockwave Player, ESET Endpoint Antivirus, FOG Service, Hao Zip</p>

Перечень программного обеспечения:

Лицензионное программное обеспечение:

AutoCAD;

Autodesk 3DS Max;

Microsoft Visio;

SPSS Statistics Premium Campus Edition;

MathCad Education University Edition;
Microsoft Office 365;
Office Professional Plus 2019;
Photoshop CC for teams All Apps AL;
SolidWorks Campus 500;
Windows Edu Per Device 10 Education;
KOMPAS 3D;
Microsoft Teams

Свободно распространяемое программное обеспечение:

Adobe Reader DC 2015.020 - пакет программ для просмотра электронных публикаций в формате PDF:
http://www.images.adobe.com/content/dam/acom/en/legal/licenses-terms/pdf/PlatformClients_PC_WWEULA-en_US-20150407_1357.pdf ;

ArgoUML - программный инструмент моделирования UML:
<http://argouml.tigris.org> ;

Dia - пакет программ для создания диаграмм в виде блок-схем алгоритмов программ, древовидных схем, статических структур UML, баз данных, диаграмм сущность-связь и др. диаграмм:
https://portableapps.com/support/portable_app#using ;

DiagramDesigner - пакет программ для создания потоковых диаграмм, диаграмм классов UML, иллюстраций и др. диаграмм:
<https://www.foosshub.com/Diagram-Designer.html#clickToStartDownload> ;

IrfanView - пакет программ для просмотра (воспроизведения) графических, видео- и аудиофайлов: <http://www.irfanview.com/eula.htm> ;

LibreOffice - офисный пакет:
<http://www.libreoffice.org/about-us/licenses/> ;

Maxima – система для работы с символьными и численными выражениями:
<http://maxima.sourceforge.net/maximalist.html> ;

Project Libre - аналог программной системы управления проектами Microsoft Project для стационарного компьютера:
<https://континентсвободы.рф:/офис/проекты/projectlibre-система-управления-проектами.html> ;

Python - система программирования - динамический интерактивный объектно-ориентированный язык программирования:
<https://python.ru.uptodown.com/windows/download> ;

Ramus Educational - пакет программ для разработки и моделирования бизнес-процессов в виде диаграмм IDEF0 и DFD:
<https://www.obnovisoft.ru/ramus-educational> ;

Scilab –система - язык программирования высокого уровня, рассчитанный на научные расчеты: <http://www.scilab.org/scilab/license> ;

WhiteStarUML –программный инструмент моделирования UML, полученный из StarUML, совместимый с Windows 7-10: <https://github.com/StevenTCramer/WhiteStarUml/blob/master/staruml/deploy/License.txt/> ;

WinDjView – программа для просмотра электронных публикаций в формате DJV и DjVu: <https://windjview.sourceforge.io/ru/> .

Для проведения учебных занятий по дисциплине, а также для организации самостоятельной работы студентам доступно следующее лабораторное оборудование и специализированные кабинеты, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении учебных и научно-производственных работ.

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной поддержки.

VIII. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Самостоятельная работа состоит в выполнении на компьютере индивидуальных практических заданий по изучаемым темам (94 час.).

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Виды СРС	Всего часов	Форма контроля
1.	1-2 неделя обучения	Изучение теоретического материала к СР-1 по литературным источникам	5	Собеседование
		Оформление отчета по СР-1	3	Проверка отчета
2.	3-5 неделя обучения	Изучение теоретического материала к СР-2 по литературным источникам	4	Собеседование
		Оформление отчета по СР-2	4	Проверка отчета

3.	6-8 неделя обучения	Изучение теоретического материала к СР-3 по литературным источникам	4	Собеседование
		Оформление отчета по СР-3	4	Проверка отчета
4.	9-11 неделя обучения	Изучение теоретического материала к СР-4 по литературным источникам	4	Собеседование
		Оформление отчета по СР-4	4	Проверка отчета
5.	12-14 неделя обучения	Изучение теоретического материала к СР-5 по литературным источникам	4	Собеседование
		Оформление отчета по СР-5	4	Проверка отчета
6.	15-18 неделя обучения	Изучение теоретического материала к СР-6 по литературным источникам	5	Собеседование
		Оформление отчета по СР-6	4	Проверка отчета
7.	19-21 неделя обучения	Изучение теоретического материала к СР-7 по литературным источникам	5	Собеседование
		Оформление отчета по СР-7	4	Проверка отчета
8.	22-24 неделя обучения	Изучение теоретического материала к СР-8 по литературным источникам	5	Собеседование
		Оформление отчета по СР-8	4	Проверка отчета
9.	25-28 неделя обучения	Изучение теоретического материала к СР-9 по литературным источникам	5	Собеседование
		Оформление отчета по СР-9	4	Проверка отчета
10.	29-32 неделя обучения	Изучение теоретического материала к СР-10 по литературным источникам	5	Собеседование
		Оформление отчета по СР-10	4	Проверка отчета
11.	33-36 неделя обучения	Изучение теоретического материала к СР-11 по литературным источникам	5	Собеседование
		Оформление отчета по СР-11	4	Проверка отчета
Итого:			94	

Рекомендации по самостоятельной работе студентов Самостоятельная работа (94 час.)

В процессе самостоятельной работы студенты выполняют задания на компьютере по программированию с использованием интерпретатора языка C++.

Примеры задач для самостоятельного решения

Требования: Перед каждой самостоятельной работой обучающемуся необходимо изучить Методические указания по выполнению лабораторных работ по дисциплине «Объектно-ориентированное программирование».

Самостоятельная работа №1.

Описать структуру с именем STUDENT, содержащую следующие поля: NAME—фамилия и инициалы; GROUP—номер группы; SES—успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в массив STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию содержимого поля GROUP; вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0; если таких студентов нет, вывести соответствующее сообщение.

Самостоятельная работа №2.

Простой список можно описать следующим образом:

- простой список может содержать ноль или более элементов определенного типа;
- можно создавать пустой список;
- можно добавлять элемент в список;
- можно определять, пуст ли список;
- можно определять, полон ли список.
- можно посетить каждый элемент списка и выполнить над ним определенное действие.

Спроектируйте класс List для представления этого абстрактного типа. Вы должны подготовить заголовочный файл list.h с объявлением класса и файл list.cpp с реализацией его методов. Вы должны также написать короткую программу, которая будет использовать полученный класс.

Самостоятельная работа №3.

Описать класс «обыкновенная дробь», содержащий числитель и знаменатель (целые числа). В классе предусмотреть конструкторы с параметрами и без параметров. Предусмотреть метод info(), выводящий на экран информацию об объекте. Предусмотреть перегруженные операторы «==» и «!=», сравнивающие две дроби. Написать демонстрационную программу, создающую объекты и проверяющую работу перегруженных операторов.

Самостоятельная работа №4.

Описать класс «массив вещественных чисел». В классе предусмотреть методы удаления и добавления элементов (используя выделение динамической памяти). Предусмотреть метод `info()`, выводящий на экран информацию об объекте. Предусмотреть перегруженный оператор «`/`», выполняющий деление всех элементов массива на число. Написать демонстрационную программу, создающую объекты и проверяющую работу перегруженного оператора.

Самостоятельная работа №5.

Описать класс «Автобус». Написать программу моделирования работы автобусного парка. Программа должна обеспечивать начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла). Кроме того, программа должна обеспечивать имитацию выезда автобуса из парка: вводится номер автобуса, программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке и записывает данные в список автобусов, находящихся на маршруте. Для представления необходимых списков использовать шаблонный класс `vector`.

Самостоятельная работа №6.

Напишите программу, запрашивающую у пользователя ввод двух целых чисел. Затем программа должна вычислить и выдать сумму всех целых чисел, лежащих между этими двумя целыми. Предполагается, что меньшее значение вводится первым. Например, если пользователь ввел 2 и 9, программа должна сообщить, что сумма всех целых чисел от 2 до 9 равна 44.

Самостоятельная работа №7.

Предположим, что вы продаете книгу по программированию на языке C++ для начинающих. Напишите программу, которая позволит ввести ежемесячные объемы продаж в течение года (в количестве книг, а не в деньгах). Программа должна использовать цикл, в котором выводится приглашение с названием месяца, применяя массив указателей на `char` (или массив объектов `string`, если вы предпочитаете его), инициализированный строками — названиями месяцев, и сохраняя введенные значения в массиве `int`. Затем программа должна найти сумму содержимого массива и выдать общий объем продаж за год.

Самостоятельная работа №8.

Определить шаблон функции, осуществляющей сортировку элементов массива по убыванию. Продемонстрировать работу шаблона на массивах объектов как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

Самостоятельная работа №9.

Написать программу «Англо-русский и русско-английский словарь». Программа должна обеспечивать загрузку в контейнер типа `map` базы англо-русского перевода и формирование по нему другого контейнера типа `map` – базы русско-английского перевода. Реализовать функции вывода вариантов перевода заданного слова на альтернативный язык.

Самостоятельная работа №10.

Найти количество элементов матрицы равных нулю. Количество нулевых элементов по каждой строке матрицы рассчитывается отдельным потоком.

Самостоятельная работа №11.

Организовать программу, позволяющую имитировать работу банкоматов (работающих только на выдачу денег) отделения банка. На одном банкомате может обслуживаться одновременно только один клиент, у каждого банкомата есть резерв денежных средств и лимит на единовременную выдачу. У каждого клиента есть счёт с определенным балансом, клиент может снимать деньги в количестве не превышающим лимит банкомата и размер остатка на счету с банкомата несколько раз. После каждого снятия деньги списываются со счета клиента. Программа должна выводить на экран фио или номер клиента и данные количестве снимаемых денег. Работа программы заканчивается тогда, когда заканчиваются деньги в автомате.

Рекомендации по работе с литературой

Для более эффективного освоения и усвоения материала рекомендуется ознакомиться с теоретическим материалом по той или иной теме до проведения лабораторного занятия. Всю учебную литературу желательно изучать «под конспект».

Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

Работу с теоретическим материалом по теме можно проводить по следующей схеме:

- название темы;
- цели и задачи изучения темы;
- основные вопросы темы;
- характеристика основных понятий и определений, необходимых для усвоения данной темы;

- краткие выводы, ориентирующие на определенную совокупность сведений, основных идей, ключевых положений, систему доказательств, которые необходимо усвоить.

При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении консультаций, либо в индивидуальном порядке.

Подготовка к лабораторным работам

Подготовку к каждой лабораторной работе каждый студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы по теме задания, правильном выполнении лабораторной работы.

В процессе выполнения лабораторной работы студент должен создать требуемый документ (отчет) с помощью предлагаемого программного средства и выполнить требуемые в задании операции. Задание по лабораторной работе содержит методические указания по подготовке документа, который должен быть получен в результате выполнения работы. При подготовке к лабораторной работе следует их внимательно прочесть.

При формировании индивидуальных заданий по самостоятельной работе используется материал из следующих источников: Программирование: метод. указания/ сост. Л.И.Прудникова - Владивосток: Изд-во ДВФУ, 2012. - 24 с., C/C++. Структурное и объектно-ориентированное программирование: практикум / Т.А. Павловская, Ю.А. Щупак.- СПб.: Питер, 2011.- 352 с., Программирование на языке C++ в среде Qt Creator/ Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк, О. В. Чеснокова, А. С. Чмыхало - М. : ALT Linux, 2015. — 448 с.

Содержание отчета по лабораторной работе

1. Титульный лист.
2. Постановка задачи.
3. Текст программы с комментариями.
4. Тестовые условия.