

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»

СОГЛАСОВАНО

Декан ФИТ
Авдеев

А.С.

Рабочая программа дисциплины

Код и наименование дисциплины: Б1.О.2 «Программирование»

Код и наименование направления подготовки (специальности): 09.03.04
Программная инженерия

Направленность (профиль, специализация): Разработка программно-информационных систем

Статус дисциплины: обязательная часть

Форма обучения: очная

Статус	Должность	И.О. Фамилия
Разработал	доцент	Е.В. Егорова
Согласовал	Зав. кафедрой «ПМ»	Е.Г. Боровцов
	руководитель направленности (профиля) программы	Е.Г. Боровцов

г. Барнаул

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Содержание компетенции	Индикатор	Содержание индикатора
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2	Использует современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1	Применяет стандарты, нормы, правила, техническую документацию в профессиональной деятельности
		ОПК-4.2	Участствует в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1	Устанавливает программное обеспечение согласно инструкциям
		ОПК-5.2	Коммутирует аппаратное обеспечение в составе информационных и автоматизированных систем
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1	Формализует задачу и предлагает алгоритмическое решение
		ОПК-6.2	Проектирует программные продукты с применением основ информатики
		ОПК-6.3	Осуществляет разработку и тестирование программных продуктов

2. Место дисциплины в структуре образовательной программы

Дисциплины (практики), предшествующие изучению дисциплины, результаты освоения которых необходимы для освоения данной дисциплины.	
Дисциплины (практики), для которых результаты освоения данной дисциплины будут необходимы, как входные знания, умения и владения для их изучения.	Объектно-ориентированное программирование, Типы и структуры данных

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Общий объем дисциплины в з.е. /час: 18 / 648

Форма обучения	Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
очная	96	96	48	408	285

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Форма обучения: очная

Семестр: 1

Объем дисциплины в семестре з.е. /час: 6 / 216

Форма промежуточной аттестации: Экзамен

Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
32	32	16	136	95

Лекционные занятия (32ч.)

- 1. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ {лекция с разбором конкретных ситуаций} (4ч.)[1,2,3,4,8,9,11]**
 - 1.1 Основные этапы решения задач на ЭВМ. Жизненный цикл программ. Постановка задачи. Нисходящее и восходящее проектирование. Алгоритм. Способы записи алгоритма. Программа на языке высокого уровня. Тестирование и отладка программ. Интерфейс программ. Документирование
 - 1.2 Типы вычислительных процессов: линейный, разветвляющийся, циклический
 - 1.3 Основные элементы Си-программы
 - 1.4 Стандартные типы данных
 - 1.5 Основные операторы языка Си
 - 1.6 Интегрированная среда разработки Си-программ
- 2. ВВОД-ВЫВОД В СИ. ОПЕРАЦИИ В СИ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]**
 - 2.1 Основные типы данных в Си
 - 2.2 Ввод-вывод в Си. Стандартные функции ввода-вывода
 - 2.3 Операции в Си
 - 2.4 Приоритет операций. Порядок выполнения выражений. Приведение типов
 - 2.5 Решение задач с линейным алгоритмом
- 3. ОПЕРАТОРЫ РАЗВЕТВЛЕНИЯ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]**
 - 3.1 Условный оператор if. Понятие флажка. Условная операция «?:»
 - 3.2 Оператор переключения switch
 - 3.3 Решение задач с разветвляющимся алгоритмом
- 4. ЦИКЛИЧЕСКИЕ ОПЕРАТОРЫ {лекция с разбором конкретных**

- ситуаций} (4ч.)[8,9,11,18]
- 4.1 Оператор цикла с предусловием while
- 4.2 Оператор цикла с постусловием do...while
- 4.3 Оператор цикла for
- 4.4 Операторы break, continue, goto
- 5. ЗАДАЧ С ЦИКЛИЧЕСКИМ АЛГОРИТМОМ. ВЛОЖЕННЫЕ ЦИКЛЫ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
- 5.1 Частичная сумма ряда. Знакопеременные ряды
- 5.2 Бесконечное суммирование с заданной точностью
- 5.3 Печать таблиц
- 5.4 Вложенные циклы
- 5.5 Задачи на перебор
- 5.6 Примеры циклических задач
- 6. ОДНОМЕРНЫЕ МАССИВЫ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
- 6.1 Основные понятия: массив, элемент, размерность, количество элементов, базовый тип
- 6.2 Объявление массивов в Си
- 6.3 Типовые задачи по обработке одномерных массивов
- 7. ДВУМЕРНЫЕ МАССИВЫ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
- 7.1 Двумерные массивы
- 7.2 Типовые задачи по обработке двумерных массивов
- 8. УКАЗАТЕЛИ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
- 8.1 Понятие указателя
- 8.2 Адресные операции
- 8.3. Адресная арифметика
- 8.4 Динамическое распределение памяти. Динамические массивы

Практические занятия (16ч.)

1. Основные этапы разработки программ. Линейный вычислительный процесс. {работа в малых группах} (2ч.)[1,8,9,11,18] Этапы разработки программ на примере линейных программ. Структурные схемы линейных программ.
2. Ввод-вывод. {тренинг} (2ч.)[8,9,11,18] Буферизированный и небуферизированный ввод-вывод. Примеры использования стандартных функций ввода-вывода. Примеры использования спецификаций формата.
3. Разветвляющийся вычислительный процесс. {тренинг} (2ч.)[8,9,11,18] Структурные схемы разветвляющихся программ. Примеры использования флажка. Примеры решения задач с разветвляющимся алгоритмом
4. Циклический вычислительный процесс. {тренинг} (2ч.)[8,9,11,18] Структурные схемы циклических программ. Примеры решения задач с циклическим алгоритмом.
5. Алгоритм бесконечного суммирования с заданной точностью. {тренинг} (2ч.)[8,9,11,18] Частичная сумма ряда. Бесконечная сумма ряда. Организация рекуррентных вычислений. Примеры решения задач на бесконечное суммирование с заданной точностью.

6. Типовые задачи по обработке одномерных массивов. {тренинг} (2ч.)[8,9,11,18] Примеры решения задач по обработке одномерных массивов. Сортировка массивов.
7. Типовые задачи по обработке двумерных массивов. {тренинг} (2ч.)[8,9,11,18] Примеры решения задач по обработке двумерных массивов.
8. Указатели {тренинг} (2ч.)[8,9,11,18] Примеры решения задач с одномерными динамическими массивами. Примеры решения задач с двумерными динамическими массивами.

Лабораторные работы (32ч.)

1. Интегрированная среда программирования. {работа в малых группах} (4ч.)[2,3,4,5,9,18] Структура компьютерной сети университета. Использование современной среды программирования для разработки программ. Установка среды программирования и коммутация аппаратного обеспечения. Основные этапы работы в интегрированной среде.
2. Линейный вычислительный процесс. {творческое задание} (4ч.)[8,9,11,18] Разработка структурных схем в соответствии со стандартами, нормами и правилами. Разработка линейных алгоритмов и программ. Проектирование, конструирование и тестирование линейных программ. Использование стандартной библиотеки математических функций.
3. Разветвляющийся вычислительный процесс.(4ч.)[8,9,11,18] Разработка разветвляющихся алгоритмов и программ. Проектирование, конструирование и тестирование разветвляющихся программ.
4. Циклический вычислительный процесс.(4ч.)[8,9,11,18] Разработка циклических алгоритмов и программ. Проектирование, конструирование и тестирование циклических программ. Использование программы-отладчика, встроенной в среду программирования.
5. Бесконечное суммирование с заданной точностью. Печать таблиц.(4ч.)[8,9,11,18] Разработка алгоритмов и программ с использованием вложенных циклов. Проектирование, конструирование и тестирование программ, реализующих алгоритмы бесконечного суммирования с заданной точностью и вывода таблиц.
6. Одномерные массивы.(4ч.)[8,9,11,18] Разработка алгоритмов и программ, содержащих одномерные массивы. Проектирование, конструирование и тестирование программ, содержащих одномерные массивы.
7. Двумерные массивы.(4ч.)[8,9,11,18] Разработка алгоритмов и программ, содержащих двумерные массивы. Проектирование, конструирование и тестирование программ, содержащих двумерные массивы.
8. Указатели. Динамическое распределение памяти.(4ч.)[8,9,11,18] Использование указателей. Разработка алгоритмов и программ, содержащих динамические массивы. Проектирование, конструирование и тестирование программ, содержащих динамические массивы.

Самостоятельная работа (136ч.)

1. Подготовка к практическим занятиям(16ч.)[1,8,9,11,18]
2. Подготовка к защите лабораторных работ.(16ч.)[1,8,9,11,18]
3. Подготовка к лабораторным работам(32ч.)[1,2,3,4,5,8,9,11,18]
4. Подготовка к лекциям.(8ч.)[1,8,9,11,18]
5. Выполнение расчетного задания(28ч.)[7,8,9,11,18]
6. Подготовка к сдаче экзамена(36ч.)[1,8,9,11,18]

Семестр: 2

Объем дисциплины в семестре з.е. /час: 6 / 216

Форма промежуточной аттестации: Экзамен

Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
32	32	16	136	95

Лекционные занятия (32ч.)

1. СТРОКИ В СИ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
 - 1.1 Массивы и указатели
 - 1.2 Массивы, указатели и строки
 - 1.3 Массивы строк
 - 1.4 Многомерные массивы и указатели
2. СТАНДАРТНЫЕ ФУНКЦИИ ДЛЯ РАБОТЫ СО СТРОКАМИ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
 - 2.1 Функции обработки строк
 - 2.2 Типовые задачи по обработке строк
3. ПРОИЗВОДНЫЕ ТИПЫ (СТРУКТУРЫ, ОБЪЕДИНЕНИЯ) {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
 - 3.1 Структуры (записи)
 - 3.2 Объединения
 - 3.4 Переменные структуры
 - 3.5 Массивы структур
4. ПОДПРОГРАММЫ В СИ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
 - 4.1 Технология модульного программирования
 - 4.2 Классификация модулей
 - 4.3 Определение и описание функций в Си
 - 4.4 Управление видимостью функций
 - 4.5 Вызов функций. Возвращение результата
 - 4.6 Передача параметров
 - 4.7 Передача массивов в качестве параметров
 - 4.8 Связь функций из разных файлов
 - 4.9 Локальные и глобальные данные
5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ФУНКЦИЙ. КЛАССЫ ПАМЯТИ {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18]
 - 5.1 Рекурсивные функции

- 5.2 Функции с переменным числом параметров
- 5.3 Функция main() – передача параметров и возвращение результата
- 5.4 Передача имени функции в качестве параметра. Указатель на функцию
- 5.5 Классы памяти
- 6. **ФАЙЛЫ** {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18] 6.1
Файлы и потоки
- 6.2 Определение потока
- 6.3 Стандартные и не стандартные потоки
- 6.4 Текстовые и двоичные потоки
- 7. **ОСНОВНЫЕ ЭТАПЫ РАБОТЫ С ФАЙЛАМИ (С ПОТОКАМИ)** {лекция с разбором конкретных ситуаций} (4ч.)[8,9,11,18] 7.1 Определение файла
- 7.2 Открытие файла
- 7.3 Чтение и запись
- 7.4 Закрытие файла
- 7.5 Дополнительные функции работы с файлами
- 7.6 Примеры организации работы с файлами
- 8. **МЕТОДОЛОГИИ СТРУКТУРНОГО И ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ**(4ч.)[8,9,11,18] 9.1
Технология структурного программирования
- 9.2 Технология объектно-ориентированного программирования

Практические занятия (16ч.)

1. Строки(2ч.)[8,9,11,18] Примеры решения задач с символьными массивами. Примеры решения задач с использованием строк.
2. Стандартные функции обработки строк(2ч.)[8,9,11,18] Примеры решения задач с использованием стандартных функций обработки строк.
3. Массивы, структуры, объединения(2ч.)[8,9,11,18] Примеры решения задач с использованием массива обычных структур, объединений и массива переменных структур.
4. Функции(2ч.)[8,9,11,18] Примеры программ с функциями, возвращающими значение по оператору return. Примеры программ с функциями, возвращающими результат по списку параметров.
5. Функции. Организация меню(2ч.)[8,9,11,18] Модульная структура программы на примере реализации курсового проекта. Организация в программе диалогового меню.
6. Файлы(2ч.)[8,9,11,18] Примеры решения задач с использованием файлов. Разработка функций создания файла, вывода файла, обработки файла.
7. Стандартизация и документирование, связанные с профессиональной деятельностью(2ч.)[6,8,9,11,18] Применение и разработка стандартов, норм, правил, а также технической документации, связанных с профессиональной деятельностью.
8. Введение в объектно-ориентированное программирование(2ч.)[8,9,11,18] Примеры решения простейших задач с использованием классов.

Лабораторные работы (32ч.)

- 1. Строки(4ч.)[8,9,11,18]** Разработка алгоритмов и программ по обработке строк. Проектирование, конструирование и тестирование программ по обработке строк.
- 2. Массив строк(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих массивы строк (текст). Проектирование, конструирование и тестирование программ, содержащих массивы строк.
- 3. Массив структур(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих массивы структур. Проектирование, конструирование и тестирование программ, содержащих массивы структур.
- 4. Функции(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих функции. Проектирование, конструирование и тестирование программ, содержащих функции.
- 5. Организация меню. Разработка диалоговых программ(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих элементы меню и диалога. Проектирование, конструирование и тестирование программ, реализующих диалог с пользователем.
- 6. Файлы (потоки)(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих работу с файлами. Проектирование, конструирование и тестирование программ, содержащих работу с файлами.
- 7. Файлы. Творческие задачи(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих работу с файлами и имеющих творческую составляющую. Проектирование, конструирование и тестирование программ, содержащих работу с файлами и имеющих творческую составляющую.
- 8. Структурное и объектно-ориентированное программирование(4ч.)[8,9,11,18]** Разработка алгоритмов и программ, содержащих классы. Проектирование, конструирование и тестирование простейших программ, содержащих классы.

Курсовые работы (0ч.)

- . Курсовой проект(0,ч.)[6,8,9,11,18]** Используя технологию структурного программирования и язык программирования Си, разработать программный продукт для решения определенной задачи согласно заданию. Обеспечить самодокументируемость программного кода. В соответствии со стандартами, нормами и правилами, связанными с профессиональной деятельностью, оформить документацию на проведенную работу в виде отчета.

Самостоятельная работа (136ч.)

- 1. Подготовка к защите лабораторных работ(8ч.)[8,9,11,18]**
- 2. Подготовка к практическим занятиям(8ч.)[8,9,11,18]**
- 3. Подготовка к лабораторным работам(24ч.)[8,9,11,18]**

4. Подготовка к лекциям(8ч.)[8,9,11,18]
5. Выполнение курсового проекта(52ч.)[6,8,9,11,18]
6. Подготовка к сдаче экзамена(36ч.)[8,9,11,18]

Семестр: 3

Объем дисциплины в семестре з.е. /час: 6 / 216

Форма промежуточной аттестации: Экзамен

Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
32	32	16	136	95

Лекционные занятия (32ч.)

1. Система управления версиями (Version Control System (VCS) или Revision Control System) {лекция с разбором конкретных ситуаций} (1ч.)[12] Что такое контроль версий, и зачем он нужен. История развития VCS. Git и его дополнительные сервисы.
2. Абстрактный тип данных (АТД) {лекция с разбором конкретных ситуаций} (1ч.)[10,13] "Абстрактность" и "абстракция". Что такое АТД, его интерфейс и внутреннее представление. Что дает программисту использование АТД. Процедурное программирование и АТД. Структуры языка С как одна из реализаций АТД и проблемы такой реализации. Идеи объединения данных и действий над ними.
3. Введение в ООП (Объектно-Ориентированное Программирование) с примерами на С++. {лекция с разбором конкретных ситуаций} (2ч.)[10,13] Основная идея объектно-ориентированного программирования. Классы, экземпляры и объекты. Инкапсуляция и сокрытие. Классы и объекты на С++. Ассоциация, композиция и агрегация (с примерами на С++). Дерево (или граф) объектов. Язык UML. Диаграмма классов UML.
4. Язык Java и основные реализующие его технологии. {лекция с разбором конкретных ситуаций} (2ч.)[10,14] Основы языка Java (Байт код Java, JVM, JDK, JRE). Основные возможности и идеи. Классы, объекты и объектные переменные.
5. Язык С# и основные реализующие его технологии. {лекция с разбором конкретных ситуаций} (2ч.)[10,11,15] Основы языка С# (С-подобный синтаксис, NET Framework, CLR, CIL). Основные возможности и идеи. Классы (ссылочные типы) и структуры (значимые типы).
6. Создание и уничтожение объектов, конструкторы и деструкторы (с примерами на С++, С# и Java). {лекция с разбором конкретных ситуаций} (3ч.)[10,11,13,14,15] Что такое конструкторы и для чего они необходимы. Конструкторы по умолчанию. Создание конструкторов. Инициализаторы объектов. Списки инициализаторов членов. Конструкторы копий (мелкое и глубокое копирование). Конструкторы перемещения. Общий порядок работы конструктора. Деструкторы, Finalize, оператор delete и сборка мусора.

7. Особенности реализации некоторых распространенных программных конструкций на языках C++, C# и Java {лекция с разбором конкретных ситуаций} (3ч.)[10,11,13,14,15] Пространство имен и пакеты; Особенности языков C++, Java и C# при работе со строками;

Работа с одномерными и многомерными массивами, особенности работы с массивами объектов; Передача параметров и возврат результата при вызове подпрограмм, особенности передачи объектов в качестве параметров; Перегрузка функций и операторов; Оператор this, что это такое и для чего он нужен; Дружественные функции и дружественные классы; Статические поля и методы, что это такое и как работают; Перегрузка оператора присваивания; Ошибки времени выполнения, исключения;

8. Наследование {лекция с разбором конкретных ситуаций} (4ч.)[10,11,13,14,15] Что такое наследование и чем оно полезно. Базовый и производный класс. Одиночное и множественное наследование. Транзитивное наследование и иерархия наследования. Как это отображается на UML-диаграмме классов. Что не наследуется производными классами от базового. Видимость и доступность. Модификатор protected. Переопределение реализации базового класса, когда можно это делать, и когда это делать обязательно. Неявное наследование. Предотвращение дальнейшего наследования. Практические примеры на C++, C# и Java: Процесс создания базового класса и его производных классов.

9. Полиморфизм -одна из трех основных парадигм ООП {лекция с разбором конкретных ситуаций} (4ч.)[10,11,13,14,15] Что такое полиморфизм. В каких случаях объекты производного класса могут обрабатываться как объекты базового класса и зачем это нужно. Абстрактные и виртуальные методы. Абстрактные базовые классы, интерфейсы и их реализация. Виртуальный деструктор и виртуальное наследование в C++. Практические примеры на C++, C# и Java:Разработка абстрактных базовых классов и их производных классов.

10. Шаблоны (templates). Обобщенное программирование(Generics). {лекция с разбором конкретных ситуаций} (2ч.)[10,11,12,14,15] Шаблоны функций и классов (template) на C++ ; Обобщенные методы и классы (generics) на Java и C#; Делегаты на C#.

11. Примеры применения ООП. Часть 1. {лекция с разбором конкретных ситуаций} (3ч.)[10,11,13,14,15] Использование библиотеки STL на C++, контейнер, итератор, алгоритм, дополнительные контейнеры и функциональный объект. Сортировка и поиск с предикатом для контейнера с объектами базового и производного классов. Коллекции на Java и C#. Реализация интерфейсов Comparable, Comparer. Обобщенные коллекции. Чтение из файла и запись в файл объектов. Сериализация.

12. Примеры применения ООП. Часть 2 (GUI). {лекция с разбором конкретных ситуаций} (5ч.)[10,11,13,14,15,16,17] Обсуждается реализация пяти GUI приложений на разных платформах (Qt, Windows Form C++,

Windows Form C#, Java Swing, Android). В программах используются элементы ввода (строки ввода, надписи, кнопки, меню, списки, таблицы), главная и вспомогательные формы, диалоговые окна, а также работа с графикой (загрузка и вывод изображения, преобразование изображения по отдельным точкам, рисование геометрических фигур и анимация).

Практические занятия (16ч.)

1. КП. Выбор темы и постановка задачи. {разработка проекта} (2ч.)[6] Выбор, формулировка темы курсового проекта и ее согласование с преподавателем. Формирование групп разработчиков (если предполагается групповая разработка проекта). Обзор предметной области (в письменном виде). Постановка задачи (в письменном виде). Результатом работы будет письменный отчет который станет первой частью отчета по КП.

2. КП. Проектирование {разработка проекта} (6ч.)[6] Используя так называемое "нисходящее проектирование", определяется детальная структура проекта, четко специфицируются его компоненты (на естественном языке или некотором языке спецификаций), их взаимодействие между собой и пользователем, фиксируется представление данных и алгоритмы их обработки.

Проектированию подлежат Архитектура ПО; Устройство компонентов ПО; Пользовательские интерфейсы. Результатом работы является письменный отчет который станет второй главой отчета по КП.

3. КП. Реализация {разработка проекта} (6ч.)[6] Разработка двух различных реализации проекта на объектно-ориентированном языке программирования. Реализации должны отличаться друг от друга либо языком программирования, либо платформой разработки. Результатом работы будет письменный отчет который станет заключительной частью отчета по КП.

4. КП. Защита. {разработка проекта} (2ч.)[6] Демонстрация ПО и отчета по КП. Защита КП.

Лабораторные работы (32ч.)

1. Git + Github {разработка проекта} (1ч.)[12] Установка Git, первоначальная настройка, инициализация, индексация изменений, создание коммитов, игнорирование, сравнение, удаление и перемещение файлов, команда git log, отмена изменений, работа с метками, создание ветки, перемещение по ним, объединение и удаление, решение конфликтов слияния, механизм прятанья (git stash), работа с удаленным репозиторием Github.com.

2. Реализация АТД на языке С. {разработка проекта} (1ч.)[1,4,5,9,12] Придумать и кратко описать некоторую прикладную задачу для решения которой обрабатываемые объекты разумно представлять в виде структур; Выделить набор значимых свойств этих объектов и разработать соответствующую структуру языка С для их представления; Разработать

набор функций на языке C для работы с этими структурами; В main продемонстрировать работу со статическими и динамическими переменными данного типа.

3. Классы и объекты языка C++ {разработка проекта} (2ч.)[1,4,12,13] Модифицировать программу из ЛР № 2. Заменить struct на class; Функции становятся методами класса; Поля становятся private, доступ к ним при необходимости осуществляется через методы public; Дополнительно в main продемонстрировать работу с динамическим массивом объектов класса и с массивом динамических объектов класса; применение операторов new, delete, delete[]; использование (*a) и ->.

4. Ассоциация {разработка проекта} (2ч.)[1,4,12,13] Доработать описание прикладной задачи используя принцип ассоциации (должен быть как минимум один вспомогательный и один основной класс); Доработать программу из лабораторной работы №3, добавить новые, модифицировать существующие классы и функцию main для демонстрации принципа ассоциации; В отчет по ЛР добавить UML-диаграмму классов.

5. Первая программа на Java {разработка проекта} (2ч.)[12,14] Для задания из л/р 4 реализовать необходимые классы на языке Java; Продемонстрировать их использование в консольном Java-приложении; При выполнении задания разрешается использовать только JDK (не пользоваться IDE);

6. Первая программа на C# {разработка проекта} (2ч.)[12,15] Для задачи из л/р 4 реализовать необходимые классы на языке C#; Продемонстрировать их использование в консольном C# приложении; Продемонстрировать сборку проекта и запуск приложения на платформе Mono под Linux, Mac OS и пр.;

7. Реализация некоторых распространенных программных конструкций на языках C++, C# и Java {разработка проекта} (2ч.)[12,13,14,15] На C++ продемонстрировать возврат значения из метода через указатель, через ссылку; разумное использование this; разумное использование дружественных функций; перегрузку операторов '+', '++'; работу с std::string. На Java продемонстрировать работу с массивом объектов; возврат значения из метода через вспомогательный класс; разумное использование this; обработку строк. На C# добавить в класс свойства; показать работу с массивом объектов; отличие класса (class) от структуры (struct); возврат значения через параметр out и через параметр ref; разумное использование this; перегрузку операторов '+', '++'; обработку строк.

8. Статические поля и методы {разработка проекта} (2ч.)[12,13,14,15] Придумать для чего в классе может быть разумным использование статического поля и статического метода; Модифицировать программы на C++, C# и Java из предыдущей лабораторной работы добавив в один из классов статическое поле и статический метод.

9. Конструкторы {разработка проекта} (2ч.)[12,13,14,15] В проектах на C++, C# и Java в каждом классе создать конструкторы. Должны существовать конструктор со всеми параметрами, конструктор с одним параметром и

конструктор без параметров; Продемонстрировать вызов всех конструкторов. Для C++ дополнительно вызов всех конструкторов статическими и динамическими объектами; Инициализировать небольшой массив конструктором с одним параметром; Для C++ создать конструктор копии и перегрузку оператора присваивания. Продемонстрировать различие между мелким и глубоким копированием.

10. Исключения {разработка проекта} (2ч.) [12,13,14,15] В проектах на C++, C# и Java придумать разумное использование защищенных блоков (try-блоки), перехватов исключений (catch-блоки), инициализации исключений (инструкция throw) и продемонстрировать это.

11. Наследование {разработка проекта} (4ч.) [12,13,14,15] Модифицировать ваши проекты на C++, C# и Java; Придумать для чего в вашем проекте нужен производный класс и создать его (на C++, C# и Java); Продемонстрировать применение модификатора protected (на C++, C# и Java); Продемонстрировать перегрузку метода базового класса в производном классе (с вызовом метода базового класса и без такого вызова) (на C++, C# и Java); Продемонстрировать в конструкторе производного класса с параметрами вызов конструктора базового класса (на C++, C# и Java); Выполнить перегрузку оператора присваивания объекту производного класса объектов базового класса (C++); Перегрузить оператор << для C++ и метод toString для Java и C#; Придумать разумное использование виртуальных функций и создать их в вашем проекте. Вызов виртуальной функции продемонстрировать через ее вызов не виртуальной функцией базового класса и через динамические объекты базового и производного классов после присваивания указателя. Продемонстрировать изменение работы программы, если функция не виртуальная (C++). В проектах на C++, Java, C# придумать разумное использование абстрактного класса и создать его. Продемонстрировать его использование;

В проектах Java, C# придумать разумное использование интерфейсов и продемонстрировать их использование; В проектах на Java и C# любой из классов сделать допускающий клонирование. Продемонстрировать варианты с мелким и глубоким клонированием.

12. Шаблоны (templates). Обобщенное программирование (Generics). {разработка проекта} (2ч.) [12,13,14,15] В программах на C++, C# и Java придумать разумное использование и реализовать шаблон класса; в программе на C# придумать разумное применение делегата и продемонстрировать это;

13. Библиотека STL и ее аналоги в языках C# и Java {разработка проекта} (2ч.) [12,13,14,15] Для проектов на C++, C# и Java придумать разумное использование контейнеров, алгоритма сортировки и алгоритма поиска. В контейнере должны храниться объекты как базового, так и производного классов.

14. Диалоговый режим работы с формами {разработка проекта} (6ч.) [12,13,14,15,16,17] Создать GUI-интерфейс для своих приложений на C++,

C# и Java. Реализовать в 5-ти вариантах: Qt C++, Visual Studio C++, Java SE, Visual studio C#, Java Android Studio.

В программах должны быть использованы TextBox, Label, кнопки, радиокнопки, флажки, меню, списки выбора (ListWidget, listBox, JList, ListView), таблицы (QTableWidget, listView, JTable, GridView), стандартные диалоги. Приложение должно иметь несколько форм. В форме "о программе" реализовать мини-игру (простая анимация + обработка ввода с клавиатуры и мышки).

Курсовые работы (0ч.)

1. Курсовой проект {разработка проекта} (0,ч.)[6,12,13,14,15,16,17] Используя технологию объектно-ориентированного программирования, согласно выданному заданию на разработку программного обеспечения, разработать его проект и две различных реализации этого ПО. В соответствии со стандартами, нормами и правилами, связанными с профессиональной деятельностью, оформить документацию на проведенную работу в виде отчета.

Самостоятельная работа (136ч.)

1. Подготовка к лекциям и практическим занятиям(20ч.)[12,13,14,15,16,17]
2. Оформление отчетов по лабораторным работам и подготовка к их защите(20ч.)[12,13,14,15,16,17]
3. Выполнение курсового проекта(60ч.)[12,13,14,15,16,17]
4. Подготовка к сдаче экзамена(36ч.)[12,13,14,15,16,17]

5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

Для каждого обучающегося обеспечен индивидуальный неограниченный доступ к электронной информационно-образовательной среде АлтГТУ:

1. Егорова Е.В. Основные этапы решения задач на ЭВМ: Методические указания к лабораторным работам по курсу «Основы программирования» / Алт.гос.техн.ун-т им.И.И.Ползунова. – Барнаул: Изд-во АлтГТУ, 2020. – 9 с. - Прямая ссылка: <http://elib.altstu.ru/eum/download/pm/egorova-osn.pdf>

2. Егорова Е.В. ,Лукоянычев В.Г. Разработка консольных приложений в среде Code::Blocks: учебно-методическое пособие/ Алт.гос.техн.ун-т им.И.И.Ползунова. – Барнаул: Изд-во АлтГТУ, 2018. - 36 с.- Прямая ссылка: http://elib.altstu.ru/eum/download/pm/Egorova-Code_Blocks.pdf

3. Егорова Е.В., Лукоянычев В.Г. Интегрированная среда MinGW Developer Studio: Методические указания к лабораторным работам по

курсу «Основы программирования» / Алт.гос.техн.ун-т им.И.И.Ползунова. - Барнаул: Изд-во АлтГТУ, 2020. - 19 с. - Прямая ссылка: <http://elib.altstu.ru/eum/download/pm/metMinGW.pdf>

4. Егорова Е.В., Лукоянычев В.Г. Разработка консольных приложений в среде Visual C++: методические указания к лабораторным работам по курсу

«Основы программирования»/ Алт.гос.техн.ун-т им. И.И.Ползунова. - Барнаул:

Изд-во АлтГТУ, 2020.- 29 с. - Прямая ссылка: <http://elib.altstu.ru/eum/download/pm/MVC-egorova.pdf>

5. Егорова Е.В., Лукоянычев В.Г. Разработка консольных приложений в среде Visual Studio 2019: методические указания к лабораторным работам по курсу «Программирование» / Алт.гос.техн.ун-т им. И.И.Ползунова. - Барнаул: Изд-во АлтГТУ, 2020.- 19 с.- Прямая ссылка: <http://elib.altstu.ru/eum/download/pm/Egorova-VS2019.pdf>

6. Егорова Е.В., Лукоянычев В.Г. Учебно-методическое пособие по выполнению курсового проекта для студентов направления 09.03.04 «Программная инженерия» по курсу «Программирование» / Алт.гос.техн.ун-т им. И.И.Ползунова. - Барнаул: Изд-во АлтГТУ, 2020.- 25 с. - Прямая ссылка:

http://elib.altstu.ru/eum/download/pm/Kursov_pr_UMP_Egorova.pdf

7. Егорова Е.В. Учебно-методическое пособие по выполнению расчетного задания для студентов направления 09.03.04 «Программная инженерия» по курсу «Программирование»/ Алт.гос.техн.ун-т им. И.И.Ползунова. - Барнаул: Изд-во АлтГТУ, 2020.- 8 с. - Прямая ссылка: http://elib.altstu.ru/eum/download/pm/Egorova_Programming_rz_ump.pdf

6. Перечень учебной литературы

6.1. Основная литература

8. Егорова Е.В. Программирование на языке высокого уровня: учеб. пособие / Е.В.Егорова. - Барнаул: Изд-во АлтГТУ, 2020. - 165 с. - <http://elib.altstu.ru/eum/download/pm/Ci-egorova.pdf>

9. Егорова Е.В. Программирование на языке Си. Учебное пособие / Е.В.Егорова. - Барнаул: Изд-во АлтГТУ, 2020. - 184 с. - <http://elib.altstu.ru/eum/download/pm/egorova-ci12.pdf>

10. Кошелев К.Б. Объектно-ориентированное программирование с использованием языка C++/ Алт. госуд. технич. ун-т им. И.И. Ползунова.- Барнаул: 2007.- 86 с.- ил.

Прямая ссылка: <http://elib.altstu.ru/eum/download/pm/oop.pdf>

6.2. Дополнительная литература

11. Барышев, Д.Д. Курс лекций по дисциплине «Высокоуровневые методы информатики и программирования» / Д.Д. Барышев; Алт. гос. техн. ун-т им. И. И. Ползунова. – Барнаул: кафедра ИСЭ, АлтГТУ, 2010. – 571 с.–
Прямая ссылка:
http://elib.altstu.ru/eum/download/ise/Baryshev_vmi.pdf

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

12. <https://githowto.com/ru>

13. <https://docs.microsoft.com/ru-ru/cpp/?view=msvc-160>

14. <https://docs.oracle.com/javase/tutorial/>

15. <https://docs.microsoft.com/ru-ru/dotnet/csharp/tutorials/>

16. <http://www.doc.crossplatform.ru/qtcreator/>

17. <https://developer.android.com/>

18. <https://intuit.ru/studies/courses/13869/1266/info>

Программирование

8. Фонд оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации

Содержание промежуточной аттестации раскрывается в комплекте контролирующих материалов, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, которые хранятся на кафедре-разработчике РПД в печатном виде и в ЭИОС.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Для успешного освоения дисциплины используются ресурсы электронной информационно-образовательной среды, образовательные интернет-порталы, глобальная компьютерная сеть Интернет. В процессе изучения дисциплины происходит интерактивное взаимодействие обучающегося с преподавателем через личный кабинет студента.

№пп	Используемое программное обеспечение
1	LibreOffice
1	Android Studio
2	Code::Blocks
2	Windows
3	Java Runtime Environment
3	Антивирус Kaspersky
5	NetBeans IDE
6	Qt Creator Open Source

№пп	Используемое программное обеспечение
7	Visual Studio
10	7-Zip

№пп	Используемые профессиональные базы данных и информационные справочные системы
1	Ассоциация Разработчиков Программных Продуктов «Отечественный софт» Общедоступная база данных профессиональных сообществ и их членов (https://www.arppsoft.ru/)
2	Национальная электронная библиотека (НЭБ) – свободный доступ читателей к фондам российских библиотек. Содержит коллекции оцифрованных документов (как открытого доступа, так и ограниченных авторским правом), а также каталог изданий, хранящихся в библиотеках России. (http://нэб.рф/)
3	Крупнейший веб-сервис IT-проектов и их совместной разработки (https://github.com/)
4	На сайте проекта OpenNet размещается информация о Unix системах и открытых технологиях для администраторов, программистов и пользователей (http://www.opennet.ru/)
5	Программа Microsoft и интернет-ресурс, содержащий техническую информацию, новости и предстоящие события для профессионалов в сфере информационных технологий. На данный момент представляет собой сборник технической информации на русском языке для IT-специалистов (https://technet.microsoft.com/ru-ru/ https://docs.microsoft.com/ru-ru/welcome-to-docs)
6	Ресурсы Android для разработчиков (https://developer.android.com)

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Наименование специальных помещений и помещений для самостоятельной работы
учебные аудитории для проведения учебных занятий
помещения для самостоятельной работы

Материально-техническое обеспечение и организация образовательного процесса по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Положением об обучении инвалидов и лиц с ограниченными возможностями здоровья».