

МИНОБРНАУКИ РОССИИ



Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
ФГАОУ ВО «РГГУ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ
Факультет информационных систем и безопасности
Кафедра фундаментальной и прикладной математики

**РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММНЫХ СРЕДСТВ
В ЗАДАЧАХ МАШИННОГО ОБУЧЕНИЯ**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Прикладная математика (бакалавриат) 01.03.04

Математические основы искусственного интеллекта

Уровень квалификации выпускника (бакалавр)

Форма обучения (очная)

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2026

РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММНЫХ СРЕДСТВ В ЗАДАЧАХ МАШИННОГО ОБУЧЕНИЯ

Рабочая программа дисциплины

Составители:

Канд. тех. наук, доцент кафедры Фундаментальной и прикладной математики А.Б.Клименко

УТВЕРЖДЕНО

Протокол заседания кафедры
фундаментальной и прикладной математики
№ 5 от 19.12.2025

ОГЛАВЛЕНИЕ

1.	<u>Пояснительная записка</u>	4
1.1.	<u>Цель и задачи дисциплины</u>	4
1.2.	<u>Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций</u>	4
1.3.	<u>Место дисциплины в структуре образовательной программы</u>	4
2.	<u>Структура дисциплины</u>	5
3.	<u>Содержание дисциплины</u>	5
4.	<u>Образовательные технологии</u>	5
5.	<u>Оценка планируемых результатов обучения</u>	6
5.1.	<u>Система оценивания</u>	6
5.2.	<u>Критерии выставления оценки по дисциплине</u>	6
5.3.	<u>Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине</u>	7
6.	<u>Учебно-методическое и информационное обеспечение дисциплины</u>	12
6.1.	<u>Список источников и литературы</u>	12
6.2.	<u>Перечень ресурсов информационно-телекоммуникационной сети «Интернет»</u>	13
6.3.	<u>Профессиональные базы данных и информационно-справочные системы</u>	13
7.	<u>Материально-техническое обеспечение дисциплины</u>	13
8.	<u>Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов</u>	14
9.	<u>Методические материалы</u>	15
9.1.	<u>Планы практических занятий</u>	15
	<u>Приложение 1. Аннотация рабочей программы дисциплины</u>	18

1. Пояснительная записка

1.1. Цель и задачи дисциплины

Цель дисциплины: обучение студентов программированию типовых задач управления информационными процессами и обработки информации с использованием современных программных средств в задачах машинного обучения.

Задачи дисциплины: обеспечить овладение будущими специалистами методами создания эффективных алгоритмов и программ с использованием современных структур данных, программной документацией и способами оценки результатов работы программ.

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

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
ПК-3. Способен осуществлять поиск, изучение и разработку новых теоретических или практических проблем, сведений, относящихся к решению текущих научных исследований, производственных задач; в информационных средах находить, создавать основные элементы будущих математических структур или конструктивных математических моделей	ПК-3.1. Владеет навыками работы с информационными системами для разработки новых теоретических положений и решения практических проблем.	<p>Знать: конструирование средствами используемого языка программирования новых типов данных, соответствующих специфике решаемой задачи.</p> <p>Уметь: ставить задачу; эффективно выбирать структуры данных и представления их в ЭВМ в зависимости от решаемой задачи и доступных вычислительных ресурсов.</p> <p>Владеть: проектированием эффективных алгоритмов обработки информационных структур; созданием программных модулей, реализующих модели информационных структур; обработкой на ЭВМ данных различной структуры, используемых в современных задачах проблемного и системного программирования.</p>

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Разработка алгоритмов и программных средств в задачах машинного обучения» относится к части, формируемой участниками образовательных отношений, блока дисциплин учебного плана.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин: «Иностранный язык», «Линейная алгебра», «Аналитическая геометрия», «Программные и аппаратные средства информатики», «Введение в теоретическую информатику».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: «Базы данных», «Информационные технологии», «Численные методы», «Высокоуровневые языки программирования», «Алгоритмы и структуры данных в задачах ИИ», «Основы теории алгоритмов», «Программные средства научных исследований», Учебная практика «Научно-исследовательская работа (получение первичных навыков научно-исследовательской деятельности)».

2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 4 з.е., 144 академических часа (ов).

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
3	Лекции	24
3	Практические занятия	32
Всего:		56

Объем дисциплины в форме самостоятельной работы обучающихся составляет 88 академических часа(ов).

3. Содержание дисциплины

Тема 1. Классы и их структура в языке С++.

Структуры и классы языка С++. Классы и объекты. Заголовочные файлы классов. Файлы реализации классов. Члены класса – методы и поля. Конструкторы классов. Деструкторы классов.

Тема 2. Свойства членов класса.

Типы защищённости членов класса. Обеспечение доступа к защищённым членам класса. Статические поля данных и методы классов. Встроенные методы классов. Объекты как параметры функций.

Тема 3. Доступ операторов и функций к членам класса.

Операторные функции – члены класса. Операторные функции – не члены класса. Перегрузка операторов в классе. Функции – друзья класса. Функции потокового ввода и вывода объектов.

Тема 4. Наследование и связывание классов. Виртуальные методы.

Принципы объектно-ориентированного программирования. Отношения классов. Типы наследования классов. Доступность членов класса в классах-наследниках. Раннее и позднее связывание классов. Сравнение статического и динамического связывания классов. Виртуальные методы классов.

4. Образовательные технологии

Для проведения *занятий лекционного типа* по дисциплине применяются такие образовательные технологии как лекция-визуализация с применением слайд-проектора.

Для проведения *практических занятий* используются такие образовательные технологии как: решение типовых задач для закрепления и формирования знаний, умений, навыков.

В рамках *самостоятельной работы* студентов проводится консультирование и проверка домашних заданий посредством электронной почты.

В период временного приостановления посещения обучающимися помещений и территории РГГУ для организации учебного процесса с применением электронного обучения и дистанционных образовательных технологий могут быть использованы следующие образовательные технологии:

- видео-лекции;
- онлайн-лекции в режиме реального времени;
- электронные учебники, учебные пособия, научные издания в электронном виде и доступ к иным электронным образовательным ресурсам;
- системы для электронного тестирования;
- консультации с использованием телекоммуникационных средств.

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль: - домашнее задание - тестирование	8 баллов 20 баллов	40 баллов 20 баллов
Промежуточная аттестация – зачет с оценкой (Ответы на вопросы)		40 баллов
Итого за семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала	Шкала ECTS
95 – 100	Отлично	A
83 – 94		B
68 – 82	Хорошо	C
56 – 67		D
50 – 55		E
20 – 49	Неудовлетворительно	FX
0 – 19		не зачтено

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A, B	отлично	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
82-68/ С	хорошо	Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей. Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами. Достаточно хорошо ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».
67-50/ D,E	удовлетворительно	Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».
49-0/ F,FX	неудовлетворительно	Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Примерные варианты домашних заданий:

Задание 1. Объявить структуру с именем **PlanetInfo** для хранения следующей информации: **name** (string), **diameter** (int), **haslife** (bool) и **moons** (int).

В классе **Planet** определить конструктор, деструктор и поле данных типа **PlanetInfo**.

Написать метод класса **GetPlanet** для чтения и сохранения информации об одной планете в поле данных класса. Формат ввода дан ниже.

Написать метод класса **Print** для печати информации в удобном формате.

Объявить константу с понятным именем для количества планет и создайте массив планет в **main()**.

Циклическим вызовом **GetPlanet** ввести с клавиатуры информацию о 9 планетах. Возможные данные о планетах (уточните их!):

Меркурий 4878 false 0

Венера 12104 false 0

Земля 12774 true 1

Марс 6786 true 2

Юпитер 142796 false 16

Сатурн 120000 false 17

Уран 51108 false 5

Нептун 49600 false 2

Плутон 2280 false 1

Изменить функцию **Print** для однократного её вызова при печати данных о всех планетах.

Написать перегруженную функцию сортировки массива планет по диаметру, количеству спутников и названиям.

Задание 2. Для заданного целого $n > 1$ наименьшее целое $d > 1$, на которое делится n – **простой множитель**. Разложение n на простые множители можно произвести поиском d с дальнейшей заменой n на частное от деления n на d и повторяя этот процесс, пока n не станет равным 1.

Написать программу разложения n на простые множители этим способом с печатью простых множителей сначала по убыванию, а потом – по возрастанию. **ИСПОЛЬЗОВАТЬ СТЕК!**

Например, для $n=3960$ программа должна дать

$11 * 5 * 3 * 3 * 2 * 2 * 2$ и затем $2 * 2 * 2 * 3 * 3 * 5 * 11$

Задание 3. Создать класс для представления и работы с обыкновенными дробями. На экране значения дробей должны иметь вид $-2 \frac{3}{4}$ (для отрицательных), $\frac{5}{8}$ (для положительных, в случае, когда числитель меньше знаменателя) или 3 (если знаменатель равен 1).

Сконструировать арифметические операции и операции сравнения для обыкновенных дробей (+, -, *, /, >, <, ==, !=). Использовать функции - члены класса и функции – не члены класса.

В главной программе проверить все операции и вывести результаты на экран.

Задание 4. Написать программу ввода и оперирования полиномами, состоящими из термов. Использовать классы **Term** and **Polynomial**.

Ввод

- Термы полинома могут вводиться в любом порядке.
- Коэффициент 1 или -1 может присутствовать или отсутствовать.
- Терм может вычитаться из другого терма (например, допустим ввод $3x^2 - x$)
- Пробелы могут появляться где угодно.

класс Term

- Целые члены-данные для коэффициента и показателя степени
- Три конструктора
 - Без параметров для представления $0x^0$
 - С одним параметром, например 3, для представления $3x^1$
 - С двумя параметрами, например 3 и 2, для представления $3x^2$
- Конструктор копирования и оператор присваивания
- Друзья класса: **operator ***, **operator +**, каждый из которых получает 2 терма как параметры и возвращает терм-результат.
- Друг класса **ostream** << для печати терма в виде:
 $3x^0$ как **3**, $3x^1$ как **3x**, $1x^3$ как **x^3**, $-3x^2$ как **-3x^2**

Дружественный класс Polynomial

класс Polynomial

- Члены-данные **poly** (массив из 6 термов), и целое **degree**
- Три конструктора
 - Без параметров для представления полинома 0
 - С одним целым параметром, например 3, для представления полинома 3
 - С одним параметром-термом, например **Term(3,2)**, для представления полинома $3x^2$
- Конструктор копирования и операторы присваивания =, +=, *=

- Скрытый член **Order** для хранения термов по возрастанию или убыванию степени
- Друзья класса: **operator ***, **operator +**, каждый из которых получает 2 полинома как параметры и возвращает полином-результат.
- Друг класса **ostream** << для печати полинома в виде:
 $5x^5 - 4x^3 + 1x^2 + 0x^1 - 7x^0$ как $5x^5 - 4x^3 + x^2 - 7$

Представить *main()* для демонстрации работы класса *Polynomial* и 5 файлов: *Term.h*, *Term.cpp*, *Polynomial.h*, *Polynomial.cpp*, *mymain.cpp*.

Задание 5. Создать класс с использованием динамической памяти для реализации множества строк. Класс с именем **MySet** включает следующие члены:

- **Члены - данные (private)**
 - **size** – количество элементов в множестве
 - **elements** – указатель на строку, содержащий адрес динамического массива строк
- **Конструкторы и деструктор**
 - Конструктор без параметров для создания пустого множества
 - Конструктор с одним строковым параметром для создания множества размером 1
 - Конструктор, копирующий множество
 - Деструктор
- **Методы доступа**
 - **Size** для возврата количества элементов в множестве
 - **IsElement**, который даёт **true**, если строка-параметр есть в множестве, иначе даёт **false**
- **Методы изменения**
 - **AddElement** для добавления строки в множество, если её там ещё нет
 - **DeleteElement** для удаления строки из множества, если она там есть
- **Операторы**
 - **[]** для возврата ссылки на элемент множества
 - **операторы присваивания** =, -=, +=, *= где - означает разность, + - объединение и * - пересечение. (См. примеры ниже.)
- **Функции – не члены класса (не друзья класса!)**
 - Функция **печати** элементов множества - не более 5 элементов в строке
 - **Операторы** + (объединение), - (разность), * (пересечение) и == (сравнение: истина, если элементы двух множеств совпадают).

Например:

$\{1, 4, 5, 6\} + \{1, 2, 3, 4\} \Rightarrow \{1, 2, 3, 4, 5, 6\}$

$\{1, 4, 5, 6\} * \{1, 2, 3, 4\} \Rightarrow \{1, 4\}$

$\{1, 4, 5, 6\} - \{1, 2, 3, 4\} \Rightarrow \{5, 6\}$

Нельзя добавлять методы-члены класса в раздел **public** и данные-члены в раздел **private**. При желании можно добавить методы в раздел **private**.

Примерные вопросы для тестирования:

1. Конструктор имеет то же имя, что
 А) класс В) объект С) программа D) другое.
2. Кратко опишите назначение деструктора.
3. Когда вызывается конструктор копирования?
4. Какова разница между членами класса из разделов private и protected?

5. Для класса с именем BlackBox напишите заголовок функции-члена класса для преобразования объекта BlackBox к типу int.
6. Если в классе X функция A объявлена другом, становится ли эта функция членом класса X ?
7. Объект какого типа возвращает перегруженный оператор >> ?
8. Перегруженные функции имеют разные типы, разные списки параметров или и то и другое вместе?
9. Объяснить, почему в разделе private недопустимо описание виртуальных функций.
10. Каковы признаки того, что созданный класс является абстрактным?

11. Что выведет на экран эта программа?

```
#include <iostream>
using namespace std;
class Tank
{
private:
    int litres;
public:
    Tank()
        {litres = 50;}
    Tank(int gal)
        {litres = gal;}
    int getLitres()
        {return litres;}
};
void main()
{
    Tank storage1, storage2, storage3(20);
    cout << storage1.getLitres() << endl;
    cout << storage2.getLitres() << endl;
    cout << storage3.getLitres() << endl;
}
```

12. Что будет выведено на экран следующей программой?

```
#include <iostream>
using namespace std;
class Base
{
protected:
    int baseVar;
public:
    Base(int val=2){baseVar=val;}
    int getVar() {return baseVar;}
};
class Derived : public Base
{
private:
    int deriVar;
```

```

public:
    Derived(int val=100){deriVar=val;}
    int getVar() {return deriVar;}
};
void main()
{
    Derived object;
    Base *optr = &object;
    cout << optr -> getVar() << endl;
}

```

13. Что будет выведено на экран следующей программой?

```

#include <iostream>
using namespace std;
class First
{
    protected:
        int a;
    public:
        First(int x=1) {a=x;}
        int GetVal(){return a;}
};
class Second: public First
{
    private:
        int b;
    public:
        Second(int y=5){b=y;}
        int GetVal(){return b;}
};
void main()
{
    First Object1;
    Second Object2;
    cout << Object1.GetVal() << endl;
    cout << Object2.GetVal() << endl;
}

```

14. Что выведет на экран эта программа?

```

#include <iostream>
using namespace std;
class Sky
{
    public:
        Sky() {cout << "Entering the sky.\n";}
        ~Sky() {cout << "Leaving the sky.\n";}
};
class Ground: public Sky
{
    public:
        Ground() {cout << "Entering the ground.\n";}
}

```

```

~Ground() {cout << "Leaving the ground.\n";}
};
void main()
{
    Ground Hog;
}

```

Промежуточная аттестация (зачет с оценкой)

Контрольные вопросы по дисциплине:

1. Структуры и классы языка C++.
2. Классы и объекты.
3. Заголовочные файлы классов.
4. Файлы реализации классов.
5. Члены класса – методы и поля.
6. Конструкторы классов.
7. Деструкторы классов.
8. Типы защищённости членов класса.
9. Обеспечение доступа к защищённым членам класса.
10. Статические поля данных и методы классов.
11. Встроенные методы классов.
12. Объекты как параметры функций.
13. Операторные функции – члены класса.
14. Операторные функции – не члены класса.
15. Перегрузка операторов в классе.
16. Функции – друзья класса.
17. Функции потокового ввода и вывода объектов.
18. Принципы объектно-ориентированного программирования.
19. Отношения классов.
20. Типы наследования классов.
21. Доступность членов класса в классах-наследниках.
22. Раннее и позднее связывание классов.
23. Сравнение статического и динамического связывания классов.
24. Виртуальные методы классов.
25. Понятие абстрактных классов, основные правила построения.
26. Применение абстрактных классов.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Литература

Основная

1. Подбельский В.В. Язык СИ++ : учеб. пособие для студентов вузов, обучающихся по направлениям "Прикладная математика" и "Вычислительные машины, комплексы, системы и сети" / В. В. Подбельский. - 5-е изд. - М. : Финансы и статистика, 2008. - 559 с. : рис.,табл.
2. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2023. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/515142> .

3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 214 с. — (Профессиональное образование). — ISBN 978-5-534-15731-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/510042>.
4. Сухарев, А. Г. Численные методы оптимизации : учебник и практикум для вузов / А. Г. Сухарев, А. В. Тимохов, В. В. Федоров. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 367 с. — (Высшее образование). — ISBN 978-5-534-17381-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/538182>.
5. Жмудь, В. А. Теория автоматического управления. Замкнутые системы : учебное пособие для вузов / В. А. Жмудь. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 234 с. — (Высшее образование). — ISBN 978-5-534-05119-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539500>.

Дополнительная

1. Дейтел Харви М. Как программировать на C++ / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. В. В. Тимофеева. - 5-е малое изд. - М. : БИНОМ, 2007. - 799 с. : рис. +2008г.
2. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2023. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511891>.
3. Чернышев, С. А. Основы программирования на Python : учебное пособие для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/544194>.
4. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 227 с. — (Профессиональное образование). — ISBN 978-5-534-17319-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539652>.

6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет».

1. Ефимов С. С. Обзор методов распараллеливания алгоритмов решения некоторых задач вычислительной дискретной математики // МСМ. 2007. №1 (17). URL: <https://cyberleninka.ru/article/n/obzor-metodov-rasparallelvaniya-algoritmov-resheniya-nekotoryh-zadach-vychislitelnoy-diskretnoy-matematiki> (дата обращения: 01.04.2024).
2. Прокопенко Евгения Викторовна, Славолубова Ярославна Викторовна, Ли Сергей Робертович Применение компьютерной математики для решения многомерных задач // Вестник КузГТУ. 2014. №5 (105). URL: <https://cyberleninka.ru/article/n/primenenie-kompyuternoy-matematiki-dlya-resheniya-mnogomernyh-zadach> (дата обращения: 01.04.2024).
3. Митрофанов Алексей Сергеевич, Оверченко Вероника Андреевна Автоматизация решения задачи о составлении рациона // Научный журнал. 2017. №4 (17). URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-resheniya-zadachi-o-sostavlenii-ratsiona> (дата обращения: 01.04.2024).

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7. Материально-техническое обеспечение дисциплины

Для обеспечения дисциплины используется материально-техническая база образовательного учреждения:

- для лекций: учебные аудитории, оснащённые доской, компьютером или ноутбуком, проектором (стационарным или переносным) для демонстрации учебных материалов.

Состав программного обеспечения:

1. Windows
2. Microsoft Office
3. Kaspersky Endpoint Security

- для практических занятий: компьютерный класс или лаборатория, оснащённые доской, компьютером или ноутбуком для преподавателя, компьютерами для обучающихся, проектором (стационарным или переносным) для демонстрации учебных материалов.

Состав программного обеспечения:

1. Windows
2. Microsoft Office
3. Microsoft Visual Professional 2019
4. Mozilla Firefox
5. Kaspersky Endpoint Security

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.
- для глухих и слабослышащих: в печатной форме, в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1 Планы практических занятий

Тема 1. Классы и их структура в языке C++.

Задания:

1. Изучить разделы темы.
Структуры и классы языка C++. Классы и объекты. Заголовочные файлы классов. Файлы реализации классов. Члены класса – методы и поля. Конструкторы классов. Деструкторы классов.

Указания по выполнению заданий:

1. Ответить на контрольные вопросы
 - Структуры и классы языка C++.
 - Классы и объекты.
 - Заголовочные файлы классов.
 - Файлы реализации классов.

- Члены класса – методы и поля.
- Конструкторы классов.
- Деструкторы классов.

Тема 2. Свойства членов класса.

Задания:

1. Изучить разделы темы.
Типы защищённости членов класса. Обеспечение доступа к защищённым членам класса. Статические поля данных и методы классов. Встроенные методы классов. Объекты как параметры функций.

Указания по выполнению заданий:

1. Ответить на контрольные вопросы
 - Типы защищённости членов класса.
 - Обеспечение доступа к защищённым членам класса.
 - Статические поля данных и методы классов.
 - Встроенные методы классов.
 - Объекты как параметры функций.

Тема 3. Доступ операторов и функций к членам класса.

Задания:

1. Изучить разделы темы.
Операторные функции – члены класса. Операторные функции – не члены класса. Перегрузка операторов в классе. Функции – друзья класса. Функции потокового ввода и вывода объектов.

Указания по выполнению заданий:

1. Ответить на контрольные вопросы
 - Операторные функции – члены класса.
 - Операторные функции – не члены класса.
 - Перегрузка операторов в классе.
 - Функции – друзья класса.
 - Функции потокового ввода и вывода объектов.

Тема 4. Наследование и связывание классов. Виртуальные методы.

Задания:

1. Изучить разделы темы.
Принципы объектно-ориентированного программирования. Отношения классов. Типы наследования классов. Доступность членов класса в классах-наследниках. Раннее и позднее связывание классов. Сравнение статического и динамического связывания классов. Виртуальные методы классов.

Указания по выполнению заданий:

1. Ответить на контрольные вопросы
 - Принципы объектно-ориентированного программирования.
 - Отношения классов.
 - Типы наследования классов.
 - Доступность членов класса в классах-наследниках.
 - Раннее и позднее связывание классов.

- Сравнение статического и динамического связывания классов.
- Виртуальные методы классов.

Тема 5. Абстрактные классы.

Задания:

1. Изучить разделы темы.
Понятие абстрактных классов, основные правила построения. Применение абстрактных классов.

Указания по выполнению заданий:

1. Ответить на контрольные вопросы
 - Понятие абстрактных классов.
 - Основные правила построения абстрактных классов.
 - Применение абстрактных классов.

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Дисциплина «Разработка алгоритмов и программных средств в задачах МО» реализуется на факультете информационных систем и безопасности кафедрой Фундаментальной и прикладной математики.

Цель дисциплины: обучение студентов программированию типовых задач математики и обработки информации с использованием современных программных средств

Задачи дисциплины: обеспечить овладение будущими специалистами методами создания эффективных алгоритмов и программ с использованием современных структур данных, программной документацией и способами оценки результатов работы программ.

Дисциплина направлена на формирование следующих компетенций:

- ПК-3. Способен осуществлять поиск, изучение и разработку новых теоретических или практических проблем, сведений, относящихся к решению текущих научных исследований, производственных задач; в информационных средах находить, создавать основные элементы будущих математических структур или конструктивных математических моделей.

В результате освоения дисциплины обучающийся должен:

Знать: конструирование средствами используемого языка программирования новых типов данных, соответствующих специфике решаемой задачи.

Уметь: ставить задачу; эффективно выбирать структуры данных и представления их в ЭВМ в зависимости от решаемой задачи и доступных вычислительных ресурсов.

Владеть: проектированием эффективных алгоритмов обработки информационных структур; созданием программных модулей, реализующих модели информационных структур; обработкой на ЭВМ данных различной структуры, используемых в современных задачах проблемного и системного программирования.

По дисциплине предусмотрена промежуточная аттестация в форме зачета с оценкой.

Общая трудоемкость освоения дисциплины составляет 4 зачетные единицы.

ЛИСТ ИЗМЕНЕНИЙ

№	Текст актуализации или прилагаемый к РПД документ, содержащий изменения	Дата	№ протокола
1	Актуализация основных и дополнительных источников литературы	19.12.2025	5