

Документ подписан простой электронной подписью Информация о владельце: ФИО: Гаскаев Сергей Валерьевич Должность: Ректор	МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)	
Дата подписания: 04.06.2025 13:02:01 Уникальный программный ключ: 04c19ed8bf98f73b6cb77a486b0a8788b8322727	Рабочая программа дисциплины "Объектно-ориентированное программирование на языке Java" по направлению подготовки (специальности) 01.03.02 "Прикладная математика и информатика" направленности (профиль) Прикладная математика и искусственный интеллект ФГБОУ ВО «ЧелГУ»	стр. 1

Рабочая программа дисциплины (модуля)*
Объектно-ориентированное программирование на языке Java

Направление подготовки (специальность)

01.03.02 Прикладная математика и информатика

Направленность (профиль)

Прикладная математика и искусственный интеллект

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

очная

Год(ы) набора 2025

*Рабочая программа дисциплины (модуля) адаптирована для инклюзивного обучения инвалидов и лиц с ограниченными возможностями здоровья

Челябинск 2025 г.



Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре ОПОП
3. Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля)
4. Объем дисциплины (модуля)
5. Структура и содержание дисциплины (модуля)
6. Фонд оценочных средств
 - 6.1. Перечень видов оценочных средств
 - 6.2. Типовые контрольные задания и иные материалы для текущей аттестации
 - 6.3. Типовые контрольные вопросы и задания для промежуточной аттестации
 - 6.4. Критерии оценивания
7. Учебно-методическое и информационное обеспечение дисциплины (модуля)
 - 7.1. Рекомендуемая литература
 - 7.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"
 - 7.3. Перечень информационных технологий
8. Материально-техническое обеспечение дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Специальные условия освоения дисциплины обучающимися с инвалидностью и ограниченными возможностями здоровья



1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Преподаваемая дисциплина является средством решения прикладных задач, основой современного подхода к анализу предметной области и проектированию систем. Преподавание и изучение дисциплины следует рассматривать как важную составляющую профессиональной подготовки. Целью преподавания и изучения дисциплины является обучение студентов языку C++, методике разработки программ искусственного интеллекта с использованием технологии объектно-ориентированного программирования. Задачи дисциплины заключаются в том, чтобы студенты получили опыт разработки компьютерных программ на языке C++, могли реализовать на C++ математические алгоритмы; освоили синтаксис и стандартные библиотеки C++; научились адаптировать и использовать шаблоны объектно-ориентированного программирования для решения профессиональных задач.

ПК-2.1. Обладает знаниями о существующих типовых шаблонах проектирования программного обеспечения, о методах и средствах проектирования программного обеспечения, структур данных, баз данных.

ПК-2.2. Демонстрирует умение применять типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, применять методы и средства проектирования программного обеспечения, баз данных, структур данных.

ПК-2.3. Имеет практический опыт (навыки) применения стандартных алгоритмов при проектировании программного обеспечения, разработки алгоритмов решения задач в соответствии с поставленными условиями, использования методов и приёмов алгоритмизации поставленных задач.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Цикл (раздел) ОПОП: Б1.В.1.02

2.1 Требования к предварительной подготовке обучающегося:

Технология программирования:

Знает: основные методы и средства разработки ПО, принципы представление данных в памяти компьютера, порядок работы операторов языка программирования, основные принципы распределения ролей в командной работе.

Умеет:

выполнять разработку и отладку программ на языке Си, нести ответственность за свою работу и реализовать собственный потенциал в команде.

Имеет практический опыт: проектирования, кодирования и отладки разрабатываемого программного обеспечения, работы с различными системами программирования, с различными средами программирования.

Операционные системы:

Знает: принципы построения, назначение, структуру, функции и эволюцию операционных систем.

Умеет: проводить инсталляцию, конфигурирование и загрузку операционных систем, в том числе сетевых.

Имеет практический опыт: использования сетевых технологий для решения прикладных задач

Технологическая (проектно-технологическая) практика:

Знает: способы первичной обработки информации, этические нормы и установленные правила командной работы.

Умеет: разрабатывать математические модели, алгоритмы и компьютерные программ для предложенных задач, находить и критически анализировать информацию, необходимую для решения поставленной задачи, использовать математический аппарат в решении

профессиональных задач, критически оценить эффективность использования времени при решении поставленных задач, а также, относительно полученного результата.

Имеет практический опыт: декомпозиции поставленной

задачи, выделяя её базовые составляющие, участия в обмене информацией, знаниями и опытом в интересах выполнения командной задачи, программной реализации алгоритмов задач профессиональной деятельности, оценки личностных ресурсов по достижению целей

управления своим временем для успешного

выполнения порученной работы и саморазвития.

Дискретная математика



Математический анализ

Алгоритмы и структуры данных

Информатика

2.2 Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

Компьютерная графика

Методы машинного обучения

Основы фронтенд-разработки

Теория кодов

Фундаментальные алгоритмы на C++

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ПК-2: Способен использовать базовые алгоритмы и средства проектирования программного обеспечения

Знать:

существующие типовые шаблоны проектирования программного обеспечения, методы и средства проектирования программного обеспечения.

Уметь:

применять типовые решения, библиотеки программных модулей, шаблоны, используемые при разработке программного обеспечения; применять методы и средства проектирования программного обеспечения, применять методы и средства создания программного обеспечения

Владеть:

навыком применения стандартных алгоритмов при проектировании и создании программного обеспечения; разработки и реализации алгоритмов решения задач в соответствии с поставленными условиями; использования методов и приемов алгоритмизации поставленных задач

В результате освоения дисциплины обучающийся должен

3.1	Знать:
3.1.1	синтаксис языка объектно-ориентированного программирования Java; устройство и принципы построения объектно-ориентированных библиотек;
3.1.2	
3.2	Уметь:
3.2.1	адаптировать и использовать шаблоны объектно-ориентированного программирования для решения профессиональных задач;
3.3	Владеть:
3.3.1	применения объектных технологий разработки программных систем;

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость	4 ЗЕТ
Часов по учебному плану : 144	Виды контроля в семестрах: зачеты с оценкой 3
в том числе : аудиторные занятия : 68	
самостоятельная работа : 69,1	
: контактная работа: 74,9	
ИКР: 6,9	

5. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература
	Раздел 1. Язык C++			



1.1	Рекомендуемая литература. Эволюция языка C++. Отличия языка C++ от C. Замена для препроцессора. Ссылки. Операции new и delete. Перегрузка функций и аргументы по умолчанию. Ввод-вывод. Пространства имен. Операции преобразования. Другие улучшения. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.2	Классы. Определение класса. Спецификаторы доступа. Определение и вызов методов. Указатель this. Конструкторы. Деструктор. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.3	Специальные виды конструкторов. Специальные элементы класса (спецификаторы const и static, реализация паттерна «одиночка»). Друзья класса. Рекомендации по проектированию классов. /Лек/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.4	Перегрузка функций и операций. Правила связывания. Правила перегрузки операций. Форматы перегрузки операций. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.5	Примеры перегрузки операций. Перегрузка операции преобразования. Перегрузка new и delete. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.6	Шаблоны функций и классов. Специализация и инстанцирование шаблонов. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.7	Отношение наследования между классами. Виртуальные методы и абстрактные классы. Множественное наследование /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.8	Исключительные ситуации. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.9	STL. Общие сведения. Вспомогательные компоненты. Итераторы. Основные алгоритмы. Классы vector, string. Ассоциативные контейнеры. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.10	Поточные классы, управление форматированием, определение манипуляторов. Примеры решения задач с использованием классов STL. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.11	Выделение операций АТД. Реализация отдельных операций АТД "Последовательность" и "Матрица". Применение ссылок и аргументов по умолчанию. /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.12	Реализация класса "Очередь целых чисел". Конструкторы и деструкторы. /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.13	Реализация класса "День недели" и перегрузка операций ++, --, +, +=, ==, !=, ввода-вывода. Перегрузка операций =, + для класса "Строка" /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.14	Реализация класса "Фигура" и производных классов "Лист", "Закрашенный лист". /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.15	Шаблоны классов, исключительные ситуации. Реализация шаблонов классов "Стек" и "Матрица". /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.16	Изучение среды разработки Visual Studio для разработки программ. Использование отладчика. Структура программы на C++. Обработка массивов. Динамическое выделение памяти. Поточковый ввод-вывод. /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.17	Определение и тестирование класса с использованием динамического выделения памяти в соответствии с вариантом. /Лаб/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.18	Определение и тестирование класса с перегрузкой операций в соответствии с вариантом /Лаб/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.19	Определение и тестирование иерархии из 3 классов в соответствии с вариантом. /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
1.20	Определение и тестирование шаблона класса в соответствии с вариантом. /Лаб/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1
Раздел 2. Объектно-ориентированный подход				



2.1	Сложность программного обеспечения. Пять признаков сложной системы. Роль декомпозиции, абстракции и иерархии в разработке сложной системы. Эволюция объектной модели. Определение ОО подхода. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
2.2	Концепции ОО подхода: абстрагирование, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
2.3	Определение объекта. Состояние, поведение и идентичность объектов. Отношения между объектами. Определение класса. Отношения между классами. Выявление классов и выбор операций. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
2.4	Выделение объектов и классов (на примере типовых курсовых проектов). /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
2.5	Проектирование класса. Разбор типовых ошибок и правил. /Лаб/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
Раздел 3. Основные шаблоны проектирования				
3.1	Порождающие паттерны. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
3.2	Структурирующие паттерны. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
3.3	Паттерны поведения. Классы и ОО-библиотеки для задач искусственного интеллекта. /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1
3.4	Использование ОО-библиотек и платформ в задачах искусственного интеллекта. /Лаб/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1
Раздел 4. Иная контактная работа				
4.1	Индивидуальные консультации, текущий контроль /ИКР /ИКР/	3	6,9	Л1.1 Л1.2Л2.1
Раздел 5. Самостоятельная работа студентов				
5.1	Подготовка курсовой работы /Ср/	3	48	Л1.1 Л1.2Л2.1 Э1 Э2 Э3
5.2	Подготовка к диф.зачету /Ср/	3	5,1	Л1.1 Л1.2Л2.1 Э1 Э2 Э3
5.3	Изучение теоретического материала для выполнения лабораторных работ /Ср/	3	16	Л1.1 Л1.2Л2.1 Э1 Э2 Э3

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1. Перечень видов оценочных средств

Задания
Контроль по теме
Курсовая работа Анализ предметной области
Курсовая работа Разработка иерархии и интерфейса классов
Курсовая работа Реализация основных классов, функций
Курсовая работа Тестирование программы и/или классов, улучшение и исправление ошибок
Курсовая работа Оформление программной документации и пояснительной записки к курсовой работе
Курсовая работа Защита курсовой работы
Итоговый тест
Дифференцированный зачет

6.2. Типовые контрольные задания и иные материалы для текущей аттестации

Материалы представлены в приложении

6.3. Типовые контрольные вопросы и задания для промежуточной аттестации

1. Принципы ООП, свойства объектов, отношения между классами
2. Повторение C, новые резервированные слова, bool, inline, прототипы
3. Ссылки, const, namespace, using, ::
4. new (все формы), delete, перегрузка, bad_alloc
5. Ввод-вывод, перегрузка << >> для собственных классов



6. Классы, методы, static, const, mutable, friend
7. Конструкторы и деструкторы
8. Перегрузка функций и методов
9. Перегрузка операций
10. Динамически выделяемая память в классах
11. Простое наследование, виртуальные методы, абстрактные классы
12. Множественное наследование, виртуальные классы
13. Исключительные ситуации, стандартные исключения
14. *_cast, typeid
15. Шаблоны, специализация, инстанцирование
16. STL
17. Паттерны проектирования
18. Ошибки проектирования

Примеры задач

1. Для класса `String { char *s; }` определить операцию `<<` для вывода значения в форме `"te\xt"`
2. Для класса `String { char *s; }` определить операцию `+` для сцепления двух строк
3. Определить функцию-шаблон для поиска максимального значения в массиве произвольного типа. Функция должна возвращать максимальное значение.

6.4. Критерии оценивания

Дифференцированный зачет

Оценка выставляется по результатам текущего контроля, включающего 5 практических заданий, контрольную работу, 7 контрольных тестов СРС и итоговый тест по изучаемым темам. Студент вправе пройти контрольное мероприятие, включающего 5 вопросов и задачу по темам.

Итоговая оценка выставляется по балльной системе. Суммируются баллы, полученные за контрольные работы, домашние работы и за активную работу на занятиях, баллы, полученные на экзамене. Итоговая оценка выставляется по 100-балльной шкале (согласно процентам от общего количества баллов для каждого задания), исходя из полученной суммы баллов:

- От 0 до 49 баллов – «неудовлетворительно»
- От 50 до 69 баллов – «удовлетворительно»
- От 70 до 90 баллов – «хорошо»
- От 91 до 100 баллов – «отлично».

КМ текущего контроля, в которых он получил наименьшее количество баллов, в рамках промежуточной аттестации (дифференцированный зачет) для улучшения своего рейтинга и получить оценку по дисциплине согласно описанию ниже.

Курсовые работы

Оценка выставляется по БРС, с разделением выполнения на 5 этапов, обязательным этапом является защита работы перед комиссией.

Критерии оценивания текущего контроля

Задание 0 - максимальное количество баллов 10

Подключены необходимые заголовочные файлы и пространство имен `std` - 1 балл, иначе 0 баллов

Корректно выполнен ввод или инициализация и вывод данных - 2 балла, иначе 0 баллов

Корректно выполнено выделение и освобождение памяти для массива - 2 балла, иначе 0 баллов

Выполнена обработка данных, в программе содержатся необходимое количество циклов и проверок условий - 3 балла, иначе 0 баллов

Обработка данных выполнена без ошибок и эффективно - 2 балла, иначе 0 баллов

Задания 1-4 - максимальное количество баллов 20

Отчет содержит необходимые пункты - 2 балла, иначе 0 баллов

в `main` содержатся вызовы для всех открытых методов и других операций класса - 2 балла, иначе 0 баллов

действия приводят к исключительной ситуации не менее 2 раз, имеется обработка исключений (см. пример) - 2



балла, иначе 0 баллов

тесты исследуют все возможности класса и не требуют определения порядка действий от человека - 2 балла, иначе 0 баллов

есть результаты выполнения main - 2 балла, иначе 0 баллов

все указанные в задании операции были реализованы (возможно с ошибками) - 2 балла, иначе 0 баллов

есть комментарии в интерфейсе класса ко всем полям, методам и функциям - 2 балла, иначе 0 баллов

нет ошибок в реализации методов 4 балла, иначе 0 баллов

разделение реализации и интерфейса класса выполнено правильно (критерий: в интерфейсе могут быть только реализации методов из одного оператора) - 2 балла, иначе 0 баллов

Контроль по теме 1-7 - максимальное количество баллов 5

Контроль проводится в форме компьютерного теста перед проведением занятий по теме для проверки СРС по изучению теоретического материала.

Тест содержит 5 вопросов, каждый вопрос оценивается в 1 балл.

Перечень тем:

1. Улучшения языка С
2. Классы
3. Перегрузка функций и операций
4. Наследование
5. Шаблоны. Наследование
6. Объектно-ориентированный подход
7. Паттерны проектирования

Контрольная работа - максимальное количество баллов 20

Контрольная работа проводится на практическом занятии после прохождения темы "Перегрузка функций и операций. Шаблоны" КР содержит 4 небольших задания (от 4 до 10 строк)

на 45 минут. Правильное решение задачи оценивается в 5 баллов, оценка снижается на 1 балл за каждую ошибку

Максимальная оценка, итого 20 баллов Проверяется знание синтаксиса по

темам

* ввод-вывод в С++

* динамическое выделение памяти

* определение классов и методов

* объявление объектов и вызов методов

* перегрузка операций

* шаблоны функций

Курсовая работа

Задание выполнено вовремя - 3 балла, оценка снижается на 1 балл за каждую неделю превышения срока по неважной причине Набор классов и иерархия соответствует анализу предметной области - 2 балла, оценка снижается на 1 балл за каждую ошибку Правильно выбран способ объявления класса (class/struct) - 1 балл, оценка снижается на 1 балл за каждую ошибку Правильно указаны права доступа к элементам классам - 2 баллов, оценка снижается на 1 балл за каждую ошибку Методы и их параметры соответствуют поведению объектов - 2 балла, оценка снижается на 1 балл за каждую ошибку Поля и их типы позволяют сохранять состояние объектов между вызовами методов - 2 балла, оценка снижается на 1 балл за каждую ошибку.



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

7.1. Рекомендуемая литература

7.1.1. Основная литература

	Авторы,	Заглавие	Издательство,	Ресурс
Л1.1	Баранова И.В., Баранов С.Н., Баженова И.В., Кучунова Е.В., Толкач С.Г.	Объектно-ориентированное программирование на C++: учебник (https://znanium.com/catalog/document?id=380554)	Красноярск : Сибирский федеральный университет, 2019	ЭБС
Л1.2	Рацеев С. М.	Программирование на языке Си: учебное пособие для вузов (https://e.lanbook.com/book/351863)	Санкт- Петербург : Лань, 2023	ЭБС

7.1.2. Дополнительная литература

	Авторы,	Заглавие	Издательство,	Ресурс
Л2.1	Лямин А. В., Череповская Е. Н.	Объектно-ориентированное программирование: компьютерный практикум (https://e.lanbook.com/book/110457)	Санкт- Петербург : НИУ ИТМО, 2017	ЭБС
Л2.2	Барков И. А.	Объектно-ориентированное программирование: учебник для вузов (https://e.lanbook.com/book/329549)	Санкт- Петербург : Лань, 2023	ЭБС

7.1.3. Методические разработки

	Авторы,	Заглавие	Издательство,	Ресурс
Л3.1	Морозова Ю. В.	Практикум по объектно-ориентированному программированию: учебное пособие (https://e.lanbook.com/book/313631)	Москва : ТУСУР, 2021	ЭБС

7.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Алексеев Михаил Николаевич [Электронный ресурс] : сайт / Челяб. гос. ун-т. – Челябинск, 2011-. – Режим доступа: http://math.csu.ru/~alexeev/ , свободный http://math.csu.ru/~alexeev/			
Э2	Интернет-университет информационных технологий [Электронный ресурс] : сайт / НОУ «ИНТУИТ». – Москва, 2003-. – Режим доступа: http://www.intuit.ru/ , свободный http://www.intuit.ru/			
Э3	ALGOLIST.MANUAL.RU Алгоритмы и методы [Электронный ресурс] : сайт / И. Кантор. – [Б.м., 2000-2007]. – Режим доступа: http://algotist.ru/ , свободный http://algotist.ru/			

7.3 Перечень информационных технологий

7.3.1 Программное обеспечение

LMS Moodle

Open Project

OpenOffice

7.3.2 Профессиональные базы данных и информационно-справочные системы

Электронный каталог научной библиотеки ЧелГУ [Электронный ресурс] : база данных / Челяб. гос. ун-т. – Челябинск, 1992

eLIBRARY.RU [Электронный ресурс] : научная электронная библиотека [научной периодики на русском языке]. – Москва, [1999-]. – Доступ к полным текстам после регистрации из сети ЧелГУ. – URL: <http://elibrary.ru/defaultx.asp>.

Moodle [Электронный ресурс]: система дистанционного обучения : [база данных] / Челяб. гос. ун-т. – Челябинск, [б.г.]. – Доступ из сети ЧелГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. – URL: <http://moodle.uio.csu.ru/login/index.php>.

Научная библиотека Челябинского государственного университета [Электронный ресурс] : [сайт] / Челяб. гос. ун-т. – Челябинск, [2001-]. – Режим доступа: <http://www.lib.csu.ru/>, свободный. – Загл. с экрана.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)



Для реализации дисциплины используются учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения (мультимедийное устройство, проектор, ноутбук или стационарный компьютер).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с подключением к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета.

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

При изучении данной дисциплины используются лекции, практические, лабораторные занятия и самостоятельная работа студента. На лекциях и семинарских занятиях излагается основное содержание тем программы, рассматриваются основные методы и приёмы решения задач.

Для наиболее эффективного изучения дисциплины обучающемуся рекомендуется:

- посещать лекционные занятия, кратко и вдумчиво конспектировать материал лекции, с указанием даты проведения лекции и темы;
- посещать практические (семинарские) занятия, на которых рассматриваются основные методы и приёмы решения задач. Рекомендуется перед каждым практическим занятием выполнить домашнее задание, что позволит лучше усвоить предыдущий материал, и изучить лекционный материал по предстоящей теме;
- самостоятельно прорабатывать материал как после каждого занятия, так и по завершению темы, что позволяет связать воедино полученные сведения и составить цельную картину.

В случае применения при обучении дисциплины электронного обучения, дистанционных образовательных технологий общение обучающихся и преподавателя осуществляется в режиме реального времени (онлайн-лекции (вебинары), чаты, видео-конференции и др.) или отложенного времени (система дистанционного обучения Moodle, MS Office365, форумы, электронная почта и др.).

Большую часть времени обучающиеся самостоятельно работают с учебно-методическими материалами. Студенты имеют возможность консультироваться с преподавателем по всем вопросам, возникающим в ходе самостоятельной работы посредством электронной почты, социальных сетей и т.п.

Доступ обучающегося к учебным ресурсам в режиме отложенного времени, самостоятельной работы осуществляется через сеть Интернет в удобном для него месте, времени и темпе.

При обучении лиц с ограниченными возможностями здоровья электронное обучение, дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Реализация дисциплины с применением электронного обучения, дистанционных образовательных технологий (далее – ЭО, ДОТ) осуществляется на основании «Положения о реализации основных и дополнительных образовательных программ с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Челябинский государственный университет», «Положения о порядке зачета обучающимся по основным профессиональным образовательным программам высшего образования в ФГБОУ ВО «ЧелГУ» результатов освоения в организациях, осуществляющих образовательную деятельность, учебных предметов, курсов, дисциплин (модулей), практик, дополнительных образовательных программ» посредством электронной информационно-образовательной среды ФГБОУ ВО «ЧелГУ». В исключительных случаях (форс-мажор и т.п.) при реализации образовательной деятельности с применением ЭО, ДОТ могут применять компоненты, не входящие в перечень электронной информационно-образовательной среды.

10. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ ОБУЧАЮЩИМИСЯ С ИНВАЛИДНОСТЬЮ И ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Освоение дисциплины инвалидами и лицами с ограниченными возможностями здоровья осуществляется с использованием специальных технических средств и информационных технологий, предоставляемых Ресурсным учебно-методическим центром по обучению инвалидов и лиц с ограниченными возможностями здоровья ЧелГУ по запросу обучающегося (мобильные специальные технические средства для лиц с нарушениями зрения и с нарушением слуха, ассистивные информационные технологии).

При необходимости для обучающихся с нарушениями зрения на рабочих местах для проведения практических или лабораторных занятий устанавливается специальное программное обеспечение (программа речевой навигации, речевые синтезаторы, экранные лупы).

В учебные аудитории обеспечивается беспрепятственный доступ для обучающихся с инвалидностью и с ограниченными возможностями здоровья. В каждой аудитории, где обучаются инвалиды и лица с ограниченными возможностями здоровья, предусматривается соответствующее количество мест для обучающихся с учетом нарушений их здоровья.



Для освоения дисциплины инвалидам и лицам с ограниченными возможностями здоровья предоставляется доступ к печатным источникам, имеющимся в научной библиотеке ЧелГУ, с помощью специальных технических средств; доступ с помощью специальных технических и программных средств к электронным источникам, представленным в форме электронного документа в фонде научной библиотеки ЧелГУ или электронно-библиотечных системах.

Учебно-методические материалы для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья предоставляются в формах, адаптированных к ограничениям их здоровья и особенностям восприятия информации.

Для инвалидов и лиц с ограниченными возможностями здоровья освоение дисциплины может быть частично или полностью осуществлено с использованием дистанционных образовательных технологий.

При проведении промежуточной аттестации по дисциплине обучающимся с инвалидностью и с ограниченными возможностями здоровья обеспечивается по их заявлению предоставление в доступной форме в зависимости от их индивидуальных особенностей инструкции о порядке проведения промежуточной аттестации, оценочных средств и возможности ответов на задания (письменно на бумаге, набор ответов на компьютере, письменно шрифтом Брайля, с использованием услуг ассистента, устно).

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование предоставленных ЧелГУ или собственных технических средств, необходимых им в связи с их индивидуальными особенностями. При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на задания, процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

ООП «Прикладная математика и искусственный интеллект»
по направлению 01.03.02 – Прикладная математика и информатика
Дисциплина «Объектно-ориентированное программирование»

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
1.	Текущий контроль	Задание 0	<p>Пример задания: Объявить массив целых чисел и заполнить его случайными значениями с помощью функций <code>srand()</code> и <code>rand()</code>. Размер массива и диапазон значений его элементов вводятся с клавиатуры. Заменить все элементы с отрицательным значением на значение минимального положительного (больше 0) элемента. Если положительных элементов нет, то оставить массив без изменений. Примеры контрольных вопросов 1. Синтаксис вызова операции для динамического выделения памяти? 2. Ввод данных из потока в C++?</p>
2.	Текущий контроль	Задание 1	<p>Пример задания: Реализовать класс с заданным интерфейсом на языке C++. Множество целых от <i>a</i> до <i>b</i> Использовать массив флагов, <i>i</i>-й флаг указывает наличие числа <i>i+a</i></p> <pre>class ASet { public: ASet(int a, int b); ASet(int b); // a по умолчанию 0 ~ASet(); void incl(int); // включить число в множество // если такое число уже есть, то ничего не делать void excl(int); // исключить bool contain(int); // содержится в множестве? void print(); // распечатать состояние объекта };</pre> <p>Примеры контрольных вопросов 1. Назначение конструктора? 2. С каким спецификатором доступа определяются поля класса? 3. Какие методы можно определять в интерфейсе класса?</p>
3.	Текущий контроль	Задание 2	<p>Пример задания: Реализовать класс с заданным интерфейсом на языке C++. Вектор на плоскости (x,y) <code>Vector</code> Конструктор: <code>Vector(x,y)</code> Операции: <i>a</i>+<i>b</i>, <i>a</i>-<i>b</i>, <i>a</i>+=<i>b</i>, <i>a</i>-=<i>b</i>, <i>a</i>*<i>b</i> (скалярное произведение), <i>a</i>==<i>b</i>, <i>a</i>!=<i>b</i> !<i>a</i> (<i>a</i> является нулевым вектором), <i>a</i>/<i>z</i>, <i>z</i>*<i>a</i> ("масштабирование" вектора) где <i>a,b</i> - вектора, <i>z</i> - double вывод, ввод в виде (1.4,-2.5) Методы: <code>double getX();</code></p>

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<p>double gety();</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> 1. Для каких методов нужно указывать спецификатор const? 2. Какие операции нужно определять как friend? 3. Какие операции нужно определять как обычные функции (не friend)?
4.	Текущий контроль	Задание 3	<p>Пример задания:</p> <p>Реализовать методы класса Figure с заданным интерфейсом на языке C++.</p> <pre>class Figure { int c; // цвет bool visible; protected: int x,y; // базовая точка virtual void draw(); public: Figure(int c, int x, int y); ~Figure(); void move(int dx, int dy); // сместить фигуру на (dx,dy) // видимая фигура гасится, затем рисуется в другом месте // у невидимой просто меняются поля x,y void setcolor(int c); // установить цвет фигуры // видимая фигура рисуется новым цветом // у невидимой просто меняется поле c int getcolor(); // получить цвет void hide(); // спрятать: нарисовать черный прямоугольник // по размерам area() void show(); // показать bool isvisible(); // видима? virtual void area(int &x1,int &y1,int &x2,int &y2); // получить размеры прямоугольной области, содержащей фигуру };</pre> <p>Определить производный класс Эллипс: Ellipse(цвет линий, x и y центра, радиус1, радиус2)</p> <p>Определить дополнительный метод в производном классе для изменения размеров: void setsizes(радиус1, радиус2);</p> <p>От написанного класса произвести новый дочерний класс - закрашенная фигура. Определить дополнительный метод для изменения цвета заполнения: void setfillcolor(int c);</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> 1. Как нужно определить деструктор Figure и производных классов, чтобы видимый объект исчезал с экрана при уничтожении? 2. Какой спецификатор доступа используется по умолчанию при наследовании в классах?
5.	Текущий контроль	Задание 4	<p>Пример задания:</p> <p>Определить класс-шаблон с использованием динамического распределения памяти и необходимые конструкторы и операции, включая конструктор копий, операции присваивания и индексации. Класс вектор (одномерный массив элементов заданного типа и размера, указанного в аргументах конструктора), получение i-го элемента с помощью операции индексации.</p> <p>При выходе за границу, переполнении и т.п. вызвать исключительную ситуацию (определить собственные классы) для информирования программы, вызвавшей метод.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> 1. Синтаксис оператора перехвата исключения? 2. От какого класса STL можно наследовать для создания класса исключений?
6.	Текущий контроль	Контроль по теме 1	<p>Пример контрольного задания:</p> <ol style="list-style-type: none"> 1. Определить функцию dec, которая уменьшает на 1 свой параметр, переданный по ссылке. Функция не возвращает никакого значения. int x=2;

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
	роль		<p>Написать вызов функции dec для изменения значения x:</p> <p>_____</p> <p>2. int *p; выделить память под массив из 100 целых значений p=_____</p> <p>3. Освободить память для p из вопроса 2</p> <p>_____</p> <p>4. int x,y; Вывести значения x и y способом из C++, не используя printf</p> <p>5. int adr; Преобразовать адрес, содержащийся в p, в целое число adr=_____</p>
7.	Текущий контроль	Контроль по теме 2	<p>Пример контрольного задания:</p> <p>1. Что такое класс?</p> <p>2. Что такое селектор?</p> <p>3. В классе ABC есть метод test</p> <pre>class ABC { ... void test(); } a1, *a2;</pre> <p>Вызвать test для объекта a1: _____</p> <p>Вызвать test для указателя на объект a2: _____</p> <p>4. Напишите спецификатор для закрытого доступа: _____</p> <p>5. Напишите заголовок деструктора для класса ABC: _____</p>
8.	Текущий контроль	Контроль по теме 3	<p>Пример контрольного задания:</p> <p>1. Назначение конструктора копий</p> <p>2. Заголовок конструктора-преобразователя для класса ABC из int.</p> <p>_____</p> <p>Какой спецификатор нужно указать, чтобы конструктор нельзя было применить неявно для преобразования из int в ABC: _____</p> <p>3. В классе ABC нужна константа size, равная 100. Что нужно написать ?</p> <pre>class ABC { _____ int size=100; int a[size]; };</pre> <p>4. Заголовок для перегрузки операции префиксный ++ для класса ABC как метода</p> <p>_____</p> <p>5. В сигнатуру функции (метода) входят _____ но не входит _____</p>
9.	Текущий контроль	Контроль по теме 4	<p>Пример контрольного задания:</p> <p>1. int x=7; int main() { int x=1; _____ // напишите оператор, который выведет значение глобальной переменной x</p> <p>2. class A { public: A(int x, int y); }; class B: A { public:</p> <p>_____</p> <p>};</p> <p>Напишите определение конструктора B, который передает 2 своих параметра в конструктор A</p> <p>3. class A{ void func(); // Метод func должен быть чисто виртуальным, напишите правильное объявление };</p> <p>_____</p> <p>4. class A { public: void func(); }; class B { public: void func(); }; class C: public A, public B { }; C c1; Вызовите метод func из A для объекта c1 _____</p> <p>5. Какой спецификатор доступа нужно указать для элементов класса,</p>

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<p>Способность объекта существовать во времени и (или) в пространстве</p> <p>Способ защититься от использования объектов одного класса вместо другого, или по крайней мере способ управлять такой подменой</p> <p>Состояние системы, разложенной на внутренне связанные и слабо связанные между собой части</p> <p>3. Главными концепциями объектно-ориентированного подхода являются</p> <p>Выберите один или несколько ответов:</p> <ul style="list-style-type: none"> a. параллелизм b. абстрагирование c. модульность d. инкапсуляция (ограничение доступа) e. сохраняемость f. типизация g. иерархия <p>4. Свойство объекта, которое отличает его от всех других объектов называется</p> <p>Выберите один ответ:</p> <ul style="list-style-type: none"> a. типизация b. поведение c. инкапсуляция d. индивидуальность e. состояние f. идентичность <p>5. Отношение между классами "автомобиль" и "двигатель" является примером следующего отношения между классами:</p> <p>Выберите один ответ:</p> <ul style="list-style-type: none"> a. метаклассы b. агрегатирования (включения) c. конкретизации (инстанцирования) d. наследования e. использования
12.	Текущий контроль	Контроль по теме 7	<p>Пример контрольного задания:</p> <p>1. Как классифицируются паттерны</p> <p>Паттерны, обеспечивающие взаимодействия между классами или реакцию системы на воздействия _____</p> <p>Паттерны, определяющие связи между классами _____</p> <p>Паттерны для создания объектов _____</p> <p>2. Для создания экземпляров объектов путем простого копирования используется паттерн проектирования:</p> <p>Выберите один ответ:</p> <ul style="list-style-type: none"> a. заместитель b. абстрактная фабрика c. шаблонный метод d. адаптер e. прототип <p>3. Назначение паттерна адаптер</p> <p>Выберите один ответ:</p> <ul style="list-style-type: none"> a. Является суррогатом другого объекта и контролирует доступ к нему b. Преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты c. Предоставляет унифицированный интерфейс вместо набора интерфейсов некоторой подсистемы d. Отделить абстракцию от ее реализации так, чтобы то и другое можно было изменять независимо

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<p>е. Предоставляет способ последовательного доступа ко всем элементам составного объекта, не раскрывая его внутреннего представления</p> <pre> 4. class SelectedColors: public DrawObject { int fc, // цвет рисования bc; // цвет закрашки // конструктор получает координаты углов x1,y1,x2,y2 SelectedColors(int x1, int y1, int x2, int y2): DrawObject(x1,y1,x2,y2),fc(BLACK),bc(WHITE) {} public: static SelectedColors &instance(); // объект для текущих цветов void draw(); // нарисовать объект void set_fc(int c); // установить цвет рисования int get_fc() { return fc; } // получить цвет рисования void set_bc(int c); // установить цвет закрашки int get_bc() { return bc; } // получить цвет закрашки }; SelectedColors &SelectedColors::instance() { static SelectedColors sc(20,120,70,170); return sc; } void SelectedColors::set_fc(int c) { fc=c; draw(); } </pre> <p>Данный класс реализует паттерн проектирования Выберите один ответ: а. Адаптер б. Абстрактная фабрика в. Итератор г. Строитель д. Одиночка</p> <p>5. Какие паттерны проектирования реализованы в STL Выберите один или несколько ответов: а. наблюдатель б. заместитель в. абстрактная фабрика г. фасад д. итератор</p>
13.	Текущий контроль	Контрольная работа	<p>Пример варианта: 1. (ввод-вывод в с++, динамическое выделение памяти) Вывести N, выделить память для массива из N целых чисел, ввести N чисел, Вывести эти числа в обратном порядке через пробел, освободить память. 2. (классы и методы) Определить класс Vector с полями x, y, конструктор с соответствующими аргументами и метод length, который возвращает $\sqrt{x^2 + y^2}$ В main создать объект этого класса, указав при создании аргументы (1.5,2). Вывести результат вызова метода length. 3. (перегрузка операций) Определите в классе Vector операцию * как дружественную функцию. Операция должна вычислять векторное произведение двух аргументов. 4. (шаблоны функций) Определить функцию-шаблон print(a), где a может быть любого типа, которая выводит в cout значение a и переход на новую строку.</p>
14.	Текущий контроль	Итоговый тест	<p>Вопрос 1 Укажите главные концепции объектно-ориентированного подхода -модульность -параллелизм -инкапсуляция (ограничение доступа) -типизация</p>

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<p>-абстрагирование -иерархия -сохраняемость</p> <p>Вопрос 2 void f() { ... // 1 { ... // 2 int i; for(i=0; i<10; ++i) { ... // 3 } ... // 4 } ... // 5 }</p> <p>Укажите все области действия переменной i 3 5 4 2 1</p> <p>Вопрос 3 Какое из следующих объявлений создает константный указатель на изменяемый символ? char* const p; const char* p; &const* char p; const char& p; нет правильного варианта</p> <p>Вопрос 4 Укажите правильный вариант для перегрузки операции delete в классе A: class A { public: ... void operator delete(A *); void delete(void); void operator delete(void *p, size_t size); static void operator delete(size_t size, A *); }</p> <p>Вопрос 5 Укажите правильный заголовок для перегрузки операции вывода значения класса X: void operator <<(ostream &, const X); ostream &operator <<(X &, ostream &); friend ostream operator <<(ostream, X); ostream &operator <<(ostream &, const X &); void operator <<(ostream &, X &);</p> <p>Вопрос 6 Чтобы объявить константу MAXSIZE в классе A перед ней нужно написать class A { _____ int MAXSIZE=100; ... };</p> <p>Вопрос 7 Деструктор - это метод класса, который</p>

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<p>не может вызывать другие методы этого класса может быть автоматически создан компилятором, если не определен в классе должен быть без аргументов и возвращаемого значения должен всегда вызываться явно должен иметь имя из символа ~ и имени класса может иметь произвольное имя, но начинающееся с символа ~ не может быть виртуальным</p> <p>Вопрос 8 void f(int &); // 1 void f(double); // 2 void f(int *); // 3 f(10); В последней строке примера будет вызвана функция 2 ошибка компиляции вызвана функция 3 вызвана функция 1</p> <p>Вопрос 9 Нельзя перегружать следующие операции (укажите все правильные варианты): . :: && , -></p> <p>Вопрос 10 Если для хранения данных объекта некоторого класса динамически выделяется память, то для этого класса необходимо определить: конструктор по умолчанию конструктор копий операцию new конструктор-преобразователь операцию присваивания деструктор</p> <p>Вопрос 11 struct A { public: void f(); }; struct B : A { public: void g(); }; Метод f класса A в производном классе B является public protected private</p> <p>Вопрос 12 class A { public: A(int n); }; class B { public: B(int n); }; class C { public: C(int n); }; class D : B, A { C c1; public: D(int n) : A(n), B(0), c1(1) {} }; Укажите порядок выполнения конструкторов при выполнении конструктора D: A, B, C C, B, A</p>

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<p>B, A, C C, A, B Вопрос 13 class A { int n; public: A() { n=0; } A(int k) { n=k; } }; Укажите все правильные варианты порождения исключительной ситуации класса A: throw A(int k=10); throw new A(10); throw A; throw A(10); throw A(); throw 10;</p> <p>Вопрос 14 double a=52.74; int b; Для преобразования вещественного числа a в целое нужно использовать оператор: b=static_cast<int>(a); b=reintepret_cast<int>(a); b=dynamic_cast<int>(a); b=const_cast<int>(a); это сделать невозможно</p> <p>Вопрос 15 template void f<int *>(int *a); Это объявление является явной специализацией шаблона функции f частичной специализацией шаблона функции f определением шаблона функции f неявным инстанцированием шаблона функции f явным инстанцированием шаблона функции f</p> <p>Вопрос 16 Среди алгоритмов в STL есть меньшее из двух значений решение системы линейных уравнений решение квадратного уравнения замена элементов последовательности с заданным значением слияние двух отсортированных последовательностей</p> <p>Вопрос 17 Укажите соответствие между паттернами и их назначением: 1. Мост 2. Адаптер 3. Фасад 4. Заместитель Является суррогатом другого объекта и контролирует доступ к нему Отделить абстракцию от ее реализации так, чтобы то и другое можно было изменять независимо Преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты Предоставляет унифицированный интерфейс вместо набора интерфейсов некоторой подсистемы</p> <p>Вопрос 18 Укажите правильно спроектированный класс: class Date { int day, month, year; public: Date(int d, int m, int y); int getDay() const; int getMonth() const;</p>

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
			<pre>int getYear() const; void setDate(int d, int m, int y); ... }; class Date { int day, month, year; public: Date(int d, int m, int y); int &Day() { return day; } int &Month() { return month; } int &Year() { return year; } ... }; class Date { public: int day, month, year; Date(int d, int m, int y); ... };</pre> <p>Задача Определить операцию + для сложения двух значений класса матрица:</p> <pre>class Matrix { double **data; // массив чисел int size1,size2; // размеры public: Matrix(int,int); };</pre> <p>Функция должна возвращать новую матрицу При разных размерах матриц должна порождаться искл. ситуация bad_size</p>
15.	Промежуточный аттестация	Дифференцированный зачет	<p>Темы вопросов промежуточной аттестации:</p> <ol style="list-style-type: none"> 1. Принципы ООП, свойства объектов, отношения между классами 2. Повторение C, новые резервированные слова, bool, inline, прототипы 3. Ссылки, const, namespace, using, :: 4. new (все формы), delete, перегрузка, bad_alloc 5. Ввод-вывод, перегрузка << >> для собственных классов 6. Классы, методы, static, const, mutable, friend 7. Конструкторы и деструкторы 8. Перегрузка функций и методов 9. Перегрузка операций 10. Динамически выделяемая память в классах 11. Простое наследование, виртуальные методы, абстрактные классы 12. Множественное наследование, виртуальные классы 13. Исключительные ситуации, стандартные исключения 14. *_cast, typeid 15. Шаблоны, специализация, инстанцирование 16. STL 17. Паттерны проектирования 18. Ошибки проектирования <p>Примеры задач</p> <ol style="list-style-type: none"> 1. Для класса String { char *s; } определить операцию << для вывода значения в форме "te\xt" 2. Для класса String { char *s; } определить операцию + для сцепления двух строк 3. Определить функцию-шаблон для поиска максимального значения в массиве произвольного типа. Функция должна возвращать максимальное значение.

