

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 27.08.2025

Рабочая программа дисциплины

Алгоритмы и структуры данных

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.08 Алгоритмы и структуры данных относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-4: Способен проектировать программное обеспечение	<p>ПК-4.1: Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода</p> <p>ПК-4.2: Формализует задачу ИТ отрасли в язык естественнонаучных дисциплин</p> <p>ПК-4.3: Осуществляет обоснованный выбор методов и алгоритмов для программной реализации формальной математической модели</p> <p>ПК-4.4: Разрабатывает программный код с использованием языков программирования</p> <p>ПК-4.5: Оформляет программный код в соответствии с установленными требованиями</p> <p>ПК-4.6: Работает с системой управления версиями программного кода</p> <p>ПК-4.7: Проверяет и отлаживает программный код</p>	<p>ПК-4.1: ЗНАТЬ основные идеи, лежащие в основе современных языков программирования и основные составляющие современного языка программирования C++;</p> <p>общие принципы использования вычислительной техники для решения прикладных задач.</p> <p>ПК-4.2: УМЕТЬ выполнять реализацию основных структур данных; разрабатывать и отлаживать программы средней сложности в составе малой группы;</p> <p>ПК-4.3: ЗНАТЬ технологии разработки программного обеспечения (структурное, модульное и объектно-ориентированное программирование) и способы их выражения в языке программирования C++;</p> <p>ПК-4.4: УМЕТЬ использовать существующие реализации</p>	Проект	<p>Экзамен: Контрольные вопросы</p> <p>Зачёт: Контрольные вопросы</p>

		<p>основных структур данных из стандартной библиотеки языка программирования C++.</p> <p>ПК-4.5: УМЕТЬ разрабатывать и отлаживать программы средней сложности с использованием языка программирования C++ и технологии объектно-ориентированного программирования;</p> <p>ПК-4.6: Уметь использовать систему контроля версий GIT</p> <p>ПК-4.7: Уметь тестировать программу с помощью библиотеки Google Test</p>		
--	--	--	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	7
Часов по учебному плану	252
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	64
- занятия семинарского типа (практические занятия / лабораторные работы)	128
- КСР	3
самостоятельная работа	21
Промежуточная аттестация	36
	Экзамен, Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе	
		Контактная работа (работа во взаимодействии с преподавателем),	Самостоятельная работа

		часы из них			обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/ лабора торные работы), часы	Всего	
Введение в предмет. Структура и содержание курса. Литература.	2	2	0	2	0
Введение в вычислительную сложность алгоритмов	12	4	8	12	0
Структуры данных и структуры хранения	7	2	4	6	1
Множества и битовые поля	7	2	4	6	1
Работа с векторами и матрицами в C++	13	4	8	12	1
Динамические структуры данных. Стек	7	2	4	6	1
Постфиксная форма арифметических выражений	7	2	4	6	1
Динамическое распределение памяти	13	4	8	12	1
Динамическая структура данных очередь	7	2	4	6	1
Списковые структуры хранения	12	3	8	11	1
Списки в динамической памяти	12	3	8	11	1
Представление полиномов в виде структур данных	7	2	4	6	1
Введение в трансляторы	7	2	4	6	1
Организация доступа по имени. Таблицы	7	2	4	6	1
Неупорядоченные таблицы на массиве и списке	7	2	4	6	1
Упорядоченные таблицы на массиве	7	2	4	6	1
Деревья поиска	7	2	4	6	1
Сбалансированные деревья поиска	7	2	4	6	1
Хеш-таблицы	13	4	8	12	1
Список с пропусками	7	2	4	6	1
Приоритетные очереди и d-кучи	13	4	8	12	1
Работа с текстом	13	4	8	12	1
Работа с геометрическими объектами	7	2	4	6	1
Инициализация в современном C++	6	2	4	6	0
Сводка по курсу	6	2	4	6	0
Аттестация	36				
КСР	3			3	
Итого	252	64	128	195	21

Содержание разделов и тем дисциплины

Цели и задачи дисциплины:

Подготовка высококвалифицированных разработчиков сложных программных систем моделирования объектов и явлений реального мира, управления экономико-социальными и производственными процессами, а также решения других задач автоматизации, научных исследований и проектирования на основе применения современной вычислительной техники.

Содержание разделов и тем:

1. Введение в предмет. Структура и содержание курса. Литература.
2. Введение в вычислительную сложность алгоритмов
3. Структуры данных и структуры хранения
4. Множества и битовые поля
5. Работа с векторами и матрицами в C++
6. Динамические структуры данных. Стек
7. Постфиксная форма арифметических выражений
8. Динамическое распределение памяти
9. Динамическая структура данных очередь
10. Списковые структуры хранения
11. Списки в динамической памяти
12. Представление полиномов в виде структур данных
13. Введение в трансляторы
14. Организация доступа по имени. Таблицы
15. Неупорядоченные таблицы на массиве и списке
16. Упорядоченные таблицы на массиве
17. Деревья поиска
18. Сбалансированные деревья поиска
19. Хеш-таблицы
20. Список с пропусками
21. Приоритетные очереди и d-кучи
22. Работа с текстом
23. Работа с геометрическими объектами
24. Инициализация в современном C++
25. Сводка по курсу

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:

Электронные курсы, созданные в системе электронного обучения ННГУ:

Алгоритмы и структуры данных, <https://e-learning.unn.ru/course/view.php?id=1617>, <https://e-learning.unn.ru/enrol/index.php?id=1618>.

Иные учебно-методические материалы:

–Гергель В.П. Рабочие материалы по курсу «ЭВМ и программирование».

–Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Лабораторный практикум (по программе "Алгоритмы и структуры данных") Учебно-методическое пособие.

http://www.unn.ru/books/met_files/Pract_ADS.pdf

–Технопарк Mail.ru Group. Курс "Алгоритмы и структуры данных".

<http://www.intuit.ru/studies/courses/3496/738/info>

–Седжвик Р. Курс "Алгоритмы на C++". <http://www.intuit.ru/studies/courses/12181/1174/info>

–Алексеев В., Таланов В. Курс "Структуры данных и модели вычислений".

<http://www.intuit.ru/studies/courses/100/100/info>

–Meyer B. Курс "Инструменты, алгоритмы и структуры данных".
<http://www.intuit.ru/studies/courses/683/539/info>

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Проект) для оценки сформированности компетенции ПК-4:

Студентам предлагаются следующие темы проектов (лабораторных работ):

1. Лабораторная работа 1. Структура хранения множеств.

Цель лабораторной работы – разработка структуры данных для хранения множеств с использованием битовых полей, а также освоение таких инструментов разработки программного обеспечения, как система контроля версий Git и фреймворк для разработки автоматических тестов GoogleTest.

Перед выполнением работы студенты получают данный проект-шаблон, содержащий следующее:

- Интерфейсы классов битового поля и множества (h-файлы)
- Готовый набор тестов для каждого из указанных классов
- Пример использования класса битового поля и множества для решения задачи поиска простых чисел с помощью алгоритма "Решето Эратосфена"

Выполнение работы предполагает решение следующих задач:

- Реализация класса битового поля TBitField согласно заданному интерфейсу.
- Реализация класса множества TSet согласно заданному интерфейсу.
- Обеспечение работоспособности тестов и примера использования.
- Реализация нескольких простых тестов на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

1. Лабораторная работа 2. Структуры хранения матриц специального вида.

В рамках лабораторной работы ставится задача создания программных средств, поддерживающих эффективное хранение матриц специального вида (верхнетреугольных) и выполнение основных операций над ними:

- сложение/вычитание;
- копирование;
- сравнение.

В процессе выполнения лабораторной работы требуется использовать систему контроля версий Git и фреймворк для разработки автоматических тестов GoogleTest.

Перед выполнением работы студенты получают данный проект-шаблон, содержащий следующее:

- Интерфейсы классов Вектор и Матрица (h-файл)
- Начальный набор готовых тестов для каждого из указанных классов.
- Набор заготовок тестов для каждого из указанных классов.

- Тестовый пример использования класса Матрица

Выполнение работы предполагает решение следующих задач:

- Реализация методов шаблонного класса TVector согласно заданному интерфейсу.
- Реализация методов шаблонного класса TMatrix согласно заданному интерфейсу.
- Обеспечение работоспособности тестов и примера использования.
- Реализация заготовок тестов, покрывающих все методы классов TVector и TMatrix.
- Модификация примера использования в тестовое приложение, позволяющее задавать матрицы и осуществлять основные операции над ними.

1. Лабораторная работа 3. Вычисление арифметических выражений.

Цель данной работы – разработка структуры данных Стек и ее использование для расчета арифметических выражений с использованием обратной польской записи (постфиксной формы).

Выполнение работы предполагает решение следующих задач:

- Разработка интерфейса шаблонного класса TStack.
- Реализация методов шаблонного класса TStack.
- Разработка интерфейса класса TPostfix для работы с постфиксной формой.
- Реализация методов класса TPostfix.
- Разработка и реализация тестов для классов TStack и TPostfix на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

1. Лабораторная работа 4. Обслуживание процессором ЭВМ очереди заданий.

Цель данной работы – разработка структуры данных Очередь и ее использование для имитационного моделирования системы обслуживания потока заданий на ЭВМ.

Выполнение работы предполагает решение следующих задач:

- Разработка интерфейса шаблонного класса TQueue.
- Реализация методов шаблонного класса TQueue.
- Разработка интерфейсов и реализация методов остальных необходимых классов.
- Разработка и реализация тестов для классов TQueue и остальных разработанных классов на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

1. Лабораторная работа 5. Система для арифметических действий над полиномами.

В рамках лабораторной работы ставится задача создания программных средств, поддерживающих эффективное представление полиномов и выполнение следующих операций над ними:

- ввод полинома;
- организация хранения полинома;
- удаление введенного ранее полинома;
- копирование полинома;
- сложение двух полиномов;
- вычисление значения полинома при заданных значениях переменных;
- вывод.

Выполнение работы предполагает решение следующих задач:

- Разработка интерфейсов и реализация методов класса TPolinom.
- Разработка интерфейсов и реализация методов остальных необходимых классов.

- Разработка и реализация тестов для классов TPolinom и остальных разработанных классов на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

1. Лабораторная работа 6. Редактирование текстов.

В рамках лабораторной работы ставится задача разработки учебного редактора текстов с поддержкой следующих операций:

- выбор текста для редактирования (или создание нового текста);
- демонстрация текста на экране дисплея;
- поддержка средств указания элементов (уровней) текста;
- вставка, удаление и замена строк текста;
- запись подготовленного текста в файл.

Выполнение работы предполагает решение следующих задач:

- Разработка интерфейсов и реализация методов класса TText.
- Разработка интерфейсов и реализация методов остальных необходимых классов.
- Разработка и реализация тестов для классов TText и остальных разработанных классов на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

1. Лабораторная работа 7. Структуры хранения геометрических объектов.

В рамках лабораторной работы ставится задача разработки программного комплекса для представления, редактирования и отображения на экране дисплея чертежей, образуемых из ограниченного набора геометрических элементов (точек, линий, окружностей и т.п.).

Разрабатываемый комплекс должен обеспечивать:

- представление чертежей в памяти ЭВМ;
- демонстрацию хранимых чертежей на экране дисплея;
- редактирование чертежей (вставку и удаление линий);
- запись (чтение) данных о чертежах в файлах.

Выполнение работы предполагает решение следующих задач:

- Разработка интерфейсов и реализация методов класса TPlex.
- Разработка интерфейсов и реализация методов остальных необходимых классов.
- Разработка и реализация тестов для классов TPlex и остальных разработанных классов на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

1. Лабораторная работа 8. Сравнительная характеристика способов организации таблиц.

В рамках данной лабораторной работы ставится задача создания программных средств, поддерживающих табличные динамические структуры данных (таблицы) и базовые операции над ними:

- поиск записи;
- вставка записи (без дублирования);
- удаление записи.

В лабораторной работе предлагается реализовать следующие типы таблиц:

- просмотрные (на массиве или линейном списке);
- упорядоченные (на массиве или на основе деревьев поиска);
- хеш-таблицы.

Выполнение работы предполагает решение следующих задач:

- Разработка интерфейсов и реализация методов классов TTable, TLineTable, TSortTable, THashTable.
- Разработка интерфейсов и реализация методов остальных необходимых классов.
- Разработка и реализация тестов для классов TTable, TLineTable, TSortTable, THashTable и остальных разработанных классов на базе GoogleTest.
- Публикация исходных кодов в личном репозитории на GitHub.

Критерии оценивания (оценочное средство - Проект)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Программа и результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, программа работает некорректно, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатор достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но	Продемонстрированы все основные умения. Решены все основные задачи с отдельным и несущест	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов

			не в полном объеме	в полном объеме, но некоторые с недочетами	некоторые с недочетами	енными недочетам и, выполнены все задания в полном объеме	
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. Структуры данных и структуры хранения.
2. Линейные структуры данных.

3. Динамические структуры данных.
4. Структуры хранения динамических структур типа стек.
5. Структуры хранения динамических структур типа очередь.
6. Сравнение структур хранения линейных и динамических структур.
7. Статическое и динамическое распределение памяти.
8. Управление памятью путем перепакровки структур хранения.
9. Структура хранения нескольких стеков в общей памяти.
10. Роль гипотез о росте структур при разработке систем управления памятью путем перепакровки.
11. Оценка параметров модели в ходе выполнения программ (адаптация).
12. Линейный список.
13. Способы реализации списков на языках высокого уровня.
14. Реализация структуры хранения нескольких стеков с использованием списков на языке высокого уровня.
15. Сравнение непрерывной и списковой структур хранения.
16. Динамическое распределение памяти в языке C/C++ (выделение и освобождение памяти).
17. Реализация стека с использованием динамически распределяемой памяти.
18. Преобразование арифметических выражений в обратную польскую форму записи.
19. Система для арифметических действий над полиномами (представление полиномов, управление памятью, выполнение операций).
20. Представление текста связным списком.
21. Алгоритм обхода иерархического списка.
22. Плексы как представление рисунков, состоящих из точек и соединяющих их отрезков.
23. Алгоритм обхода плекса.
24. Плекс, как представление арифметического выражения.
25. Организация доступа по имени. Таблицы. Поиск по ключу (просмотр и двоичный поиск).
26. Представление таблиц с использованием деревьев поиска.
27. Деревья поиска. Алгоритмы обхода.
28. Деревья поиска. Алгоритмы поиска и вставки.
29. Деревья поиска. Алгоритм удаления.
30. Сбалансированные и идеально сбалансированные деревья поиска. Общая схема балансировки при вставке.
31. Таблицы с вычислимым входом. Запись и поиск при переполнении (способ открытого перемешивания).

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
превосходно	Студент дал развернутый ответ на все вопросы и при этом продемонстрировал знание дополнительного материала.
отлично	Студент дал развернутый ответ на все вопросы.
очень хорошо	Студент дал ответ на все вопросы, возможно с незначительными недочетами.
хорошо	Студент ответил на большую часть вопросов с незначительными недочетами.
удовлетворительно	Студент ответил на большую часть вопросов с существенными недочетами.
неудовлетворительно	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.
плохо	Отсутствие знаний материала, отсутствует способность решения стандартных задач.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. Структуры данных и структуры хранения.
2. Линейные структуры данных.
3. Динамические структуры данных.
4. Структуры хранения динамических структур типа стек.
5. Структуры хранения динамических структур типа очередь.
6. Сравнение структур хранения линейных и динамических структур.
7. Статическое и динамическое распределение памяти.
8. Управление памятью путем перепакетки структур хранения.
9. Структура хранения нескольких стеков в общей памяти.
10. Роль гипотез о росте структур при разработке систем управления памятью путем перепакетки.
11. Оценка параметров модели в ходе выполнения программ (адаптация).
12. Линейный список.
13. Способы реализации списков на языках высокого уровня.
14. Реализация структуры хранения нескольких стеков с использованием списков на языке высокого уровня.
15. Сравнение непрерывной и списковой структур хранения.
16. Динамическое распределение памяти в языке C/C++ (выделение и освобождение памяти).
17. Реализация стека с использованием динамически распределяемой памяти.

18. Преобразование арифметических выражений в обратную польскую форму записи.
19. Система для арифметических действий над полиномами (представление полиномов, управление памятью, выполнение операций).
20. Представление текста связным списком.
21. Алгоритм обхода иерархического списка.
22. Плексы как представление рисунков, состоящих из точек и соединяющих их отрезков.
23. Алгоритм обхода плекса.
24. Плекс, как представление арифметического выражения.
25. Организация доступа по имени. Таблицы. Поиск по ключу (просмотр и двоичный поиск).
26. Представление таблиц с использованием деревьев поиска.
27. Деревья поиска. Алгоритмы обхода.
28. Деревья поиска. Алгоритмы поиска и вставки.
29. Деревья поиска. Алгоритм удаления.
30. Сбалансированные и идеально сбалансированные деревья поиска. Общая схема балансировки при вставке.
31. Таблицы с вычислимым входом. Запись и поиск при переполнении (способ открытого перемешивания).

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент дал развернутый ответ на все вопросы без существенных ошибок.
не зачтено	При ответе студент допускает грубые ошибки в основном материале.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Лабораторный практикум (по программе «Алгоритмы и структуры данных») : учебно-методическое пособие / И. В. Барышева, И. Б. Мееров, А. В. Сысоев, Н. В. Шестакова ; ННГУ им. Н. И. Лобачевского, Институт информационных технологий, математики и механики. - Нижний Новгород : Изд-во ННГУ, 2017. - 105 с. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=822925&idb=0>.
2. Подбельский В.В. Язык Си++ : учебное пособие / Подбельский В.В. - Москва : Финансы и статистика, 2022. - 560 с. - ISBN 978-5-00184-082-4., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=869552&idb=0>.

Дополнительная литература:

1. Инструменты, алгоритмы и структуры данных / Мейер Б. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662958&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционные системы семейства MicrosoftWindows, лицензия по подписке Microsoft Imagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства Microsoft Visual Studio, лицензия по подписке Microsoft Imagine.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Сысоев Александр Владимирович, кандидат технических наук.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 25.06.2025, протокол № Протокол №11.