МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО Руководитель ОПОП	УТВЕРЖДАЮ Зав. кафедрой ПМИ
_____М.В. Коломина	_____М.В. Коломина
«8» сентября 2022 г.	«8» сентября 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Парадигмы программирования»

Составители	**Станкевич А.С., к.т.н., доцент ФИТиП, ИТМО**
Направление подготовки / специальность	**01.03.02 Прикладная математика и информатика**
Направленность (профиль) ОПОП	**Программирование и искусственный интеллект**
Квалификация (степень)	**бакалавр**
Форма обучения	**очная**
Год приёма	**2023**
Курс	**1**
Семестр(ы)	**2**

Астрахань – 2022

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Целью освоения дисциплины «Парадигмы программирования» является ознакомление студентов с основными понятиями, подходами и принципами программирования.

1.2. Задачами освоения дисциплины являются углубление знаний в области программирования, развитие практических навыков в области прикладной математики и информатики, подготовка к изучению других дисциплин.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина «Парадигмы программирования» относится к обязательной части и осваивается во 2 семестре.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс освоения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки / специальности:

а) общепрофессиональных (ОПК)

- ОПК-2. Способен осуществлять профессиональную деятельность с учетом экономических, финансовых, экологических, интеллектуально-правовых, социальных, этических и других ограничений на всех этапах жизненного цикла объектов профессиональной деятельности и процессов на основе оценки их эффективности и результатов

б) профессиональных (ПК).

- ПК-1. Способен создавать, отлаживать и оформлять программный код
- ПК-2. Способен осуществлять интеграцию программных модулей и компонент и проверку работоспособности кода программного обеспечения
- ПК-6. Способен выполнять работы по созданию и сопровождению информационных систем

Таблица 1 – Декомпозиция результатов обучения

Код и наименование компетенции	Планируемые результаты обучения по дисциплине (модулю)		
	Знать (1)	Уметь (2)	Владеть (3)
ОПК-2.1 Обосновывает принятие решения при осуществлении профессиональной деятельности ОПК-2.2 Выбирает средства и технологии, в том числе с учетом последствий их применения в профессиональной сфере. Исследуют границы применения определенных решений в рамках профессиональной деятельности ОПК-2.3 Принимает участие в планировании, разработке текущих и перспективных планов развития проектов в профессиональной области ОПК-2.4 Оценивает эффективность результатов профессиональной деятельности	ИОПК-2.1.1. Знания: методы принятия решений, методы оценки эффективности результатов, методы и технологии работы с информацией.	ИОПК-2.2.1. Обосновывать принятие решения, выбирать средства и технологии с учетом последствий их применения, принимать участие в планировании, разработке текущих и перспективных планов развития проектов, оценивать эффективность результатов в профессиональной деятельности, определять приоритеты профессиональной деятельности и способы ее совершенствования.	ИОПК-2.3.1. Практическими навыками обоснования принятых решений, планирования и разработки текущих и перспективных планов развития проектов, оценки эффективности результатов профессиональной деятельности

ОПК-2.5 Определяет приоритеты профессиональной деятельности и способы ее совершенствования			
ПК-1.1. Способен осуществлять формализацию и алгоритмизацию поставленных задач. ПК-1.2. Способен создавать программный код с использованием языков программирования, определения и манипулирования данными. ПК-1.3. Способен работать с системой контроля версий, оформлять в соответствии с требованиями, проверять и отлаживать программный код.	ИПК-1.1.1. Языки программирования и алгоритмы и структуры данных, методы работы с данными, системы контроля версий.	ИПК-1.2.1. Осуществлять формализацию и алгоритмизацию поставленных задач, создавать программный код с использованием языков программирования, работать с системой контроля версий, оформлять в соответствии с требованиями, проверять и отлаживать программный код.	ИОПК-1.3.1. Практическими навыками создания, отлаживания и оформления программного кода.
ПК-2.1. Способен разрабатывать тестовые наборы данных. ПК-2.2. Способен проверять работоспособность программного обеспечения ПК-2.3. Способен осуществлять интеграцию программных модулей и компонентов и верификацию выпусков программного продукта	ИПК-2.1.1. Программные продукты, программные модули и компоненты верификации выпусков программного продукта, принципы построения и функционирования процессора, оперативной памяти и внешних устройств.	ИПК-2.2.1. Проверять работоспособности программного обеспечения.	ИОПК-2.3.1. Навыками осуществления интеграции программных модулей
ПК-6.1. Способен осуществлять кодирование на языках программирования. ПК-6.2. Способен осуществлять установку и настройку системного и прикладного ПО, необходимого для функционирования ИС ПК-6.3. Способен осуществлять настройку оборудования, необходимого для работы ИС	ИПК-6.1.1. Языки программирования.	ИПК-6.2.1. Кодировать на языках программирования, осуществлять установки и настройки системного и прикладного по.	ИПК-6.3.1. Навыками создания и сопровождениея информационных систем, поддержки локально развернутых баз данных, поиска и устранения ошибок в работе базы данных.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Объём дисциплины составляет 4 зачётных единицы, в том числе 72 часа, выделенных на контактную работу обучающихся с преподавателем (из них 36 часов – лекции, 36 часов – лабораторные работы), и 72 часа – на самостоятельную работу обучающихся.

Таблица 2 – Структура и содержание дисциплины

Раздел, тема дисциплины	Семестр	Контактная работа (в часах)			Самост. работа		Форма текущего контроля успеваемости, форма промежуточной аттестации *[по семестрам]*
		Л	ПЗ	ЛР	КР	СР	
Основы языка С++	2	18		**18**		36	Лабораторная работа №1, 2, 3
Углубленные знания языка С++	2	18		18		36	Лабораторная работа №5, 6, 7
Итого		**36**		**36**		**72**	**Диф. зачет**

Таблица 3 – Матрица соотнесения разделов, тем учебной дисциплины (модуля) и формируемых компетенций

Раздел, тема дисциплины (модуля)	Кол-во часов	Код компетенции				Общее количество компетенций
		ОПК2	ПК-1	ПК-2	ПК-6	
Основы языка С++	72	+	+	+	+	4
Углубленные знания языка С++	72	+	+	+	+	4
Итого	144					4

Краткое содержание каждой темы дисциплины (модуля)

Основы языка С++

Ссылки, пространства имен. Перегрузка имен функций. Объявление переменных. Перегрузка операторов. Возвращение значения по ссылке. Классы. Контроль доступа к полям. Конструкторы и деструкторы. Механизмы ООП: абстракция, инкапсуляция, наследование, полиморфизм. Объектно-ориентированный подход к разработке программ., История языка Си . Отличия от Си.

Углубленные знания языка С++

Полиморфизм. Виртуальные методы. Порождение и обработка исключений. Шаблоны языка Си . Потоки ввода-вывода стандартной библиотеки. Структуры данных стандартной библиотеки и методы их обработки.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю)

Лекционные занятия

Основной формой реализации теоретического обучения является лекция, которая представляет собой систематическое, последовательное изложение преподавателем-лектором учебного материала теоретического характера. Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Порядок подготовки лекционного занятия включает в себя выполнение следующих этапов:

- изучение требований программы дисциплины;
- определение целей и задач лекции;
- разработка плана проведения лекции;
- подбор литературы (ознакомление с методической литературой, публикациями периодической печати по теме лекционного занятия);
- отбор необходимого и достаточного по содержанию учебного материала;
- определение методов, приемов и средств поддержания интереса, внимания, стимулирования творческого мышления студентов;
- написание конспекта лекции.

Лекция должна включать следующие разделы:

- формулировку темы лекции;
- указание основных изучаемых разделов или вопросов и предполагаемых затрат времени на их изложение;
- изложение вводной части;
- изложение основной части лекции;
- краткие выводы по каждому из вопросов;
- заключение;
- рекомендации литературных источников по излагаемым вопросам.

Лабораторные занятия

Лабораторное занятие − целенаправленная форма организации педагогического процесса, направленная на углубление научно-теоретических знаний и овладение определенными методами работы, в процессе которых вырабатываются умения и навыки выполнения тех или иных учебных действий в данной сфере науки. Они развивают научное мышление и речь, позволяют проверить знания студентов и выступают как средства оперативной обратной связи.

Правильно организованные лабораторные занятия ориентированы на решение следующих задач:

- обобщение, систематизация, углубление, закрепление полученных на лекциях и в процессе самостоятельной работы теоретических знаний по дисциплине (предмету);
- формирование практических умений и навыков, необходимых в будущей профессиональной деятельности, реализация единства интеллектуальной и практической деятельности;
- выработка при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Состав заданий для лабораторного занятия должен быть спланирован с расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством учащихся.

Лабораторные занятия должны так быть организованы, чтобы студенты ощущали нарастание сложности выполнения заданий, испытывали бы положительные эмоции от переживания собственного успеха в учении, поисками правильных и точных решений.

Самостоятельная работа

Самостоятельная работа – это вид учебной деятельности, которую студент совершает в установленное время и в установленном объеме индивидуально или в группе, без непосредственной помощи преподавателя (но при его контроле), руководствуясь сформированными ранее представлениями о порядке и правильности выполнения действий.

В учебном процессе образовательного учреждения выделяются два вида самостоятельной работы:

- аудиторная – выполняется на учебных занятиях, под непосредственным руководством преподавателя и по его заданию (выполнение самостоятельных работ; выполнение контрольных и практических работ; решение задач);
- внеаудиторная – выполняется по заданию преподавателя, но без его непосредственного участия (подготовка к аудиторным занятиям; изучение учебного материала, вынесенного на самостоятельную проработку; выполнение домашних заданий разнообразного характера; выполнение индивидуальных заданий, направленных на развитие у студентов самостоятельности и инициативы; подготовка к контрольной работе). Внеаудиторные самостоятельные работы представляют собой логическое продолжение аудиторных занятий, проводятся по заданию преподавателя, который инструктирует студентов и устанавливает сроки выполнения задания.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Лекция

- Лекция – основной вид обучения в вузе.
- В лекции излагаются основные положения теории, ее понятия и законы, приводятся факты, показывающие связь теории с практикой.
- Накануне лекции необходимо повторить содержание предыдущей лекции (а также теорию по изучаемой теме в школьных учебниках геометрии, если эта тема была представлена в них), а затем посмотреть тему очередной лекции по программе (по плану лекций).
- Полезно вести записи (конспекты) лекций: для непонятных вопросов оставлять место при работе над темой лекции с учебными пособиями.

- Записи лекций следует вести в отдельной тетради, оставляя место для дополнений во время самостоятельной работы.
- При конспектировании лекций выделяйте главы и разделы, параграфы, подчеркивайте основное.

Лабораторное занятие

- Лабораторное занятие – наиболее активный вид учебных занятий в вузе. Он предполагает самостоятельную работу над лекциями и учебными пособиями.
- К каждому лабораторному занятию нужно готовиться. Подготовку следует начинать с повторения теории (по записям лекций или по учебному пособию). После этого нужно решать задачи из предложенного домашнего задания.

Организация самостоятельной работы

Самостоятельность в учебной работе способствует развитию заинтересованности студента в изучаемом материале, вырабатывает у него умение и потребность самостоятельно получать знания, что весьма важно для специалиста с высшим образованием. Самостоятельная работа студентов представлена в следующих формах:

- работа с учебной литературой и конспектом лекций с целью подготовки к лабораторным занятиям, составление конспектов тем, выносимых на самостоятельную проработку;
- систематическое выполнение домашних работ.

Таблица 4 – Содержание самостоятельной работы обучающихся

Номер раздела (темы)	Темы/вопросы, выносимые на самостоятельное изучение	Кол-во часов	Форма работы
Раздел 1	Основы языка C++	36	Изучение теоретического материала. Подготовка к лабораторным работам.
Раздел 2	Углубленные знания языка C++	36	Изучение теоретического материала. Подготовка к лабораторным работам.

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины (модуля), выполняемые обучающимися самостоятельно

Лабораторные работы выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.

Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.

Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Зачет проводится в форме устного ответа на вопрос, а также выполнения практико-ориентированного задания письменно на листе или в электронном виде с использованием компьютера.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

При реализации различных видов учебной работы по дисциплине «Парадигмы программирования» могут использоваться электронное обучение и дистанционные образовательные технологии.

6.1. Образовательные технологии

Учебные занятия по дисциплине могут проводиться с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) интерактивном взаимодействии обучающихся и преподавателя в режимах on-line или off-line в формах.

№	Формы	Описание
1	Лекция-дискуссия	Лекция-дискуссия специально не назначается, а возникает достаточно спонтанно на большинстве лекций. Студенты устно высказываю своѐ мнение по ходу лекции, дискутируют как с лектором, так и между собой. Также дискуссии иногда возникают при защите лабораторных работ.
2	Исследовательские методы в обучении	Дает возможность учащимся самостоятельно пополнять свои знания, глубоко вникать в изучаемую проблему и предполагать пути ее решения, что важно при формировании мировоззрения.
3	Самостоятельная работа	Работа с ресурсами Internet, подготовка к лабораторным работам

6.2. Информационные технологии

При реализации различных видов учебной и внеучебной работы используются следующие информационные технологии:

- система управления обучением LMS Moodle;
- использование возможностей Интернета в учебном процессе (рассылка заданий, предоставление выполненных работ, ответы на вопросы, ознакомление обучающихся с оценками и т.д.);
- использование электронных учебников и различных сайтов (например, электронные библиотеки, журналы и т.д.) как источник информации;
- использование возможностей электронной почты;
- использование средств представления учебной информации (электронных учебных пособий, применение новых технологий для проведения занятий с использованием презентаций и т.д.);
- использование интерактивных средств взаимодействия участников образовательного процесса (технологии дистанционного или открытого обучения в глобальной сети);
- использование интегрированных образовательных сред, где главной составляющей являются не только применяемые технологии, но и содержательная часть, т.е. информационные ресурсы (доступ к мировым информационным ресурсам, на базе которых строится учебный процесс).

Перечень информационных справочных систем:

1. Электронная библиотека «Астраханский государственный университет» собственной генерации на платформе ЭБС «Электронный Читальный зал – БиблиоТех». https://biblio.asu.edu.ru
2. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем».
3. https://library.asu.edu.ru
4. Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента». www.studentlibrary.ru
5. Электронная библиотечная система издательства ЮРАЙТ, раздел «Легендарные книги». www.biblio-online.ru, https://urait.ru/
6. Электронная библиотечная система IPRbooks. www.iprbookshop.ru

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине «Парадигмы программирования» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением

дисциплин и прохождением практик, а в процессе освоения дисциплины– последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 5 – Соответствие разделов, тем дисциплины, результатов обучения по дисциплине и оценочных средств

№ п/п	Контролируемые разделы, темы дисциплины (модуля)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Основы языка С++	ОПК-2, ПК-1, ПК-2, ПК-6	лабораторные работы
2	Углубленные знания языка С++	ОПК-2, ПК-1, ПК-2, ПК-6	лабораторные работы

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Таблица 6 – Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует глубокое знание теоретического материала, умение обоснованно излагать свои мысли по обсуждаемым вопросам, способность полно, правильно и аргументированно отвечать на вопросы, приводить примеры
4 «хорошо»	демонстрирует знание теоретического материала, его последовательное изложение, способность приводить примеры, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует неполное, фрагментарное знание теоретического материала, требующее наводящих вопросов преподавателя, допускает существенные ошибки в его изложении, затрудняется в приведении примеров и формулировке выводов
2 «неудовлетворительно»	демонстрирует существенные пробелы в знании теоретического материала, не способен его изложить и ответить на наводящие вопросы преподавателя, не может привести примеры

Таблица 7 – Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы
4 «хорошо»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует отдельные, несистематизированные навыки, испытывает затруднения и допускает ошибки при выполнении заданий, выполняет задание по подсказке преподавателя, затрудняется в формулировке выводов
2 «неудовлетворительно»	не способен правильно выполнить задания

7.3. Контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Типовые контрольные задания, необходимые для оценки достижения запланированных результатов обучения приведены в таблице планирования результатов обучения по дисциплине (БаРС) (Приложение 1)*.

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторная работа 1

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.
Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Пример заданий для лабораторной работы 1 "Разработка классов, иерархии классов, перегрузка операторов"

1. Необходимо написать класс длинное знаковое число.

Требования к решению следующие: Реализация должна содержаться в классе big_integer и находиться в файле big_integer.h.
Класс должен содержать: Конструктор по умолчанию, инициализирующий число нулем. Конструктор копирования, после которого исходный объект и копию можно изменять независимо. Конструктор от int. Explicit конструктор от std::string. Оператор присваивания. Операторы сравнения.
Арифметические операции: сложение, вычитание, изменение знака (унарный минус), унарный плюс. Умножение работающее за время O(nm), где n,m — длины множителей в битах. Деление и остаток от деления работающие за время O(nm), где n — длина делителя в битах, а m — длина частного в битах. Префиксный/постфиксный инкремент/декремент. (опционально)
Битовые операции: и, или, исключающее или, не. Битовые сдвиги.
Должна существовать глобальная функция std::string to_string(big_integer const&) возвращающая строковое представление числа. Реализация функций класса должна быть расположена в файле big_integer.cpp.
Пользоваться сторонними библиотеками длинных чисел при выполнении этого задания нельзя. Разряды числа должны представляться 32-битными либо 64-битными числами. При этом необходимо, чтобы все биты в их представлении использовались. Реализация должна использовать арифметические операции той битности, которая естественна для данного представления разрядов.

2. Првести проверку корректности программы с помощью набора модульных тестов: https://github.com/sorokin/cpp-course/tree/master/bigint.

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных ии выходных параметров:

0. Выводы:

Шкала оценивания и критерии оценки:

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				

3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		**10**	**5**	**0**

Лабораторная работа 2

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.
Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Пример заданий для лабораторной работы 2 "Использование интерфейсов"

1. Необходимо реализовать класс длинного числа со small-object и copy-on-write оптимизациями. Длинное число должно удовлетворять всем требованиям: Если в big_integer хранится число $-2N \le a \le 2N-1$, big_integer не должен выделять динамическую память. Конкретное значение N вы можете выбрать самостоятельно, допустимыми являются любые $N \ge 30$. Числа $a < -2N$ и $a > 2N-1$ должны выделять не больше одного блока динамической памяти на каждый экземпляр big_integer. Конструктор копирования и оператор присваивания должны работать за O(1) и удовлетворять гарантии безопасности исключений nothrow

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных и выходных параметров:

0. Выводы:

Шкала оценивания и критерии оценки:

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				
3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		**10**	**5**	**0**

Лабораторная работа 3

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.
Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Пример заданий для лабораторной работы 2 "Использование интерфейсов"

1. Необходимо реализовать класс длинного числа со small-object и copy-on-write оптимизациями. Длинное число должно удовлетворять всем требованиям: Если в big_integer хранится число $-2N \le a \le 2N-1$, big_integer не должен выделять динамическую память. Конкретное значение N вы можете выбрать самостоятельно, допустимыми являются любые $N \ge 30$. Числа $a < -2N$ и $a > 2N-1$ должны выделять не больше одного блока динамической памяти на каждый экземпляр big_integer. Конструктор копирования и оператор присваивания должны работать за O(1) и удовлетворять гарантии безопасности исключений nothrow

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных ии выходных параметров:
5. Выводы:

Шкала оценивания и критерии оценки:

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				
3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		**10**	**5**	**0**

Лабораторная работа 4

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.

Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Пример заданий для лабораторной работы 4 "Работа со стандартной библиотекой, шаблонами"

1. Шаблонный класс Array может хранить объекты любого типа, для которого определён конструктор копирования, в том числе и другой Array, например, Array< Array<int> >. Глубина вложенности может быть произвольной. Напишите шаблонную функцию (или несколько) flatten, которая принимает на вход такой "многомерный" Array неизвестной заранее глубины вложенности и выводит в поток out через пробел все элементы, хранящиеся на самом нижнем уровне.

Примеры работы функции flatten:

```
Array<int> ints(2, 0);
ints[0] = 10;
ints[1] = 20;
flatten(ints, std::cout); // выводит на экран строку "10 20"
Array< Array<int> > array_of_ints(2, ints);
flatten(array_of_ints, std::cout);
// выводит на экран строку "10 20 10 20"
Array<double> doubles(10, 0.0);
flatten(doubles, std::cout);
// работать должно не только для типа int
```

Примечание: лидирующие и завершающие пробельные символы будут игнорироваться, т. е. там, где ожидается "10 20" будет так же принят, например, вариант " 10 20 ", но не вывод "1020".

Подсказка: шаблонные функции тоже можно перегружать, из нескольких шаблонных функций будет выбрана наиболее специфичная.

```
#include <iostream>
// Весь вывод должен осущствляться в поток out,
// переданный в качестве параметра.
//
// Можно заводить любые вспомогаетльные функции,
// структуры или даже изменять сигнатуру flatten,
// но при этом все примеры вызова из задания должны
// компилироваться и работать.
```

2. Выше вы реализовали простой шаблон ValueHolder, в этом задании мы используем его чтобы написать класс Any (интересно, что не шаблонный), который позволяет хранить значения любого типа. Например, вы сможете создать массив объектов типа Any, и сохранять в них int-ы, double-ы или даже объекты Array. Подробности в шаблоне кода.

Подсказка: в нешаблонном классе Any могут быть шаблонные методы, например, шаблонный конструктор.

```
struct ICloneable;
// Поле data_ типа T в классе ValueHolder
// открыто, к нему можно обращаться

template <typename T>
struct ValueHolder;
// Это класс, который вам нужно реализовать

class Any
{
public:
// В классе Any должен быть конструктор, который можно вызвать
// без параметров, чтобы работал следующий код:
// Any empty; // empty ничего не хранит

// В классе Any должен быть шаблонный конструктор от одного
// параметра, чтобы можно было создавать объекты типа Any,
// например, следующим образом:
// Any i(10); // i хранит значение 10

// Деструктор: выделенные ресурсы нужно освободить.

// В классе Any также должен быть конструктор копирования (вам
// поможет метод clone интерфейса ICloneable)

// В классе должен быть оператор присваивания и/или шаблонный
// оператор присваивания, чтобы работал следующий код:
// Any copy(i); // copy хранит 10, как и i
// empty = copy; // empty хранит 10, как и copy
// empty = 0; // а теперь empty хранит 0

// Чтобы получать хранимое значение, определите в классе Any
// шаблонный метод cast, который возвращает указатель на
// хранимое значение, или нулевой указатель в случае
// несоответствия типов или если объект Any ничего не хранит:
// int *iptr = i.cast<int>(); // *iptr == 10
// char *cptr = i.cast<char>(); // cptr == 0,
// // потому что i хранит int, а не char
// Any empty2;
// int *p = empty2.cast<int>(); // p == 0
// При реализации используйте dynamic_cast.
};
```

3. В качестве упражнения на частичную специализацию шаблонов классов вам предлагается реализовать простой шаблон SameType. Этот шаблон не содержит никаких методов, а только одно статическое константное поле типа bool с именем value. Шаблон принимает два типовых параметра, и если два типовых параметра шаблона являются одним и тем же типом, то статическое поле value должно хранить значение true, в противном случае – значение false.

Примеры:

```
struct Dummy { };
typedef int type;
std::cout << SameType<int, int>::value << std::endl;
// выведет 1, т. е. true
std::cout << SameType<int, type>::value << std::endl;
// 1, type == int
std::cout << SameType<int, int&>::value << std::endl;
// 0, int и ссылка на int - различные типы
std::cout << SameType<Dummy, Dummy>::value << std::endl; // 1
std::cout << SameType<int, const int>::value << std::endl;
// 0, const - часть типа

// Определите шаблон SameType с двумя типовыми параметрами.
// В шаблоне должна быть определена одна статическая константа // типа bool с именем
value
```

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных ии выходных параметров:

0. Выводы:

Шкала оценивания и критерии оценки (на примере одной лабораторной работы 4 модуля):

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				
3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		**10**	**5**	**0**

Лабораторная работа 5

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.
Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Пример заданий для лабораторной работы 4 «Работа со стандартной библиотекой, шаблонами»

1. Шаблонный класс Array может хранить объекты любого типа, для которого определён конструктор копирования, в том числе и другой Array, например, Array< Array<int> >. Глубина вложенности может быть произвольной. Напишите шаблонную функцию (или несколько) flatten, которая принимает на вход такой "многомерный" Array неизвестной заранее глубины вложенности и выводит в поток out через пробел все элементы, хранящиеся на самом нижнем уровне.

Примеры работы функции flatten:

```
Array<int> ints(2, 0);
ints[0] = 10;
ints[1] = 20;
flatten(ints, std::cout); // выводит на экран строку "10 20"
Array< Array<int> > array_of_ints(2, ints);
flatten(array_of_ints, std::cout);
// выводит на экран строку "10 20 10 20"
Array<double> doubles(10, 0.0);
flatten(doubles, std::cout);
// работать должно не только для типа int
```

Примечание: лидирующие и завершающие пробельные символы будут игнорироваться, т. е. там, где ожидается «10 20» будет так же принят, например, вариант «10 20», но не вывод «1020».

Подсказка: шаблонные функции тоже можно перегружать, из нескольких шаблонных функций будет выбрана наиболее специфичная.

```
#include <iostream>
// Весь вывод должен осущствляться в поток out,
// переданный в качестве параметра.
//
// Можно заводить любые вспомогаетльные функции,
// структуры или даже изменять сигнатуру flatten,
// но при этом все примеры вызова из задания должны
// компилироваться и работать.
```

2. Выше вы реализовали простой шаблон ValueHolder, в этом задании мы используем его чтобы написать класс Any (интересно, что не шаблонный), который позволяет хранить значения любого типа. Например, вы сможете создать массив объектов типа Any, и сохранять в них int-ы, double-ы или даже объекты Array. Подробности в шаблоне кода.

Подсказка: в нешаблонном классе Any могут быть шаблонные методы, например, шаблонный конструктор.

```
struct ICloneable;
// Поле data_ типа T в классе ValueHolder
// открыто, к нему можно обращаться

template <typename T>
struct ValueHolder;
// Это класс, который вам нужно реализовать

class Any
{
public:
// В классе Any должен быть конструктор, который можно вызвать
// без параметров, чтобы работал следующий код:
// Any empty; // empty ничего не хранит

// В классе Any должен быть шаблонный конструктор от одного
// параметра, чтобы можно было создавать объекты типа Any,
// например, следующим образом:
// Any i(10); // i хранит значение 10

// Деструктор: выделенные ресурсы нужно освободить.

// В классе Any также должен быть конструктор копирования (вам
// поможет метод clone интерфейса ICloneable)

// В классе должен быть оператор присваивания и/или шаблонный
// оператор присваивания, чтобы работал следующий код:
// Any copy(i); // copy хранит 10, как и i
// empty = copy; // empty хранит 10, как и copy
// empty = 0; // а теперь empty хранит 0

// Чтобы получать хранимое значение, определите в классе Any
// шаблонный метод cast, который возвращает указатель на
// хранимое значение, или нулевой указатель в случае
// несоответствия типов или если объект Any ничего не хранит:
// int *iptr = i.cast<int>(); // *iptr == 10
// char *cptr = i.cast<char>(); // cptr == 0,
// // потому что i хранит int, а не char
// Any empty2;
// int *p = empty2.cast<int>(); // p == 0
// При реализации используйте dynamic_cast.
};
```

3. В качестве упражнения на частичную специализацию шаблонов классов вам предлагается реализовать простой шаблон SameType. Этот шаблон не содержит никаких методов, а только одно статическое константное поле типа bool с именем value. Шаблон принимает два типовых параметра, и если два типовых параметра шаблона являются одним и тем же типом, то статическое поле value должно хранить значение true, в противном случае – значение false.

Примеры:

```
struct Dummy { };
typedef int type;
std::cout << SameType<int, int>::value << std::endl;
// выведет 1, т. е. true
std::cout << SameType<int, type>::value << std::endl;
// 1, type == int
std::cout << SameType<int, int&>::value << std::endl;
// 0, int и ссылка на int - различные типы
std::cout << SameType<Dummy, Dummy>::value << std::endl; // 1
std::cout << SameType<int, const int>::value << std::endl;
// 0, const - часть типа

// Определите шаблон SameType с двумя типовыми параметрами.
// В шаблоне должна быть определена одна статическая константа // типа bool с именем
value
```

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных ии выходных параметров:
5. Выводы:

Шкала оценивания и критерии оценки (на примере одной лабораторной работы 4 модуля):

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				
3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		**10**	**5**	**0**

Лабораторная работа 6

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.
Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.
Пример заданий для лабораторной работы 6 "Написание подпрограмм"

1. Написать программу для вычисления по указанной формуле. Проверить полученный ответ с помощью программы MathCAD.

1) $\left(\frac{1}{7}+\ln\sqrt{x}\right)\cdot e^{\sqrt{x-2}};$

2) $\frac{x^3}{\sqrt{3}}-e^x\ln\left|1.37^3+x^3\right|+\frac{4}{3};$

3) $\frac{\sqrt{x}\sin\frac{x^2}{2}-1.3}{\sqrt[5]{x}+e^{3x}+|\cos x|};$

4) $\frac{\ln\sqrt{\pi+|2-x|}}{3-1/x}+\sqrt[3]{x^2}\cdot\sin 1.4x;$

5) $\sqrt{e^{|\sin^3 x|}}+2\ln 3x-\frac{1}{9};$

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных ии выходных параметров:

0. Выводы:

Шкала оценивания и критерии оценки (на примере одной лабораторной работы 4 модуля):

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				
3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		10	5	0

Тест 1

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.
Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Пример заданий для лабораторной работы 6 "Написание подпрограмм"

1. Написать программу для вычисления по указанной формуле. Проверить полученный ответ с помощью программы MathCAD.

1) $\left(\frac{1}{7}+\ln\sqrt{x}\right)\cdot e^{\sqrt{x-2}};$

2) $\frac{x^3}{\sqrt{3}}-e^x\ln\left|1.37^3+x^3\right|+\frac{4}{3};$

3) $\frac{\sqrt{x}\sin\frac{x^2}{2}-1.3}{\sqrt[5]{x}+e^{3x}+|\cos x|};$

4) $\frac{\ln\sqrt{\pi+|2-x|}}{3-1/x}+\sqrt[3]{x^2}\cdot\sin 1.4x;$

5) $\sqrt{e^{|\sin^3 x|}}+2\ln 3x-\frac{1}{9};$

Отчет по лабораторной работе №____

1. Цель и задачи лабораторной работы:
2. Текстовое описание способа реализации кода:
3. Текст кода:
4. Результаты работы программы в виде набора входных ии выходных параметров:

0. Выводы:

Шкала оценивания и критерии оценки (на примере одной лабораторной работы 4 модуля):

№ п/п	Показатели	Оценка (уровень)		
		высокий	средний	низкий
Выполнение лабораторной работы				
1	Уровень оформления отчета	2	1	0
2	Понимание работы программы на уровне анализа отдельных элементов кода	3	1	0
Защита отчета по лабораторной работе				
3	Умение модифицировать программный код по новым требованиям в рамках задания	3	1,5	0
4	Умение обосновывать применяемые алгоритмов, методов и решений	2	1,5	0
Итого баллов:		**10**	**5**	**0**

Тест 2

Описание технологии проведения теста:

- Тест проводится в письменной форме
- На один тест отводится двадцать минут
- Тест содержит 10 вопросов
- Каждый вопрос имеет 4 варианта ответа

Примеры тестовых вопросов

Вопрос 1.

Дан код

```
#include <iostream>
template<class T>
void f(T) { std::cout << 1; }
template<>
void f<>(int*) { std::cout << 2; }
template<class T>
void f(T*) { std::cout << 3; }
int main() {
  int *p = nullptr;
  f( p );
}
```

Что выведет программа

1. 1
2. 2
3. Ошибка компиляции
4. Результат не определен

Правильный ответ на менее чем 3 вопроса – 0 баллов

7.4. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

Дифференцированный зачет

Зачет проводится в форме устного ответа на вопрос, а также выполнения практико-ориентированного задания письменно на листе или в электронном виде с использованием компьютера.

Теоретические вопросы:

1. Встроенные типы данных. Представление чисел в памяти. Прямой, обратный и дополнительный код. Числа с плавающей точкой.
2. Хранение данных в памяти. Доступ к данным в памяти. Указатели. Операции с указателями.
3. Массивы. Объявление и инициализация. Устройство в памяти. Операции над массивами. Двумерные массивы. Арифметика указателей.
4. Ссылки. Разница между ссылкой и указателем. Зачем в C++ были добавлены ссылки.
5. Сборка программ. Препроцессор, компилятор, компоновщик. Сборка проекта, состоящего из нескольких файлов. Макросы
6. Запуск программ. Процессы и потоки. Виды памяти. Стек вызова функций.
7. ООП. Базовые принципы. Абстракция, инкапсуляция, наследование полиморфизм. Принцип подстановки Барбары-Лисков
8. ООП. Классы. Устройство в памяти. Инкапсуляция. Модификаторы доступа. Наследование. Множественное наследование. Проблемы множественного наследование.
9. ООП. Классы. Инициализация и уничтожение. Виды конструкторов Виртуальные методы. Таблица виртуальных функций. Виртуальные деструкторы.
10. ООП. Операторы. Перегрузка операторов.
11. Исключения. Обработка ошибок в Си. Assert. Исключения. Преимущества и недостатки исключений.
12. Шаблоны. Шаблоны функций, классов.
13. Шаблоны. Частичная специализация и полная специализация
14. Variadic template
15. Преобразования типов. Явные и неявные. C-cast, static_cast, dynamic_cast, const_cast, reinterpret_cast. CRTP

16. Стандартная библиотека. Итераторы. Алгоритмы. Основные классы алгоритмов.
17. Стандартная библиотека. Контейнеры. Последовательные контейнеры.
18. Стандартная библиотека. Контейнеры. Ассоциативные контейнеры.
19. Стандартная библиотека. Адаптеры.
20. Умные указатели. auto_ptr, unique_ptr, shared_ptr, weak_ptr. Устройство, преимущества и недостатки
21. Rvalue ссылки, Move семантика.

Практические задания:

1. Разработать простейший класс.
2. Разработать иерархию классов
3. Продемонстрировать работу с исключениями
4. Разработать простое оконное приложение с двумя элементами
5. Продемонстрировать наследование классов с использованием абстрактного класса в качестве базового

Пример билета к зачету № ____

Вопрос 1: Дать определение класса и его членов. Привести пример класса со всеми возможными членами класса.

Вопрос 2: Понятие инкапсуляция, задачи и инкапсуляции..

Шкала оценивания и критерии оценки:

Критерии оценки	**Баллы обучающегося**	**Минимальное количество баллов**	**Максимальное количество баллов**
Уровень усвоения материала, предусмотренного программой		0	4
Умение выполнять задания, предусмотренные программой		0	2
Уровень знакомства с основной литературой, предусмотренной программой		0	2
Уровень знакомства с дополнительной литературой		0	2
Уровень раскрытия причинно-следственных связей		0	2
Уровень раскрытия междисциплинарных связей		0	2
Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)		0	2
Ответы на вопросы: полнота, аргументированность, убежденность, умение использовать ответы на вопросы для более полного раскрытия содержания вопроса		0	2
Деловые и волевые качества докладчика: ответственное отношение к работе, стремление к достижению высоких результатов, готовность к дискуссии, контактность		0	2
Итого баллов:		0	20

Оценка	Минимальное количество баллов	Максимальное количество баллов
Зачтено (отлично)	90	100
Зачтено (хорошо)	74	90
Зачтено (удовлетворительно)	60	74
Зачтено (неудовлетворительно)	0	60

Знания, умения и навыки обучающихся при промежуточной аттестации **в форме дифференцированного зачета** определяются оценками «зачтено (отлично)», «зачтено (хорошо)», «зачтено (удовлетворительно)», «не зачтено (неудовлетворительно)».

«Зачтено (отлично)» – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

«Зачтено (хорошо)» – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

«Зачтено (удовлетворительно)» – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

«Не зачтено (неудовлетворительно)» – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература

1. Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation (3rd Edition). — Addison-Wesley, Boston, MA, USA, 2006. — 750 с.
2. Шень А. Программирование: теоремы и задачи. — М.: МЦНМО, 2014. — 296 с.
3. Шень А., Верещагин Н. Языки и исчисления. — М.: МЦНМО, 2012. — 240 с.
4. Верещагин, Н. К. Колмогоровская сложность и алгоритмическая случайность [Электронный ресурс] / Н. К. Верещагин, В. А. Успенский, А. Шень. — Электрон. дан. — СПб: Лань, 2013. — 575 с. — Режим доступа: https://e.lanbook.com/book/56395 — Загл. с экрана.

8.2. Учебно-методическое обеспечение для самостоятельной работы обучающихся:

1. Кривцова, И. Е. Основы дискретной математики. Часть 1. Учебное пособие [Электронный ресурс] / И. Е. Кривцова, И. С. Лебедев, А. В. Настека. — Электрон. дан. — СПб: ИТМО, 2016. — 92 с. — Режим доступа: http://books.ifmo.ru/book/1869/osnovy_diskretnoy_matematiki._chast_1._uchebnoe_posobie.htm — Загл. с экрана.

8.3. Дополнительная литература

1. Вики-конспекты. — http://neerc.ifmo.ru/wiki/index.php?title=Заглавная_страница

8.4. Перечень ресурсов информационно-телекоммуникационной сети “Интернет”, необходимый для освоения дисциплины

1. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем»: https://library.asu.edu.ru
2. Корпоративный проект Ассоциации региональных библиотечных консорциумов (АРБИКОН) «Межрегиональная аналитическая роспись статей» (МАРС): http://mars.arbicon.ru
3. Единое окно доступа к образовательным ресурсам http://window.edu.ru

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для проведения лекционных занятий используется аудитория, оборудованная современной презентационной техникой (проектор, экран, ноутбук).

Для выполнения лабораторных работ используются компьютерные классы с установленным в них необходимым программным обеспечением.

При необходимости рабочая программа дисциплины может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе для обучения с применением дистанционных образовательных технологий. Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК).