

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 НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»

Направление: 09.03.02 Информационные системы и технологии

Информационные системы и технологии в административном управлении

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой
электронной подписью
Составитель программы:
Бахвалова Зинаида Андреевна
Дата подписания: 27.05.2026

Документ подписан простой
электронной подписью
Утвердил: Говорков Алексей
Сергеевич
Дата подписания: 10.06.2026

Документ подписан простой
электронной подписью
Согласовал: Аршинский
Вадим Леонидович
Дата подписания: 01.06.2026

Год набора – 2026

Иркутск, 2026 г.

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

1.1 Дисциплина «Программирование на языке высокого уровня» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ОПК ОС-6 Способность разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ОПК ОС-6.1, ОПК ОС-6.2

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ОПК ОС-6.1	Способен составлять и реализовывать алгоритмы с использованием языка программирования высокого уровня	Знать основные этапы создания программных продуктов; понятия «трансляция», «компиляция», «интерпретация», их различие; понятия тестирование и отладка; понятие алгоритма, формы записи алгоритма, свойства алгоритма, основные конструкции структур управления: следование, ветвление, цикл; синтаксис языков программирования высокого уровня; основные элементы систем программирования и их назначение. Уметь выполнять алгоритм, заданный в виде блок-схемы; реализовывать на языке программирования алгоритм с ветвлением, циклом; разрабатывать небольшие программы с использованием функций; составлять и редактировать код программы; формировать тесты; выполнять отладку программы в интегрированных средах программирования; Владеть навыком разработки, тестирования и анализа программ на языке высокого уровня.
ОПК ОС-6.2	Способен выполнять алгоритмическую декомпозицию задачи для обработки данных структурированных типов с использованием языка программирования высокого	Знать основные алгоритмические структуры; роль декомпозиции в разработке программы; основные виды алгоритмической декомпозиции и особенности использования; основные стратегии тестирования.

	уровня и выполнять отладку и тестирование разработанных подпрограмм	Уметь выполнять алгоритмическую декомпозицию задачи; разрабатывать тесты; выполнять тестирование программы; правильно оформлять результаты тестов. Владеть навыками применения алгоритмической декомпозиции для обработки данных структурированных типов на практике; Владеть навыками работы в средах программирования с использованием языка высокого уровня с++ (составление, отладка и тестирование программ).
--	---	---

2 Место дисциплины в структуре ООП

Изучение дисциплины «Программирование на языке высокого уровня» базируется на результатах освоения следующих дисциплин/практик: «Математика»

Дисциплина является предшествующей для дисциплин/практик: «Введение в веб-технологии», «Объектно-ориентированное программирование», «Web-программирование», «Технологии программирования», «Основы мобильной разработки», «Проектная деятельность»

3 Объем дисциплины

Объем дисциплины составляет – 11 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)		
	Всего	Семестр № 1	Семестр № 2
Общая трудоемкость дисциплины	396	180	216
Аудиторные занятия, в том числе:	160	80	80
лекции	64	32	32
лабораторные работы	64	32	32
практические/семинарские занятия	32	16	16
Самостоятельная работа (в т.ч. курсовое проектирование)	200	100	100
Трудоемкость промежуточной аттестации	36	0	36
Вид промежуточной аттестации (итогового контроля по дисциплине)	Зачет, Экзамен, Курсовой проект	Зачет	Экзамен, Курсовой проект

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 1

№ 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	Ввод/вывод данных	4	3					6	10	Тест
2	Выражения и операторы	2, 6	7							
2	Состав языка программировани я C++									
5	Основные алгоритмические конструкции	9, 10	5	1, 2	18	2, 3	4	1, 4, 5	24	Отчет по лаборатор ной работе
5	Структура программы	1, 3, 5, 12	8			1, 8	4	1, 3, 5	12	
6	Пользовательские структуры данных.	8, 11	5	3, 4, 5	14	4, 5, 6, 7	8	1, 2, 4, 5	54	Отчет по лаборатор ной работе
6	Типы данных	7	4							
	Промежуточная аттестация									Зачет
	Всего		32		32		16		100	

Семестр № 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	Агрегированные типы данных	4	2							
3	Модульное программировани е.	5, 6	6	1, 2	14	1	4	2, 4, 5, 6	30	Отчет по лаборатор ной работе
3	Динамические структуры	7, 8	7	3	10	4	4	2, 4, 5	12	Отчет по лаборатор ной работе
4	Компиляция программ в C++	10	3							
5	Методы отладки программного обеспечения									
6	Ссылки и указатели	2, 3	3							
8	Отладка и тестирование программ	11, 12	4	4	8	3	4	2, 4, 5	12	Отчет по лаборатор ной работе
9	Распределение памяти	1	3							
10	Файлы	9	4			2	4	1, 3	46	Проект
	Промежуточная								36	Экзамен.

	аттестация									Курсовой проект
	Всего		32		32		16		136	

4.2 Краткое содержание разделов и тем занятий

Семестр № 1

№	Тема	Краткое содержание
1	Ввод/вывод данных	Потоковый ввод\вывод данных, заголовочный файл <code>iostream</code> , иерархия потоковых классов, <code>cin</code> -стандартный входной поток, <code>cout</code> - стандартный выходной поток, <code>cerr</code> - стандартный поток вывода сообщений об ошибках, <code>clog</code> -стандартный поток вывода для разного рода диагностических сообщений, операции <code><<</code> и <code>>></code> , форматирование данных, функции, флаги, манипуляторы.
2	Выражения и операторы	Инструкции, выражения, операторы, операнды, арифметические операторы, логические операторы, операции отношения, битовые операции, операции сравнения, инкремент, декремент, префиксный(постфиксный) инкремент (декремент), приоритете операция, ассоциативность.
2	Состав языка программирования C++	Алфавит, лексемы, идентификаторы, ключевые слова, знаки операций, разделители. Основные понятия, статические переменные, представление переменной в памяти, характеристики переменных, объявления переменных, классы памяти, область действия переменных. Инструкции, выражения, операторы, операнды, арифметические операторы, логические операторы, операции отношения, битовые операции, операции сравнения, инкремент, декремент, префиксный(постфиксный) инкремент (декремент), приоритете операция, ассоциативность.
5	Основные алгоритмические конструкции	Операторы условия: <code>if...</code> <code>if...else</code> <code>if...else if...</code> Оператор (выбора) <code>switch</code> (переключатель), тернарный оператор, управляющие операторы (<code>goto</code> , <code>break</code> , <code>continue</code> , <code>exit</code> , <code>return</code>). Циклические конструкции: <code>for</code> , <code>while</code> , <code>do while</code> , цикл <code>for</code> на основе диапазона, вложенные циклы.
5	Структура программы	Функция <code>main()</code> , заголовочные файлы, библиотеки. Алгоритм, постановка задачи, тестирование и отладка программ, анализ результатов. Концепция типа данных, тип данных (базовые, пользовательские), составной тип данных (спецификаторы), преобразование типов.

6	Пользовательские структуры данных.	Определение массива, характеристики массивов, разновидности массивов (одномерные, многомерные), представление в памяти, объявление массивов, инициализация массива, обработка массивов. Строки C-styl, объявление и инициализация строки, ввод строк (cin.getline (), cin. get (), gets(), cin) , вывод строк (оператор cout), функции работы со строками. Строки класса string, объявление и инициализация строки типа string, арифметические операторы.
6	Типы данных	Концепция типа данных, Основные типы данных, спецификаторы, распределение в памяти, форматы базовых типов.

Семестр № 2

№	Тема	Краткое содержание
1	Агрегированные типы данных	Определение и инициализация агрегированных данных, доступ к полям структуры, структуры в структурах. Контейнеры, ассоциативные и последовательные контейнеры.
3	Модульное программирование.	Понятие функции, определение функции, объявление функции, вызов функции, прототип функции, возвращаемые значения, формальные и фактические параметры функции, область видимости объектов, глобальные (локальные) переменные, передача параметров (по значению, по ссылке, по указателю, по умолчанию). Передача параметров функции main(). Перегрузка функций. Шаблоны функций. Префиксная (постфиксная) рекурсия.
3	Динамические структуры	Статические, динамические переменные, выделение, освобождение памяти, списки, классификация списков, объявление динамических структур данных, доступ к данным в динамических структурах, Null указатель, основные действия, производимые над элементами односвязных списков: создание списка; добавление элемента в список (в начало, конец списка); печать (просмотр) списка; поиск элемента в списке; взаимообмен двух узлов списка; удаление элемента из списка; удаление списка. Стек. Очередь. Двухнаправленные списки.
4	Компиляция программ в C++	Многофайловые программы, заголовочные файлы, директивы компилятора #define #undef, условная компиляция, компиляция программы: препроцессинг; компиляция (ассемблирование); компоновка (сборка, линковка); загрузка.
5	Методы отладки программного обеспечения	Классификации ошибок, методы отладки программного обеспечения: метод индукции, метод дедукции, обратное прослеживание,

6	Ссылки и указатели	Переменные, ссылки, указатели, объявление ссылки. Указатели, динамическая память (куча), Связь между переменной, памятью, ссылкой и указателем, объявление и инициализация указателей, операции с указателями: разыменование указателя; получение адреса; операции сравнения, выделение, освобождение памяти, нулевые указатели, арифметика указателей.
8	Отладка и тестирование программ	Классификации ошибок, методы отладки программного обеспечения: метод индукции, метод дедукции, обратное прослеживание. Отладка, тестирование, режимы конфигурации «Debug» и «Release», точка останова, пошаговая отладка, отслеживание переменных,
9	Распределение памяти	Стек, структура стека, особенности организации стека, размещение программы в памяти, пример формирования стека, проблемы стековой памяти, отличия между стеком и кучей.
10	Файлы	Файловый ввод/вывод, основные шаги работы с файлами: создать объект; связать объект соответствующего класса файлового ввода/вывода с файлом (указав имя файла в качестве параметра) в который будет производиться запись (объявить файл, получить к нему доступ, открыть файл); обработать данные в файле (вводить и выводить данные; анализировать ошибки ввода-вывода и достижения конца файла; управлять буферизацией потока и размером буфера; получать и устанавливать указатель текущей позиции в файле); закрыть файл. Текстовые файлы, форматирование текстовых потоков, запись/чтение данных в текстовый файл. Бинарные файлы, запись/чтение данных в бинарный файл.

4.3 Перечень лабораторных работ

Семестр № 1

№	Наименование лабораторной работы	Кол-во академических часов
1	Разветвляющийся вычислительный процесс	10
2	Циклический вычислительный процесс	8
3	Массивы и матрицы	6
4	Литерные переменные	2
5	Строки	6

Семестр № 2

№	Наименование лабораторной работы	Кол-во академических часов
1	Модульный принцип программирования.	8

	Процедуры и функции	
2	Рекурсивные алгоритмы	6
3	Абстрактные типы данных. Работа с динамическими списками	10
4	Работа с графикой в визуальной среде программирования	8

4.4 Перечень практических занятий

Семестр № 1

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Этапы разработки программ	2
2	Реализация разветвляющегося вычислительного процесса	2
3	Реализация циклических алгоритмов и их отладка	2
4	Обработка массивов данных. Сдвиг.	2
5	Сортировка массива	2
6	Обработка массивов данных. Сдвиг.	2
7	Использование литерных переменных и множеств	2
8	Работа с графикой в визуальной среде программирования	2

Семестр № 2

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Модульный принцип программирования	4
2	Работа с типизированными файлами	4
3	Графические возможности среды программирования.	4
4	Работа с динамическими списками	4

4.5 Самостоятельная работа

Семестр № 1

№	Вид СРС	Кол-во академических часов
1	Оформление отчетов по лабораторным и практическим работам	14
2	Подготовка к зачёту	10
3	Подготовка к практическим занятиям	8
4	Подготовка к практическим занятиям (лабораторным работам)	34
5	Подготовка к сдаче и защите отчетов	24
6	Проработка разделов теоретического материала	10

Семестр № 2

№	Вид СРС	Кол-во академических часов
1	Написание курсового проекта (работы)	40
2	Оформление отчетов по лабораторным и практическим работам	8
3	Подготовка к практическим занятиям	6
4	Подготовка к практическим занятиям (лабораторным работам)	28
5	Подготовка к сдаче и защите отчетов	10
6	Проработка разделов теоретического материала	8

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: Работа в малых группах, мозговой штурм, метод проектов, лекция с ошибками, Дискуссия, Опережающее обучение, Работа в команде, Взаимная проверка и тестирование алгоритмов

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по курсовому проектированию/работе:

Бахвалова З.А. электронный курс «Программирование на языке высокого уровня C++»
<https://el.istu.edu/course/view.php?id=5945>

5.1.2 Методические указания для обучающихся по практическим занятиям

Бахвалова З.А. электронный курс «Программирование на языке высокого уровня C++»
<https://el.istu.edu/course/view.php?id=5945>

5.1.3 Методические указания для обучающихся по лабораторным работам:

Бахвалова З.А. электронный курс «Программирование на языке высокого уровня C++»
<https://el.istu.edu/course/view.php?id=5945>

5.1.4 Методические указания для обучающихся по самостоятельной работе:

мБахвалова З.А. электронный курс «Программирование на языке высокого уровня C++»
<https://el.istu.edu/course/view.php?id=5945>

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 семестр 1 | Тест

Описание процедуры.

Студенты проходят тест только в присутствии преподавателя и в специально отведенное для этого время.

Студент заходит в систему MOODLE, используя свой логин и пароль. Подучает доступ к тесту в определенное преподавателем время.

Студент приступает к выполнению тестовых заданий, следуя инструкциям системы.

После завершения тестирования результаты автоматически отображаются на экране, Во время тестирования студентам запрещается разговаривать, использовать посторонние материалы (учебники, тетради, мобильные телефоны и т.д.), а также списывать у других студентов.

Критерии оценивания.

Тест считается сданным, если студент ответит правильно не меньше чем на 75% вопросов
пример теста:

1. Стандартное представление периодической дроби 1.0 / 3.0 при выводе с помощью оператора cout.

Результат работы оператора

```
cout << "\n1.0 / 3.0 = " << 1.0 / 3.0;
```

2. Выберите правильные утверждения:

Флаги, функции и манипуляторы выполняют одну и ту же задачу.

Различие между функциями, флагами и манипуляторами форматирования состоит в способе их применения

Манипуляторы включаются в операторы ввода-вывода.

Флаги реализованы в виде отдельных фиксированных битов чисел типа long.

При выводе данных можно установить только один флаг форматирования.

Флаги, функции и манипуляторы задают разные форматы ввода/вывода информации.

3. Соотнесите ответы к предложенным вариантам:

1. cout.setf(ios::boolalpha) – Выберите элемент.

2. cout.precision(4); – Выберите элемент.

3. cout << showpos << 25; – Выберите элемент.

Ответы: 1 – флаг, 2 – функция, 3 – манипулятор

4. Введите результат работы программы:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    double num = 13.19370;  
    cout.precision(5);  
    cout << num << endl;  
    return 0;  
}
```

Ответ вводится студентом: 13.194

5. Операция вывода класса ostream называется [1] данных в поток. Она обозначается с помощью символа операции сдвига [2].

- Вставкой [1]
- Влево [2]

6.1.2 семестр 1 | Отчет по лабораторной работе

Описание процедуры.

Подготовка отчетов выполняется студентами самостоятельно. Отчетом по выполнению лабораторных работ является сформированный документ, названный по теме лабораторной работы и фамилии автора работы, в котором содержатся все необходимые сведения и выводы по ходу выполнения работы по конкретной теме.

При защите отчетов преподавателем проверяется: правильность и творческий подход к выполнению заданий, знание теоретического материала необходимого для выполнения работ.

Требования к оформлению отчета к лабораторным работам №№ 1-5.

1. Постановка задачи
2. Математическая модель.
3. Таблица спецификаций.
4. Таблица сообщений об ошибках.
5. Таблица тестов.
6. Блок-схема алгоритма.
7. Программа на языке программирования.
8. Результаты тестирования.

Пример вопросов для контроля к лабораторной работе №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

Пример вопросов для контроля к лабораторной работе №2.

1. Сколько операторов повторения вы знаете? Перечислите их.
2. Оператор For - общий вид, реализуемый алгоритм, примеры.
3. Оператор Repeat - общий вид, реализуемый алгоритм, примеры.
4. Оператор While - общий вид, реализуемый алгоритм, примеры.
5. Чем цикл с предусловием отличается от цикла с постусловием? Почему таких циклов два вида?
5. Определить значение переменной s после выполнения следующих операторов:
a) s = 0; i = 0;
while i < 5
{ i = i + 1; s = s + 1/i; }
б) s = 0; i = 1;
do { s = s + 1/i; i = i - 1 } while(i > 1);

Пример вопросов для контроля к лабораторной работе №3.

1. Что такое массив? Примеры массивов.
2. Основные понятия: массив, элемент массива, индекс элемента, количество элементов.
3. Размерность массива.
4. Описание массивов на C++.
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?

Пример вопросов для контроля к лабораторной работе №4.

1. Сколько байт занимает литерная переменная?
2. Как определить код ASCII литерной переменной?

3. Какая функция служит для преобразования строчной буквы в прописную?
4. Как получить следующий/предыдущий символ в таблице ASCII?

Пример вопросов для контроля к лабораторной работе №5.

1. Два способа представления текста в C++.
2. Тип данных string: описание переменных, атрибут длины, ввод, вывод и операция присваивания. Примеры.
3. Сравнение строк.
4. Стандартные процедуры перевода строки в число и числа в строку.
5. Назначение стандартных функций определения длины строки, определение позиции символа в строке, копирование строки .

Критерии оценивания.

1. Отчет содержит все необходимые компоненты и выполнен без ошибок
2. Обучающийся верно строит математическую модель задачи.
3. Правильно выбирает и описывает спецификации программы
4. Верно проектирует алгоритм
5. Производит проектирование полных тестовых наборов, доказывающих корректность алгоритма
6. Свободно владеет языком программирования

6.1.3 семестр 2 | Отчет по лабораторной работе

Описание процедуры.

Подготовка отчетов выполняется студентами самостоятельно. Отчетом по выполнению лабораторных работ является сформированный документ, названный по теме лабораторной работы и фамилии автора работы, в котором содержатся все необходимые сведения и выводы по ходу выполнения работы по конкретной теме.

При защите отчетов преподавателем проверяется: правильность и творческий подход к выполнению заданий, знание теоретического материала необходимого для выполнения работ.

Вопросы для контроля к лабораторным работам № 6,7.

1. Какова структура описания процедуры и функции?
2. В чем состоит отличие процедуры от функции?
3. Что такое область действия идентификаторов?
4. Каковы основные правила определения области действия для идентификаторов процедур и 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. Что такое рекурсивный алгоритм?

Вопросы для контроля к лабораторной работе № 8.

1. Чем отличаются статические и динамические величины?
2. Какая память называется динамически распределяемой?
3. Что такое указатель?

4. Какие виды указателей вам известны?
5. Как определяется адрес переменной?
6. Что такое "разыменование"?
7. Приведите примеры объявления указателей.
8. Как выделить память под динамическую переменную? Как освободить память от динамической переменной?

Вопросы для контроля к лабораторной работе №9.

1. С помощью каких компонент представить графические данные в VS?
2. Для чего служит канва?
3. Как поменять стиль пера?
4. Как вывести точку в заданные координаты?
5. Каким образом используется компонент Timer?

Критерии оценивания.

Работа выполнена полностью.

Работа оформлена в соответствии с требованиями.

Тестовые наборы являются полными.

Программа имеет удобный пользовательский интерфейс

Программа работоспособна

Даны ответы на контрольные вопросы по теме работы

6.1.4 семестр 2 | Проект

Описание процедуры.

Курсовая работа по учебной дисциплине «Программирование на языке высокого уровня» выполняется в соответствии с индивидуальным заданием и предполагает выполнение

следующих этапов:

1. Внешнее проектирование программы.
2. Выбор тестовых наборов.
3. Согласование результатов внешнего проектирования с руководителем курсовой работы.
4. Внутреннее проектирование программы.
5. Кодирование и отладка программы.
6. Тестирование программы.
7. Оформление и сдача работы.

Порядок выполнения курсовой работы и график выполнения

1. Получить индивидуальное задание согласно варианту (1 неделя).
2. Уточнить все вопросы по теме работы у руководителя проекта (2 неделя).
3. Изучить способы описания синтаксиса языков (3-4 недели).
4. Выполнить внешнее проектирование разрабатываемых игровых программ (5-6 недели).
5. Программирование игровых программ на языках Паскаль и Си (7-14 недели)
6. Отладка и тестирование (15 неделя)
7. Оформление пояснительной записки (16 неделя)
8. Защита курсовой работы по графику защит (17 неделя).

Критерии оценивания.

1. программа должна иметь модульную структуру, т.е. должна состоять из отдельных функций
2. формирование файла (запись введенных данных в файл);
3. редактирование данных (добавление, удаление, изменение сведений);
4. использование отдельного диалога для ввода и проверка корректности всех входных данных;
5. вывод исходных данных (просмотр всех сведений) и реализация всех указанных в задании действий по сортировке и выборке данных;
6. использование главного и контекстного меню;
7. результаты расчетов и поиска должны быть оформлены в соответствующем виде (с использованием форматированного вывода);
8. наличие в главном меню раздела «О программе», в котором указать версию программного продукта, дату внесения последних изменений в программе и координаты автора;
9. наличие в главном меню раздела «Руководство пользователя (Справка)», содержание которого загружается из текстового файла;
10. наличие в главном меню раздела «Задание», содержащего постановку задачи.
11. корректная обработка диалога работы с файлами (например, запрет открытия несуществующего файла, либо запрос на перезапись уже существующего файла, настройка фильтров и др.);
12. проверка сохранения сведений в файле при выходе из программы;
13. сортировка выводимой информации по любому из полей записи (критерий сортировки указывается в меню), используя одну процедуру сортировки для всех пунктов меню;
14. по запросу пользователя сохранить результаты поиска и расчетов в новом файле
15. формирование списка ранее открывавшихся файлов;
16. поиск по любому из полей записи, используя одну функцию для поиска;
17. предусмотреть возможность поиска по сочетанию первых символов (фамилии на одинаковое сочетание первых букв и т.д. (символы вводятся пользователем));
18. корректно обрабатывать доступность/недоступность пунктов меню и кнопок в зависимости от действий пользователя.

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-6.1	Уверенно демонстрирует знания основ структурного программирования. Способен создать программу на языке высокого уровня С++, выполнить ее компиляцию и провести тестирование разработанного кода	Устное собеседование и/или практические задания и/или лабораторные работы
ОПК ОС-6.2	Грамотно использует алгоритмические структуры при написании программ на	Устное собеседование

	языке высокого уровня C++. Обоснованно применяет различные стратегии тестирования программ.	и/или практические задания и/или лабораторные работы
--	---	--

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 1, Типовые оценочные средства для проведения зачета по дисциплине

6.2.2.1.1 Описание процедуры

Зачет проводится в устной форме и заключается в ответах на теоретические вопросы (возможны вопросы в рамках отчетов по лабораторным работам).

Для допуска к зачету должны быть выполнены и защищены все лабораторные работы за 1 семестр. На зачете студент должен устно ответить на вопросы билета, а также быть способен составить программный код по тематике вопроса по требованию преподавателя. Студент вправе не отвечать на вопросы билета и получить зачет по дисциплине, если он не пропустил ни одной лекции по дисциплине в течение 1 семестра и активно работал в ходе лабораторных работ. Студент вправе отвечать только на один из вопросов билета (по своему выбору) и получить зачет по дисциплине, если он пропустил не более 2 лекций по дисциплине в 1 семестре. Допускается письменный ответ на вопросы билета на зачете (по решению преподавателя)

Контрольные вопросы к зачету:

1. Алгоритм. Свойства алгоритма. Изображение алгоритма.
2. Этапы решения задач на ЭВМ.
3. Алфавит языка C++.
4. Типы данных языка C++.
5. Представление чисел в памяти ЭВМ.
6. Описание переменных, типов, констант. Операторы ввода/вывода.
7. Структура C++ -программы.
8. Простые операторы: присваивание, безусловный переход, пустой оператор.
9. Виды циклов. Операторы циклов в C++.
10. Массивы. Обработка массивов.
11. Работа со строками.
12. Обработка символьных данных.

Пример задания:

Пример задач к зачету:

1. Найти НОД двух целых чисел.
2. Определить является ли натуральное число простым.
3. Дан массив A(10) вещественного типа. Найти полусумму максимального и минимального элементов

Типовой билет к зачёту:

1. Массивы. Способы описания. Размещение в памяти. Обработка массивов.
2. Типы данных языка C++.

6.2.2.1.2 Критерии оценивания

Зачтено	Не зачтено
Демонстрирует навыками работы в среде программирования. Решает поставленные задачи, используя язык программирования Грамотно строит алгоритмы и подбирает тестовые наборы. Способен доказать корректность программы. Четко и ясно аргументирует использование приобретенных знания и умений при решении задач	Не владеет основным материалом курса. Демонстрирует навыки работы в среде программирования, строит алгоритмы с ошибками. Неверно выбирает типы данных, неверно описывает спецификации, не умеет проектировать тесты. Решает поставленные задачи с ошибками

6.2.2.2 Семестр 2, Типовые оценочные средства для проведения экзамена по дисциплине

6.2.2.2.1 Описание процедуры

Экзамен ставится по результатам устных ответов на вопросы из экзаменационного билета. В ходе экзамена студент должен устно ответить на вопросы билета, а также быть способен составить программный код по тематике вопроса по требованию преподавателя. Допускается письменный ответ на вопросы билета на зачете (по решению преподавателя). Для допуска к экзамену должны быть выполнены и защищены все лабораторные работы за 1 и 2 семестры.

Вопросы к экзамену:

1. Достоинства и недостатки C++.
- 2 Структура программы на C++
3. Понятия системы программирования, языка программирования, программы.
4. Алгоритм. Свойства алгоритма.
5. Основные конструкции (следование, развилка, выбор, цикл). Изображение алгоритма.
6. Этапы решения задач на ЭВМ.
7. Тестирование и отладка. Построение тестов. Примеры.
8. Алфавит языка C++. Арифметические выражения, правила их записи.
9. Основные функции библиотеки math.h. Комментарии.
10. Арифметические операции, операции отношения.
11. Логические и сдвиговые операции.
12. Типы данных языка C++.
13. Преобразование типов.
14. Представление чисел в памяти ЭВМ. Целые числа.
15. Представление чисел в памяти ЭВМ. Вещественные числа.
16. Представление символьных данных в памяти ЭВМ. Литерные переменные.
17. Описание переменных, констант.
18. Перечислимый тип данных.
19. Манипуляторы потока. Управляющие символы.
20. Структура программы C++.
21. Простые операторы: присваивание, безусловный переход, пустой оператор.
22. Операторы управления: goto, If, switch, continue, break.
23. Массивы. Способы описания.

24. Массивы. Размещение в памяти. Обработка массивов.
25. Виды циклов. Операторы циклов в C++. Примеры простых и вложенных циклов.
26. Сортировка массивов. Основные методы сортировки.
27. Матрицы. Работа с матрицами.
28. Обработка строковых данных. Функции для работы со строками.

Пример задания:

Типовой билет к экзамену:

1. Структура программы C++.
2. Алгоритм. Свойства алгоритма.
3. Задача: Из элементов целочисленного вектора $C(n)$, $n = 12$, кратных заданному целому числу T , сформировать вектор B .

6.2.2.2.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Демонстрирует навыками работы в среде программирования. 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.	Демонстрирует навыки работы в среде программирования, но при этом в ответах допускает неточности. Владеет теоретическим материалом. Решает поставленные задачи с незначительными неточностями. Способен доказать корректность алгоритма, выполнить тестирование и отладку.	Нечетко отвечает на теоретические вопросы, допускает ошибки при ответах. Демонстрирует навыки работы в среде программирования. Решает поставленные задачи, может допускать ошибки, сомневается в принятом решении. Подбирает неполный набор тестов для проверки корректности решения.	Не может дать правильных ответов на поставленные теоретические вопросы. Допускает серьезные ошибки при решении задач. Неверно выбирает способ решения задачи, технологии программирования

6.2.2.3 Семестр 2, Типовые оценочные средства для курсовой работы/курсового проектирования по дисциплине

6.2.2.3.1 Описание процедуры

При выполнении курсовой работы и защите пояснительной записке по ней необходимо продемонстрировать знание теоретического материала курса «Программирование» и практические навыки проектирования программного обеспечения, разработки приложений, отладки на примере решения игровой задачи

Пример задания:

Примеры индивидуальных заданий:

1 Игра “Морской бой” (два игрока)

На поле 4*4 клетки игроки устанавливают 3 корабля по одной клетке (у каждого игрока свое поле). Программа “запоминает” положение кораблей. Затем игроки начинают поражать корабли противника, по очереди указывая координаты предполагаемого корабля. Результат попадания (попал или мимо) отмечается на поле (например, “*” – попал, “+” – мимо).

2. Игра "Тренировка памяти– слова" (1 игрок) Программа на определенное время высвечивает несколько слов- существительных. Слова выбираются из специального файла или из специальной таблицы случайным образом. Игрок

должен воспроизвести слова. Время для запоминания может быть различным. Например, в программе предусматривается три временных режима Число слов для запоминания может быть различным. Игрок может играть в 2-х режимах:

а) просто воспроизвести слова; б) воспроизвести слова в заданном порядке.

Обучающийся может предложить свою тему работы, согласовав с преподавателем.

6.2.2.3.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Все этапы курсового проектирования выполнены в срок и качественно. Пояснительная записка к курсовой работе оформлена в соответствии с требованиями. Качественно выполнена теоретическая и практическая части работы. Задача реализована на языке программирования Программа имеет удобный пользовательский	Работа выполнена полностью согласно заданию. Пояснительная записка оформлена в соответствии с требованиями (с незначительными неточностями). Поставленная в курсовой работе задача решена с использованием языка программирования, имеются неучтенные ситуации. Обучающийся уверенно дает ответы на поставленные вопросы по теме	Пояснительная записка оформлена с незначительными нарушениями требований. Программа реализована, но рассматривается частный случай. Программа имеет небрежно оформленный интерфейс. Программа не проходит тестовые наборы преподавателя. Обучающийся путается в ответах	Пояснительная записка оформлена небрежно, с нарушением требований. Реализованная программа не решает поставленную задачу или решает частично.

интерфейс и работоспособна, предусмотрены все возможные ситуации. Даны ответы на контрольные вопросы .	выполненной работы		
--	--------------------	--	--

7 Основная учебная литература

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

2. Аршинский В. Л. Программирование на языке высокого уровня. Основы алгоритмизации и программирования : учебное пособие / В. Л. Аршинский, И. А. Серышева, 2021. - 122.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files3/er-27193.pdf>

3. Конова Е. А. Алгоритмы и программы. Язык С++ : учебное пособие для вузов / Е. А. Конова, Г. А. Поллак, 2023. - 384.

[Сайт] – URL: <https://e.lanbook.com/book/297002>

8 Дополнительная учебная литература и справочная

1. Паппас Крис. Программирование на С и С++ : [Учеб. пособие : Пер. с англ.] / Крис Паппас, Уильям Мюррей, 2000. - 318.

2. Кнут Дональд Э. Искусство программирования для ЭВМ : в 7т.: Пер. с англ. Т. 1. Основные алгоритмы/Пер. под ред. К. И. Бабенко, В. С. Штаркмана / Д. Кнут, 1976. - 735.

3. Кнут Дональд Э. Искусство программирования для ЭВМ : в 7т.: Пер. с англ. Т. 2. Получисленные алгоритмы/Пер. под ред. К. И. Бабенко / Д. Кнут, 1977. - 724.

4. Кнут Дональд Э. Искусство программирования для ЭВМ : в 7т.: Пер. с англ. Т. 3. Сортировка и поиск/Пер. под ред. Ю. М. Баяковского, В. С. Штаркмана / Д. Кнут, 1978. - 844.

5. Саттер Г. Новые сложные задачи на С++ : 40 новых головолом. примеров с решениями : пер. с англ. / Герб Саттер, 2005. - 271.

6. Павловская Т. А. С/С++. Структурное и объектно-ориентированное программирование : практикум / Т. А. Павловская, Ю. А. Щупак, 2011. - 347.

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

1. Microsoft Windows Professional 8 Russian

12 Материально-техническое обеспечение дисциплины

1. Проектор Epson EB-W04LCD.WXGA 1280*800.3000:1.2800 ANSI Lumens
2. Компьютер Asustek P8H6-M/Intel Core i5
2400/4Gb/HDD2TB/DVD-RW/ATX550W/LCD22/ИБП1