

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Институт информационных технологий и анализа данных
(121)»

УТВЕРЖДЕНА:

на заседании совета института ИТиАД им. Е.И. Попова
Протокол №8 от 16 февраля 2026 г.

Рабочая программа дисциплины

«ПРОГРАММИРОВАНИЕ»

Направление: 09.03.01 Информатика и вычислительная техника

Вычислительные машины, комплексы, системы и сети

Квалификация: Бакалавр

Форма обучения: заочная

Документ подписан простой
электронной подписью
Составитель программы:
Столбов Александр
Борисович
Дата подписания: 16.06.2026

Документ подписан простой
электронной подписью
Утвердил: Говорков Алексей
Сергеевич
Дата подписания: 17.06.2026

Документ подписан простой
электронной подписью
Согласовал: Аношко Алексей
Федорович
Дата подписания: 17.06.2026

Год набора – 2026

Иркутск, 2026 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Программирование» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ОПК ОС-2 Способность применять современные информационные технологии при решении задач профессиональной деятельности	ОПК ОС-2.3
ОПК ОС-8 Способность разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК ОС-8.1

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ОПК ОС-2.3	Умеет составлять алгоритмы программ и реализовывать их на языках программирования в среде разработки	Знать преимущества и особенности программирования на языке высокого уровня; этапы решения задачи с помощью программы; основные понятия, конструкции и структуры алгоритмического языка программирования; назначение, устройство и свойства основных абстрактных структур данных: список, очередь, стэк, дерево, хеш-таблица; эффективные алгоритмы поиска и сортировки; принципы создания пользовательских структур данных; принципы и методы функциональной декомпозиции и разработки программ модульной структуры; состав и возможности современных алгоритмических программных библиотек. Уметь применять при решении алгоритмических задач типичные алгоритмы и структуры данных; создавать пользовательские структуры данных; разрабатывать схему иерархии, алгоритмы и программные коды программы модульной структуры; выбирать и использовать различные пакеты и библиотеки языка программирования Владеть навыками написания и отладки программ на высокоуровневом языке

		программирования в интегрированной среде разработки; навыками анализа программных пакетов и библиотеки для разработки программного обеспечения.
ОПК ОС-8.1	Способен осваивать и разрабатывать программное обеспечение общего назначения для решения практических задач. Способен осваивать одну из сред программирования для решения практических задач на языке программирования (использование развилки, циклов)	Знать преимущества и особенности программирования на языке высокого уровня; этапы решения задачи с помощью программы; основные понятия, конструкции и структуры алгоритмического языка программирования. Уметь составлять алгоритмы, писать и отлаживать коды на языке программирования высокого уровня, тестировать работоспособность программы; использовать современные среды разработки; формлять полученные рабочие результаты в форме отчёта. Владеть инструментальными средствами, методами и навыками разработки программного обеспечения с использованием языка программирования высокого уровня.

2 Место дисциплины в структуре ООП

Изучение дисциплины «Программирование» базируется на результатах освоения следующих дисциплин/практик: «Математика», «Дискретная математика», «Информатика»

Дисциплина является предшествующей для дисциплин/практик: «Объектно-ориентированное программирование», «Вычислительная математика», «Инженерная графика»

3 Объем дисциплины

Объем дисциплины составляет – 9 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Учебный год № 2
Общая трудоемкость дисциплины	324	324
Аудиторные занятия, в том числе:	32	32
лекции	12	12
лабораторные работы	20	20
практические/семинарские занятия	0	0
Самостоятельная работа (в т.ч.	283	283

курсовое проектирование)		
Трудоемкость промежуточной аттестации	9	9
Вид промежуточной аттестации (итогового контроля по дисциплине)	Экзамен	Экзамен

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Учебный год № 2

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Введение в программировани е	1	1	1	2			1, 2, 4, 5, 6	37	Отчет по лаборатор ной работе
2	Основы структурного программировани я	2	1	2	4			1, 2, 4, 5	36	Отчет по лаборатор ной работе
3	Основные типы данных и пользовательские структуры	3	1	3	3			1, 2, 4, 5	31	Отчет по лаборатор ной работе
4	Модульное программировани е	4	1	4	2			1, 2, 4, 5	26	Отчет по лаборатор ной работе
5	Абстрактные структуры данных	5	2	5	3			1, 2, 4, 5	31	Отчет по лаборатор ной работе
6	Библиотеки программировани я	6	2	6	2			1, 2, 4, 5	26	Отчет по лаборатор ной работе
7	Классические алгоритмы	7	2	7	4			1, 2, 4, 5, 6	66	Отчет по лаборатор ной работе
8	Идиомы программировани я	8	2					3	30	Контрольн ая работа
	Промежуточная аттестация								9	Экзамен
	Всего		12		20				292	

4.2 Краткое содержание разделов и тем занятий

Учебный год № 2

№	Тема	Краткое содержание
1	Введение в программирование	Основные определения, архитектуры ЭВМ, ЯНУ и ЯВУ, трансляция программ, консольное приложение, инструменты разработчика. Обзор этапов разработки программ. Алгоритм и его

		свойства. Варианты описания алгоритма. Структура программы. Введение в отладку и тестирование. Рекомендации по прохождению электронного курса и MOOK.
2	Основы структурного программирования	Этапы разработки программ. Структурное программирование. Блок-схема алгоритма. Словесное описание алгоритма. Алфавит, лексемы. Переменная: объявление и инициализация. Типы данных. Константы. Структура программы: выражения, вызов функций, операторы и операции.
3	Основные типы данных и пользовательские структуры	Фундаментальные типы данных и их представления в памяти. Особенности операций со знаковыми и беззнаковыми целыми. Стандарт IEEE 754. Кодирование символов. Кодировки Unicode. Функции для работы со строками. Преобразование типов. Обработка исключений. Стандартные потоки ввода-вывода. Строковые потоки. Виды памяти. Динамическая память. Стек. Указатели. Ссылки. Массивы и указатели. Динамические массивы и память.
4	Модульное программирование	Функциональная декомпозиция. Функции в C плюс плюс. Принцип передачи параметров. Кадр стека функции. Рекурсия. Область действия и область видимости переменных. Передача параметров функций по указателю и ссылке. Массивы и функции. Директивы препроцессора. Сборка программы. Особенности глобальных переменных.
5	Абстрактные структуры данных	Пользовательские типы данных C плюс плюс: структура, перечисление, объединение. Функции в структурах. Указатели на функцию. Перегрузка функций. Перегрузка операций. Разбиение библиотеки на модули. Стек, очередь, списки, деревья, хэш-таблицы. Управление связанными структурами данных. О вычислительной сложности алгоритмов.
6	Библиотеки программирования	Основы обобщенного программирования с использованием шаблонов C плюс плюс. Стандартная библиотека шаблонов STL: контейнеры, итераторы, алгоритмы.
7	Классические алгоритмы	Алгоритмы сортировки и поиска, алгоритмы поиска и сортировки с использованием бинарного дерева, алгоритмы на графах, динамичное программирование и другие алгоритмы.
8	Идиомы программирования	Парадигмы программирования: структурное, функциональное, объектно-ориентированное. Основные типы языков программирования. Тенденции развития средств и систем для проектирования программ. Проектирование структур и методов на основе компонентного и

		каркасного подхода. Введение в тестирование программ. Формирование спецификации библиотеки. Машина Тьюринга. Основные приёмы структурного программирования.
--	--	---

4.3 Перечень лабораторных работ

Учебный год № 2

№	Наименование лабораторной работы	Кол-во академических часов
1	Этапы разработки программ	2
2	Основы структурного программирования	4
3	Основные типы данных и пользовательские структуры	3
4	Модульное программирование	2
5	Абстрактные структуры данных	3
6	Введение в библиотеки программирования	2
7	Классические алгоритмы	4

4.4 Перечень практических занятий

Практических занятий не предусмотрено

4.5 Самостоятельная работа

Учебный год № 2

№	Вид СРС	Кол-во академических часов
1	Выполнение тренировочных и обучающих тестов в дистанционном режиме	7
2	Оформление отчетов по лабораторным и практическим работам	28
3	Подготовка к контрольным работам	30
4	Подготовка к практическим занятиям (лабораторным работам)	161
5	Подготовка к сдаче и защите отчетов	7
6	Прохождение массового открытого онлайн-курса	50

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: В ходе проведения лекций, практических и лабораторных работ используются следующие интерактивные методы обучения: интерактивная демонстрация способа решения типовой проблемы; работа в команде (групповая разработка алгоритмов и их сравнение по различным критериям); взаимная проверка и тестирование алгоритмов; групповое обсуждение эффективности алгоритмов; проектный метод, метод «мозгового штурма» (постановка проблемы, генерация идей, оценка, отбор идеи и реализация алгоритма).

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по лабораторным работам:

Электронный курс по дисциплине «Программирование (09.03.01) для заочной формы обучения для наборов с 2021 года». Электрон. версия. URL: <https://el.istu.edu/course/view.php?id=7407> (дата обращения: 20.02.2025).

5.1.2 Методические указания для обучающихся по самостоятельной работе:

Электронный курс по дисциплине «Программирование (09.03.01) для заочной формы обучения для наборов с 2021 года». Электрон. версия. URL: <https://el.istu.edu/course/view.php?id=7407> (дата обращения: 20.02.2025).

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 учебный год 2 | Отчет по лабораторной работе

Описание процедуры.

В ходе выполнения лабораторной работы студент должен применить теоретические знания по темам, изложенным в курсе «Программирование», а также приобрести практические навыки проектирования, алгоритмизации и кодирования типовых программ.

Для успешной сдачи лабораторной работы студенту необходимо предоставить и при необходимости защитить подготовленный отчет. В ходе защиты отчета студенту необходимо дать краткое изложение основных результатов, полученных в ходе выполнения лабораторной работы, устно ответить на теоретические вопросы по теме лабораторной работы, а также продемонстрировать умение ориентироваться в написанном программном коде.

Контрольные вопросы к лабораторным работам.

Лабораторная работа №1

1. Какие целые типы данных Вы знаете?
2. Какие вещественные типы данных знаете?
3. Для чего служит тип Boolean?
4. Может ли в условном блоке отсутствовать одна из ветвей?
5. Как объединить несколько операторов в условном блоке?
6. Когда используется оператор выбора?

Лабораторная работа №2

7. Сколько операторов повторения вы знаете? Перечислите их.
8. Оператор For - общий вид, реализуемый алгоритм, примеры.
9. Оператор Repeat - общий вид, реализуемый алгоритм, примеры.
10. Оператор While - общий вид, реализуемый алгоритм, примеры.
11. Чем цикл с предусловием отличается от цикла с постусловием? Почему таких циклов два вида?
12. Что такое поток ввода/вывода?
13. Особенности стандартных потоков ввода/вывода?
14. Что такое строковый поток?
15. Как представить вещественное число в научном формате, как вывести это формат в поток вывода.

Лабораторная работа №3

1. Что такое массив? Примеры массивов.
2. Основные понятия: массив, элемент массива, индекс элемента, количество элементов.
3. Размерность массива.
4. Описание массивов на Паскале.
5. В каком порядке размещаются элементы массивов в памяти?
6. Каким может быть тип элементов массива?
7. Что такое тип индекса и каким он может быть?
8. Сколько байт занимает литерная переменная?
9. Как определить код ASCII литерной переменной?
10. Какая функция служит для преобразования строчной буквы в прописную?
11. Как получить следующий/предыдущий символ в таблице ASCII?
12. Два способа представления текста в C++.
13. Тип данных string: описание переменных, атрибут длины, ввод, вывод и операция присваивания. Примеры.
14. Сравнение строк.
15. Стандартные процедуры перевода строки в число и числа в строку.
16. Чем отличаются статические и динамические величины?
17. Какая память называется динамически распределяемой?
18. Что такое указатель?
19. Какие виды указателей вам известны?
20. Как определяется адрес переменных?
21. Что такое "разыменование"?
22. Приведите примеры объявления указателей.
23. Как выделить память под динамическую переменную? Как освободить память от динамической переменной?
24. В каком случае возможно присваивание указателей?
25. Какие ситуации приводят к возникновению в динамически распределяемой памяти "мусора"?

Лабораторная работа №4

1. Какова структура описания функции?
2. В чем состоит отличие процедуры от функции?
3. Что такое область действия идентификаторов?
4. Каковы основные правила определения области действия для идентификаторов функций?
5. Какие параметры называются формальными и какие – фактическими?
6. По каким признакам различаются параметры?
7. Какие способы передачи параметров реализованы в C++?
8. Каковы правила передачи параметров-значений?
9. Каковы правила передачи параметров-ссылок?
10. В чем особенности бестиповых параметров?
11. Какое определение называется рекурсивным? Приведите собственные примеры рекурсивных определений.
12. Какой вспомогательный алгоритм (подпрограмма) называется рекурсивными?
13. Что такое граничное условие, и каково его назначение в рекурсивной подпрограмме?
14. Что такое рекурсивный спуск?
15. Что такое рекурсивный подъём?
16. Что называется текущим уровнем рекурсии?
17. Что такое глубина рекурсии? Чему равна глубина рекурсии в приведённых выше

примерах?

18. На каком этапе выполнения рекурсивной подпрограммы могут выполняться её операторы?

19. В чем преимущества и недостатки рекурсивных процедур по сравнению с итеративными?

20. Что такое стек вызова функций?

Лабораторная работа №5

1. Что такое абстрактные структуры данных?

2. Что такое динамические структуры данных?

3. Что понимают под "связанным списком"?

4. Как классифицируют связанные списки?

5. Какие основные действия над списками и компонентами списков обычно реализуют?

6. Как описывается список?

7. Что понимают под "двоичным деревом"?

8. Как классифицируют двоичные деревья?

9. Какие основные действия над двоичным деревом поиска обычно реализуют?

10. Как описывается двоичное дерево поиска?

11. В чем особенность сбалансированных двоичных деревьев?

Лабораторная работа №6

1. Что такое очередь?

2. Что такое стек?

3. Что такое хеш-таблица?

4. Что такое множество?

5. В чем заключается идея обобщенного программирования?

6. Что такое контейнер и итератор?

7. Какие абстрактные структуры данных стандартной библиотеки C++ Вы знаете?

Лабораторная работа №7

1. Дайте определение терминам «алгоритм» и «алгоритмический процесс».

2. Дайте характеристику понятия «сложность алгоритма».

3. Поясните смысл оценки сложности алгоритмов в терминах O – нотации.

4. Оцените сложность разработанного алгоритма.

Критерии оценивания.

Отлично. 1. Работа выполнена полностью.

2. Работа оформлена в соответствии с требованиями.

3. Тестовые наборы являются полными.

4. Программа имеет пользовательский интерфейс с проверкой корректности вводимых данных.

5. Программа работоспособна

6. В случае рекомендации правки программы, она выполнена.

7. Даны ответы на контрольные вопросы по теме работы

Хорошо. 1. Работа выполнена полностью.

2. Работа оформлена в соответствии с требованиями.

3. Тестовые наборы не являются полными.

4. Обучающийся иногда путается в ответах на вопросы.

Удовлетворительно. 1. Работа выполнена, но рассматривает частный случай.

2. Отчет оформлен с недостатками.

3. Программа имеет небрежно оформленный интерфейс.
4. Программа не проходит тестовые наборы преподавателя.
5. Обучающийся путается в ответах.

Неудовлетворительно.

1. Работа не выполнена.
2. Не даны ответы на контрольные вопросы.

6.1.2 учебный год 2 | Контрольная работа

Описание процедуры.

Описание процедуры:

1. Выбрать и согласовать с преподавателем программную библиотеку для анализа.
 - Язык любой из предложенных/согласованных с преподавателем
 - Возможно выбрать не библиотеку, а фреймворк
2. Провести семантическое аннотирование кода библиотеки
3. Провести классификацию библиотеки
4. Результаты описать с использованием текстового редактора

Требования к контрольной работе:

1. Описание библиотеки.
 - Описание в свободной форме основана на информации с официального сайта, документации, комментариях пользователей и т.п.
 - Структурированное описание на основе специальной информационной модели
2. Описание темы.
 - Описание темы посвящается более подробному описанию тематики библиотеки. Здесь дополнительно может использоваться общая информация из учебников и руководств, напрямую не связанных с изучаемой библиотекой.
3. Описание процесса установки библиотеки с иллюстрациями
4. Таблицы семантического аннотирования (Для каждого файла - своя таблица). При необходимости возможно изменить размер и ориентацию листа.

Предварительный вариант заголовка таблицы

- №
- Описание идиомы/приёма: название/идентификатор + URL (если нет в списке)
- Комментарий о назначении блока кода (при необходимости)
- Номера строк кода, номера позиции в строке при необходимости в скобках.

Разделители: точка с запятой, дефис. Примеры:

- o 13
- o 13;15-19
- o 13;15-19; 23(7-17)
- 5. Код в отдельном файле

Критерии оценивания.

- Отлично. 1. Работа выполнена (12 и более идиом).
2. Количество типов идиом три и более.
 3. Выбрана прикладная или системная библиотека.
 4. Работа оформлена в соответствии с требованиями.
 5. Обучающийся верно отвечает на контрольные вопросы.

Хорошо. 1. Работа выполнена (10 и более идиом).
 2. Количество типов идиом три и более.
 3. Выбрана прикладная или системная библиотека.
 4. Работа оформлена в соответствии с требованиями.
 5. Обучающийся иногда путается в ответах.

Удовлетворительно. 1. Работа выполнена (8 и более идиом).
 2. Количество типов идиом не более двух.
 3. Выбрана прикладная или системная библиотека.
 4. Отчет оформлен с недостатками.
 5. Обучающийся путается в ответах.

Неудовлетворительно. 1. Работа не выполнена.
 2. Обнаружено менее 8 идиом.
 3. Рассмотрена не библиотека, а пример использования библиотеки.
 4. Рассмотрена не прикладная или системная библиотека.
 5. Количество типов идиом не более одного

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ОПК ОС-2.3	2. Неудовлетворительно (уровень не сформирован). Студентом задание не решено. 3. Удовлетворительно (пороговый уровень). Студентом задание решено с подсказками преподавателя. При этом задание понято правильно, в логическом рассуждении нет существенных ошибок, но задание решено не полностью, или в общем виде, или не верно оформлен отчет. 4. Хорошо (базовый уровень). Студентом задание решено с подсказкой преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен	Выполнение заданий и контрольной работы, оформление отчёта по лабораторной работе, собеседование по теоретическим вопросам.

	верный ответ. Верно оформлен отчет. 5. Отлично (повышенный уровень). Студентом задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Верно оформлен отчет. Выполнено дополнительное задание.	
ОПК ОС-8.1	2. Неудовлетворительно (уровень не сформирован). Студентом задание не решено. 3. Удовлетворительно (пороговый уровень). Студентом задание решено с подсказками преподавателя. При этом задание понято правильно, в логическом рассуждении нет существенных ошибок, но задание решено не полностью, или в общем виде, или не верно оформлен отчет. 4. Хорошо (базовый уровень). Студентом задание решено с подсказкой преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ. Верно оформлен отчет. 5. Отлично (повышенный уровень). Студентом задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Верно оформлен отчет. Выполнено дополнительное задание.	Выполнение заданий и контрольной работы, оформление отчёта по лабораторной работе, собеседование по теоретическим вопросам.

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Учебный год 2, Типовые оценочные средства для проведения экзамена по дисциплине

6.2.2.1.1 Описание процедуры

Экзамен ставятся по результатам выполнения лабораторных работ (допуск к экзамену – выполнение всех лабораторных работ минимум на оценку «удовлетворительно»), устных ответов на теоретические вопросы из экзаменационного билета и решения практической задачи (выбирается на основе заданий лабораторных работ). Обязательным условием для допуска к экзамену является выполнение МООК и контрольной работы. В ходе экзамена студент должен устно ответить на вопросы билета, а также быть способен составить программный код для решения практического задания. Допускается письменный ответ на вопросы билета (по решению преподавателя). Практическое задание формируется из заданий лабораторных работ с учётом тем теоретических вопросов.

Обучающийся выполняет МООК по всем изучаемым темам.

Для МООК зачёт:

1. Выполнены все теоретические задания МООК по теме.
2. Выполнено не менее 50% практических задач по теме.

Пример задания:

Список теоретических вопросов

1. Понятия системы программирования, языка программирования, программы.
2. Архитектуры ЭВМ. Адресное пространство.
3. Алгоритм. Определение и свойства.
4. Способы описания алгоритмов. Блок-схемы: основные элементы и принципы организации.
5. Уровни языков программирования. Трансляция программ. Базовые принципы.
6. Структурное программирование. Виды вычислительных процессов. Утверждение структурного программирования.
7. Назначение таблицы внешних спецификаций.
8. Назначение тестирования и отладки программы. Таблица тестов.
9. Структура программы C++. Идентификаторы в C++.
10. Описание переменных в программе. Фундаментальные типы данных C++.
11. Виды констант программы.
12. Выражения. Приоритет операций.
13. Арифметические операции, операции отношения.
14. Инкремент и декремент. Особенности постфиксной и префиксной записи.
15. Логические и побитовые операции.
16. Операция запятая, размер, тернарная операция.
17. Операторы следования.
18. Операторы перехода (передачи управления).
19. Условный оператор.
20. Оператор выбора (switch).
21. Операторы цикла, их особенности и различия
22. Строки. Основные функции для работы со строками
23. Потоки ввода-вывода. Базовые принципы. Текстовые и двоичные потоки.
24. Стандартные потоки ввода-вывода. Извлечение из потока и включение в поток.
25. Строковые потоки C++.
26. Манипуляторы потоков. Примеры C++.
27. Обработка исключений
28. Статические массивы: одномерные и многомерные
29. Динамические массивы: одномерные
30. Динамические массивы: многомерные
31. Отличия статических и динамических массивов
32. Метод сортировки вставкой
33. Метод сортировки пузырьком

34. Метод сортировки выбором
35. Двоичный (бинарный) поиск
36. Препроцессор, примеры директив препроцессора
37. Алгоритм сборки программы, примеры на C++
38. Область действия переменной
39. Область видимости переменной
40. Глобальные переменные: объявление, обращения, использование в нескольких модулях.
41. Статическая память.
42. Автоматическая память.
43. Динамическая память.
44. Указатели. Определение, размер, особенности (нулевые, void*)
45. Указатели. 4 базовые операции.
46. Указатели. Арифметика указателей.
47. Распределение памяти. Особенности new и delete.
48. Сборка мусора. Базовые принципы.
49. Функциональная декомпозиция программы. Схема иерархии функций программы. Таблицы спецификаций функций.
50. Объявление и определение функции C++.
51. Параметры функции.
52. Вызов функции. Возврат значений из функции.
53. Кадр стека функции.
54. Передача параметров в функцию с использованием указателей.
55. Ссылки. Передача параметров в функцию с использованием ссылок.
56. Способы передачи массивов в функцию.
57. Рекурсия.
58. Разбиение программы на модули, примеры на C++.
59. Фундаментальные типы композиции объектов: агрегирование, конкатенация, делегирование.
60. Пользовательские типы данных C++: перечисление, объединение, псевдонимы.
61. Пользовательские типы данных C++: структура. Объявление, инициализация, создание объектов (к понятию о фабричной функции).
62. Пользовательские типы данных C++: структура и методы. Ключевое слово this.
63. Указатели на функцию и делегирование (на примере C++ или C#)
64. Перегрузка функций (на примере C++ или C#).
65. Перегрузка операций (на примере C++ или C#).
66. Сборка программы из нескольких файлов (cpp, h)
67. Потоки и работа с файлами (на примере C++)
68. Представление в ЭВМ целых чисел без знака: коды, операции, особенности использования.
69. Представление в ЭВМ целых чисел со знаком: коды, операции, особенности использования.
70. Представление в ЭВМ действительных чисел: формы представления, стандарт IEEE 754 (формулировка, перевод числа, особенности операций).
71. Абстрактные структуры данных: определения, примеры.
72. Абстрактная структура данных – список: структура и классификация.
73. Односвязный линейный список: алгоритм основных операций.
74. Стек и очередь: основные операции и реализация на C++
75. Бинарное дерево поиска: определение, операции поиска, вставки, удаления.
76. Бинарное дерево поиска: обход в глубину и ширину.
77. Сбалансированное дерево поиска: определение и примеры.

78. Абстрактная структура данных – хеш-таблица: определение, структура, операции
79. Абстрактные структуры данных: ассоциативный массив (отображение, карта, словарь), множество, мультимножество
80. Основные принципы обобщенного программирования
81. Основные типы контейнеров библиотеки STL, примеры использования
82. Итераторы на примере библиотеки STL
83. Принципы работы машина Тьюринга
84. Основные тенденции развития языков программирования
85. Особенности разработки приложений с графическим пользовательским интерфейсом
86. Основы оконного представления и базовые элементы управления (на примере приложений для ОС Windows)
87. Идиомы переменных и выражений в структурном программировании.
88. Идиомы арифметических и логических операций.
89. Идиомы для работы со строками.
90. Идиомы для работы с массивами фиксированной длины.
91. Идиомы для работы с массивами переменной длины.
92. Идиомы для работы с картами и словарями.
93. Идиомы для работы с функциями в структурном программировании.
94. Идиомы для работы с потоками и файлами.
95. Идиомы для работы с памятью.
96. Идиомы для работы с сетью.
97. Идиомы модульного тестирования.
98. Идиомы пользовательских типов.
99. Идиомы пространства имён.
100. Идиомы исключений.
101. Идиомы контроля исполнения.

Практическое задание (примеры)

1. Пусть сумма элементов обеих диагоналей матрицы С равна S1, а сумма элементов обеих диагоналей матрицы Т равна S2. Если $S1 < S2$, то найти матрицу $H=A+S1T$, иначе матрицу $Z=B+S2(C+T)$. Здесь А- матрица порядка n, все элементы которой равны S1; В-матрица порядка n, все элементы которой равны S2.
2. Создать контейнеры STL vector добавить в него элементы – экземпляры созданной структуры. Написать собственный код сортировки элементов этого вектора любым методом (кроме метода пузырька). Реализовать его явно, а не использовать библиотеки.
3. Создать контейнер STL set и добавить в него элементы – экземпляры созданной структуры. Использовать компараторы.
4. Выполнить действие с контейнером (set или vector), используя соответствующие алгоритмы библиотеки STL.

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Демонстрирует навыками работы в среде программирования.	Демонстрирует навыки работы в среде программирования, но при этом в	Нечетко отвечает на теоретические вопросы, допускает ошибки при ответах. Демонстрирует	Не может дать правильных ответов на поставленные теоретические вопросы. Допускает

Решает поставленные задачи, используя навыки алгоритмизации. Четко и ясно аргументирует использование самостоятельно приобретённых знаний и умений при решении задач. Способен доказать корректность алгоритма, выполнить тестирование и отладку.	ответах допускает неточности. Владеет теоретическим материалом. Решает поставленные задачи с незначительными неточностями. Способен доказать корректность алгоритма, выполнить тестирование и отладку.	навыки работы в среде программирования. Решает поставленные задачи, может допускать ошибки, сомневается в принятом решении. Подбирает неполный набор тестов для проверки корректности решения.	серьезные ошибки при решении задач. Неверно выбирает способ решения задачи, технологии программирования
--	---	---	--

7 Основная учебная литература

1. Немцова Т. И. Программирование на языке высокого уровня : программирование на языке C++ : учебное пособие для студентов среднего специального образования / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев ; под ред. Л. Г. Гагариной , 2012. - 512.
2. Сосинская С. С. Программирование на языке C++ : учебное пособие / С. С. Сосинская, 2002. - 115.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-2281.pdf>

3. Егорова Н. Н. Лабораторный практикум по курсу "Программирование на языке высокого уровня" : учебное пособие / Н. Н. Егорова, А. С. Дорофеев, 2003. - 71.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files/er-2354.pdf>

4. Павловская Татьяна Александровна. C/C++. Программирование на языке высокого уровня : учеб. для вузов по направлению "Информатика и вычисл. техника" / Т. А. Павловская, 2004. - 460.

8 Дополнительная учебная литература и справочная

1. Павловская Т. А. C/C++. Программирование на языке высокого уровня : учеб. для вузов по направлению "Информатика и вычислит. техника" / Т. А. Павловская, 2006. - 460.

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

12 Материально-техническое обеспечение дисциплины