

Рабочая программа утверждена  
 Решением Ученого совета  
 ВФ НИТУ «МИСИС»  
 От «28» мая 2026г.  
 Протокол №7-26

## Рабочая программа дисциплины (модуля)

# Основы алгоритмизации и программирования

|                         |                                            |
|-------------------------|--------------------------------------------|
| Закреплена за кафедрой  | Базовых дисциплин                          |
| Направление подготовки  | 27.03.04 Управление в технических системах |
| Профиль                 | Информационные технологии в управлении     |
| Квалификация            | <b>Бакалавр</b>                            |
| Форма обучения          | <b>очная</b>                               |
| Общая трудоемкость      | <b>7 ЗЕТ</b>                               |
| Часов по учебному плану | 252 Формы контроля в семестрах:            |
| в том числе:            | экзамен 1 зачет с оценкой 2                |
| аудиторные занятия      | 90                                         |
| самостоятельная работа  | 131                                        |
| часов на контроль       | 27                                         |

**Распределение часов дисциплины по семестрам**

| Семестр<br>(<Курс>.<Семестр на<br>курсе>) | 1 (1.1) |     | 2 (1.2) |     | Итого |     |
|-------------------------------------------|---------|-----|---------|-----|-------|-----|
| Неделя                                    | 19      |     | 19      |     |       |     |
| Вид занятий                               | УП      | РП  | УП      | РП  | УП    | РП  |
| Лекции                                    | 36      | 36  | 18      | 18  | 54    | 54  |
| Практические                              | 18      | 18  | 18      | 18  | 36    | 36  |
| Контроль<br>самостоятельной<br>работы     | 2       | 2   | 2       | 2   | 4     | 4   |
| Итого ауд.                                | 54      | 54  | 36      | 36  | 90    | 90  |
| Контактная работа                         | 56      | 56  | 38      | 38  | 94    | 94  |
| Сам. работа                               | 61      | 61  | 70      | 70  | 131   | 131 |
| Часы на контроль                          | 27      | 27  |         |     | 27    | 27  |
| Итого                                     | 144     | 144 | 108     | 108 | 252   | 252 |

Программу составил(и):

*кхн, Доц., Будруев Андрей Владимирович*

---

---

Рабочая программа

**Основы алгоритмизации и программирования**

Разработана в соответствии с ОС ВО:

Самостоятельно устанавливаемый образовательный стандарт высшего образования - бакалавриат Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСИС» по направлению подготовки 27.03.04 Управление в технических системах (приказ от 02.04.2021 г. № 119 о.в.)

Составлена на основании учебного плана:

27.03.04 Управление в технических системах, УТС-26.plx Информационные технологии в управлении, утвержденного Ученым советом ВФ НИТУ "МИСИС" 21.11.2025, протокол № 3-25

Рабочая программа одобрена на заседании кафедры

**Базовых дисциплин**

Протокол от 14.05.2026 г., №9

Зав. кафедрой Уснунц-Кригер Т.Н. \_\_\_\_\_

### 1. ЦЕЛИ ОСВОЕНИЯ

|     |                                                                                                                                                                                  |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.1 | Цель дисциплины ознакомить студентов с основными принципами алгоритмизации, а также сформировать практические навыки в области процедурных и объектных методов программирования. |
| 1.2 | Задачи:                                                                                                                                                                          |
| 1.3 | - знать общие принципы построения алгоритмов                                                                                                                                     |
| 1.4 | - знать основные элементы процедурного языка программирования, структуру программы, операции, управляющие структуры, структуры данных                                            |
| 1.5 | - знать объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов                                                                       |
| 1.6 | - проведение вычислений и преобразований, связанных с этими объектами                                                                                                            |
| 1.7 | - уметь использовать языки программирования и строить логически правильные и эффективные программы                                                                               |
| 1.8 | - уметь решать конструктивно-исследовательских задач и использование основных методов применения алгоритмов на практике                                                          |

### 2. МЕСТО В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

|                   |                                                                                                                       |      |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|------|
| Цикл (раздел) ОП: |                                                                                                                       | Б1.О |
| 2.1               | <b>Требования к предварительной подготовке обучающегося:</b>                                                          |      |
| 2.1.1             | Математика                                                                                                            |      |
| 2.1.2             | Информатика                                                                                                           |      |
| 2.2               | <b>Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b> |      |
| 2.2.1             | Вычислительные машины, системы и сети                                                                                 |      |
| 2.2.2             | Производственная практика по получению профессиональных умений и опыта профессиональной деятельности                  |      |
| 2.2.3             | Технология программирования                                                                                           |      |
| 2.2.4             | Системы управления базами данных                                                                                      |      |
| 2.2.5             | Управление данными                                                                                                    |      |
| 2.2.6             | Прикладное программирование                                                                                           |      |
| 2.2.7             | Специальное программное обеспечение                                                                                   |      |

### 3. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ФОРМИРУЕМЫМИ КОМПЕТЕНЦИЯМИ

**ОПК-6:** Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности, разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности, проектировать и разрабатывать продукцию, процессы и системы в условиях неопределенности и альтернативных решений в междисциплинарных областях

**ОПК-6.2:** Применяет алгоритмы и программы, современные информационные технологии в сфере своей профессиональной деятельности

**Знать:**

ОПК-6.2-31 методологию описания классов и их взаимодействий

ОПК-6.2-32 способы создания различных типов объектов

ОПК-6.2-33 технологию использования и ввода (вывода) данных различных типов

**Уметь:**

ОПК-6.2-У1 разрабатывать алгоритмы решения прикладных задач на основе типовых структур алгоритмов

ОПК-6.2-У2 разрабатывать прикладные программные продукты с помощью современных средств разработки и языков программирования с применением современных информационных технологий

**Владеть:**

ОПК-6.2-В1 культурой алгоритмического мышления и навыками использования типовых алгоритмов для решения задач предметной области

ОПК-6.2-В2 навыками работы с современными инструментариями разработки прикладных программных продуктов на базе современных языков программирования

| 4. СТРУКТУРА И СОДЕРЖАНИЕ |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                |       |             |                                    |            |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-------|-------------|------------------------------------|------------|
| Код занятия               | Наименование разделов и тем /вид занятия/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Семестр / Курс | Часов | Компетенции | Литература и эл. ресурсы           | Примечание |
|                           | <b>Раздел 1. Основные понятия алгоритмизации</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                |       |             |                                    |            |
| 1.1                       | Введение /Лек/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 1              | 1     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 1.2                       | Основные понятия алгоритмизации. Понятие алгоритма. Свойства алгоритма. Схема решения задач на ЭВМ. Формы записи алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. /Лек/                                                                                                                                                                                                                                                                                                                                                 | 1              | 2     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 1.3                       | Консольный ввод и вывод строк и чисел /Пр/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 1              | 2     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 1.4                       | Подготовка к практическому занятию /Ср/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1              | 10    | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
|                           | <b>Раздел 2. Логические основы алгоритмизации.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |       |             |                                    |            |
| 2.1                       | Основные базовые и структурированные типы данных, их характеристика. /Лек/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 1              | 2     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 2.2                       | Работа с целыми числами и вещественными числами, символами и логическими переменными /Пр/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1              | 2     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 2.3                       | Подготовка к практическому занятию /Ср/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1              | 10    | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
|                           | <b>Раздел 3. Языки программирования.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                |       |             |                                    |            |
| 3.1                       | Основные понятия программирования. Алгоритм. Основные свойства алгоритма. Алгоритмизация задач. Типовые структуры алгоритма. Языки программирования. Структурная и объектная декомпозиция. Основные компоненты парадигмы ООП (инкапсуляция, наследование, полиморфизм). Принципы ООП (абстрагирование, ограничение доступа, модульность, иерархичность, типизация, параллелизм, устойчивость). Этапы разработки ПО на основе ОПП (анализ, проектирование, эволюция, модификация). Основные понятия платформы .NET Framework (CTS, CLS, FCL, CLR, CIL). Архитектура, состав, назначение, функции. /Лек/ | 1              | 4     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 3.2                       | Основные элементы языка. Основы C# (идентификаторы, переменные, выражения, операции, операторы, атрибуты, спецификаторы). Типы данных (значимые, ссылочные, преобразование, упаковка / распаковка). /Лек/                                                                                                                                                                                                                                                                                                                                                                                              | 1              | 2     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |
| 3.3                       | Стандартные и пользовательские типы данных (символьные, логические, целые, вещественные, структуры, перечисления, массивы, строки, классы, свойства, методы). /Лек/                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1              | 2     | ОПК-6.2     | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |            |

|                                                                      |                                                                                                                                            |   |    |         |                                    |  |
|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|---|----|---------|------------------------------------|--|
| 3.4                                                                  | Одномерные массивы. Двухмерные массивы. Ступенчатые (“зубчатые”) массивы. Типовые алгоритмы обработки массивов. Массивы как объекты. /Лек/ | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 3.5                                                                  | Функции /Лек/                                                                                                                              | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 3.6                                                                  | Организация ввода-вывода данных. Работа с файлами. /Лек/                                                                                   | 1 | 4  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 3.7                                                                  | Эволюция языков программирования. Классификация языков программирования. Интегрированная среда программирования. /Лек/                     | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 3.8                                                                  | Работа со строками /Пр/                                                                                                                    | 1 | 4  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 3.9                                                                  | Подготовка к практическому занятию /Ср/                                                                                                    | 1 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 4. Методы программирования.</b>                            |                                                                                                                                            |   |    |         |                                    |  |
| 4.1                                                                  | Основные принципы объектно-ориентированного программирования (ООП) /Лек/                                                                   | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 4.2                                                                  | Структурный, модульный, объектно-ориентированный подходы. Достоинства и недостатки методов                                                 | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 4.3                                                                  | Визуальное событийно-управляемое программирование /Лек/                                                                                    | 1 | 3  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 4.4                                                                  | Разработка оконного приложения /Лек/                                                                                                       | 1 | 3  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 4.5                                                                  | Массивы /Пр/                                                                                                                               | 1 | 6  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 4.6                                                                  | Подготовка к практическому занятию /Ср/                                                                                                    | 1 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 5. Общие принципы разработки программного обеспечения.</b> |                                                                                                                                            |   |    |         |                                    |  |
| 5.1                                                                  | Жизненный цикл программного обеспечения. Типы приложений. Консольные приложения. /Лек/                                                     | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 5.2                                                                  | Функции /Пр/                                                                                                                               | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 5.3                                                                  | Подготовка к практическому занятию /Ср/                                                                                                    | 1 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 6. Основы решения алгоритмической задачи</b>               |                                                                                                                                            |   |    |         |                                    |  |
| 6.1                                                                  | Делимость, НОД, НОК /Лек/                                                                                                                  | 1 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 6.2                                                                  | Рекурсия /Пр/                                                                                                                              | 1 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 6.3                                                                  | Подготовка к практическому занятию /Ср/                                                                                                    | 1 | 11 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 7. Основы анализа эффективности алгоритмов</b>             |                                                                                                                                            |   |    |         |                                    |  |

|                                                    |                                                                                                       |   |    |         |                                    |  |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------|---|----|---------|------------------------------------|--|
| 7.1                                                | Асимптотические обозначения и как их применяют для анализа эффективности алгоритмов. /Лек/            | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 7.2                                                | Временная сложность и пространственные сложности алгоритмов /Лек/                                     | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 7.3                                                | Асимптотические обозначения: $O$ (о большое), $\Omega$ (омега большое), $\Theta$ (тета большое) /Лек/ | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 7.4                                                | LINQ /Пр/                                                                                             | 2 | 9  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 7.5                                                | Подготовка к практическому занятию /Ср/                                                               | 2 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 8. Рекурсивные функции</b>               |                                                                                                       |   |    |         |                                    |  |
| 8.1                                                | Системный стек, и какие ошибки могут возникнуть при использовании рекурсивных алгоритмов /Лек/        | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 8.2                                                | Анализ сложности рекурсивных алгоритмов /Лек/                                                         | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 8.3                                                | Алгоритма вычисления факториала числа /Лек/                                                           | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 8.4                                                | Сортировка /Пр/                                                                                       | 2 | 9  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 8.5                                                | Подготовка к практическому занятию /Ср/                                                               | 2 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 9. Алгоритмы сортировки</b>              |                                                                                                       |   |    |         |                                    |  |
| 9.1                                                | Основные задачи сортировки. Устойчивая и неустойчивая сортировки /Лек/                                | 2 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 9.2                                                | Алгоритм сортировки слиянием. /Лек/                                                                   | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 9.3                                                | Подготовка к практическому занятию /Ср/                                                               | 2 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 10. Метод «грубой силы»</b>              |                                                                                                       |   |    |         |                                    |  |
| 10.1                                               | Алгоритма пузырьковой сортировки и ее временная сложность /Лек/                                       | 2 | 2  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 10.2                                               | Подготовка к практическому занятию /Ср/                                                               | 2 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 11. Метод «разделяй и властвуй»</b>      |                                                                                                       |   |    |         |                                    |  |
| 11.1                                               | Алгоритм бинарного поиска /Лек/                                                                       | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 11.2                                               | Метод быстрой сортировки /Лек/                                                                        | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 11.3                                               | Подготовка к практическому занятию /Ср/                                                               | 2 | 10 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 12. Графы. Типы представления графов</b> |                                                                                                       |   |    |         |                                    |  |
| 12.1                                               | Матрица смежности и списки смежных вершин /Лек/                                                       | 2 | 1  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |

|                                      |                                                                           |   |     |         |                                    |  |
|--------------------------------------|---------------------------------------------------------------------------|---|-----|---------|------------------------------------|--|
| 12.2                                 | Взвешенные графы, особенности структуры хранения данных /Лек/             | 2 | 1   | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 12.3                                 | Алгоритм «Поиск в глубину» и «Поиска в ширину» /Лек/                      | 2 | 1   | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 12.4                                 | Повторение лекционного материала. Подготовка к практическому занятию /Ср/ | 2 | 10  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| <b>Раздел 13. «Жадные алгоритмы»</b> |                                                                           |   |     |         |                                    |  |
| 13.1                                 | Минимальные остовные деревья. Алгоритм Крускала. Алгоритм Прима /Лек/     | 2 | 1   | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 13.2                                 | Алгоритма Дейкстры /Лек/                                                  | 2 | 0,5 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 13.3                                 | Алгоритм кодировки Хаффмана /Лек/                                         | 2 | 0,5 | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |
| 13.4                                 | Повторение лекционного материала. Подготовка к практическому занятию /Ср/ | 2 | 10  | ОПК-6.2 | Л1.1 Л1.2<br>Л1.3 Л1.4<br>Э1 Э2 Э3 |  |

## 5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

### 5.1. Вопросы для самостоятельной подготовки к экзамену (зачёту с оценкой)

Перечень вопросов к экзамену - теоретическая часть(семестр 1):

1. Что такое алгоритмизация и алгоритм?
2. На примере алгоритмов наибольшего общего делителя двух целых чисел объясните эффективность работы алгоритмов.
3. Что такое псевдокод?
4. В чем состоит метод Евклида нахождения наибольшего общего делителя?
5. В чем состоит метод генерация последовательности простых чисел (алгоритм Эратосфена)?
6. Расскажите, какие этапы проектирования и анализа алгоритмов существуют и в чем их ключевые особенности?
7. Какие компьютеры называются «машинами фон Неймана»?
8. Какие методы представления алгоритмов вы знаете?
9. Как проводится оценка корректности алгоритма?
10. Чем отличается эффективность алгоритма в наихудшем, наилучшем и среднем случаях?
11. Что влияет на время выполнения алгоритма (программы)?
12. Какие компьютеры называются «машиной с произвольным доступом к памяти»?
13. Как размер входных данных алгоритма влияет на эффективность работы программы?
14. Например «Алгоритма линейного поиска элемента в массиве» рассчитайте количество операций алгоритма в наихудшем, наилучшем и среднем случае.
15. Как проводится анализ сложности алгоритмов?
16. Что такое «асимптотические обозначения» и как их применяют для анализа эффективности алгоритмов?
17. Чем отличаются асимптотические обозначения:  $O$  (о большое),  $\Omega$  (омега большое),  $\Theta$  (тета большое)?
18. Опишите расчет алгоритмической сложности в наихудшем, наилучшем и среднем случае для  $O$ -большого?
19. Опишите расчет алгоритмической сложности в наихудшем, наилучшем и среднем случае для омега большое?
20. Опишите расчет алгоритмической сложности в наихудшем, наилучшем и среднем случае для тета большое?
21. Каковы свойства асимптотических функций и как их используют для расчета алгоритмической сложности?
22. На примере объясните этапы асимптотического анализа?
23. Что такое временная сложность и пространственные сложности алгоритмов?
24. Какие функции называют рекурсивными?
25. На конкретном примере алгоритма расскажите какие типы рекурсии бывают?
26. Что такое системный стек, и какие ошибки могут возникнуть при использовании рекурсивных алгоритмов?
27. Объясните как провести анализ сложности рекурсивных алгоритмов?
28. В чем состоит метод математической индукции в рамках анализ сложности рекурсивных алгоритмов?
29. Рассчитайте временную сложность алгоритма вычисления факториала числа?
30. В чем состоит общий метод решения рекуррентных соотношений?
31. Рассчитайте временную сложность алгоритма бинарного поиска.
32. Рассчитайте временную сложность алгоритма сортировки слиянием.
33. В чем состоят основные задачи сортировки?
34. Что такое устойчивая и неустойчивая сортировки?
35. Чем отличаются алгоритмы сортировок? Какова их вычислительная сложность?
36. Объясните алгоритма пузырьковой сортировки и вычислите ее временную сложность.

37. Что такое «метод грубой силы»?
38. Какие методы уменьшения размера задачи вы знаете?
39. Объясните алгоритм сортировки вставкой и вычислите ее временную сложность.
40. В чем состоит метод «разделяй и властвуй»?
41. Объясните, как работает метод сортировки слиянием?
42. Объясните, как работает метод быстрой сортировки? В чем особенность выбора опорного элемента?
43. Рассчитайте временную сложность алгоритма быстрой сортировки.
44. Проанализируйте временную сложность алгоритма сортировки подсчетом.
45. Дайте понятие структуры данных. Опишите особенности работы с линейными структурами данных.
46. Опишите на примере основные методы работы с массивами.
47. Опишите на примере основные методы работы со строками.
48. Что такое коллекции? Особенности и методы работы со стеком, очередью, словарем. В чем особенность работы с этими структурами данных?
49. Объясните внутреннюю структуру хранения данных в списках. Какие типы списков вы знаете?
50. В чем отличие списков от одномерных массивов? В каких случаях можно использовать списки, одномерные массивы?
51. Опишите особенности структуры данных «очередь с приоритетами».
52. Что такое Хэш-набор? Какие методы работы с этой структурой данных?
53. Что такое обобщенные типы данных? Их реализация.
54. Что такое графы, какие типы графов бывают?
55. Типы представления графов.
56. Что такое матрица смежности и списки смежных вершин? В каких случаях используют эти структуры хранения данных?
57. Взвешенные графы, особенности структуры хранения данных?
58. Что такое пути и циклы? Связанность, ориентированность.
59. Объясните, какие структуры данных называются деревьями, и каковы их свойства
60. Приведите псевдокод алгоритма: «Поиск в глубину» и объясните его работу.
61. Приведите псевдокод алгоритма: «Поиск в ширину» и объясните его работу.
62. Что такое «жадные алгоритмы»? Для чего используются жадные алгоритмы?
63. Для чего в программировании используются остовные (покрывающие) деревья?
64. Что такое минимальные остовные деревья?
65. В чем состоит алгоритм Крускала. Приведите его псевдокод.
66. Расскажите, какие алгоритмы поиска кратчайшего пути в графе вы знаете?
67. В чем состоит алгоритм Прима. Приведите его псевдокод.
68. Объясните работу алгоритма Дейкстры? Приведите его псевдокод.
69. В чем состоит особенность алгоритма Беллмана Форда?
70. Как работает стадия релаксации в алгоритме Беллмана Форда?
71. Как работает алгоритм кодировки Хаффмана? Объясните на конкретном примере.
72. Объясните логику жадных алгоритмов на примере алгоритма подсчет выдачи сдачи.

Перечень вопросов к экзамену - практическая часть(семестр 1):

1. Напишите программу на C #, которая проверяет, является ли введенный номер четным или нечетным.
2. Напишите программу на C #, которая читает с консоли положительное целое число N и печатает сумму первых N членов последовательности Фибоначчи (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377).
3. Напишите программу на C #, которая переводит введенное пользователем количество грамм в соответствующее количество тон, килограмм и грамм.
4. Напишите программу на C #, которая читает с консоли положительное целое число N и печатает сумму первых N членов последовательности Фибоначчи (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377). Однако для вычисления использует не цикл, а рекурсию.
5. Напишите метод на C # в который передается 2 числа и меняющее их содержимое в переменных.
6. Напишите программу, которая в качестве входных данных принимает целое число, а затем выводит на экран все простые числа, которые меньше или равны введенному числу.
7. Напишите программу на C #, которая получает число и отображает сумму всех ее цифр.
8. Напишите программу, которая читает целые числа до тех пор, пока при вводе не встретится 0. Как только ввод прекратится, программа должна сообщить общее количество четных чисел во входных данных (за исключением 0), среднее количество четных чисел, общее количество нечетных чисел во входных данных и среднее значение нечетных чисел.
9. Напишите программу на C #, которая получает число и отображает его в обратном порядке.
10. Напишите программу, которая читает входные символы до тех пор, пока не встретится символ #.
11. Напишите программу на C #, которая считает количество единиц в введенном числе.
12. Воспользуйтесь вложенными циклами, чтобы написать программу, которая выводит на печать следующую фигуру:  
\$  
\$  
\$\$  
\$\$\$  
\$\$\$\$  
\$\$\$\$\$
13. Напишите программу на C #, которая генерирует случайные числа.

14. Напишите программу, которая находит самое большое число из введенных пользователем.
15. Напишите программу на C #, которая принимает рост человека и классифицирует его как Высокий, Низкий или Средний.
16. Напишите программу, которая вычисляет сумму 20 первых целых чисел.
17. Напишите программу на C #, которая предлагает пользователю ввести свой возраст. Затем программа должна отобразить возраст в месяцах.
18. Напишите программу, которая требует от пользователя ввести количество дней, а затем переводит это значение в количество недель и дней. Например, она переводит 18 дней в 2 недели и 4 дня. Отобразите результаты в следующем формате: 18 дней составляют 2 недели и 4 дня.
19. Напишите программу на C #, которая выдает запрос на ввод значений часов и минут и отобразит эти два значения в следующем виде.
20. Напишите программу, которая запрашивает ввод целого числа, а затем печатает все целые числа, начиная (и включая) с этого числа и до числа, большего введенной величины на 10 включительно.
21. В году примерно  $3.156 \times 10^7$  секунд. Напишите программу, которая приглашает ввести возраст в годах, а затем выводит на экран эквивалентное значение в секундах.
22. Напишите программу на C #, которая читает с консоли положительное целое десятичное число и конвертирует его из десятичной в двоичную систему.
23. Напишите программу на C #, которая рассчитывает, с каким количеством нулей заканчивается факториал данного числа.
24. Напишите функцию с именем `min(x,y)`, которая возвращает меньшее из двух значений типа `double`.
25. Напишите программу на C #, которая запрашивает количество секунд в виде целого значения (используйте тип `long` или `ulong`) и затем отображает эквивалентное значение в сутках, часах, минутах и секундах. Для представления количества часов в сутках, количества минут в часе и количества секунд в минуте используйте символические константы.
26. В дюйме 2.54 сантиметра. Напишите программу, которая приглашает вас ввести рост в сантиметрах, после чего выводит на экран этот рост в дюймах.

Перечено вопроов к зачетус оценкой (семестр 2):

1. Что такое алгоритмизация и алгоритм?
2. На примере алгоритмов наибольшего общего делителя двух целых чисел объясните эффективность работы алгоритмов.
3. Что такое псевдокод?
4. В чем состоит метод Евклида нахождения наибольшего общего делителя?
5. В чем состоит метод генерация последовательности простых чисел (алгоритм Эратосфена)?
6. Расскажите, какие этапы проектирования и анализа алгоритмов существуют и в чем их ключевые особенности?
7. Какие компьютеры называются «машинами фон Неймана»?
8. Какие методы представления алгоритмов вы знаете?
9. Как проводится оценка корректности алгоритма?
10. Чем отличается эффективность алгоритма в наихудшем, наилучшем и среднем случаях?
11. Что влияет на время выполнения алгоритма (программы)?
12. Какие компьютеры называются «машиной с произвольным доступом к памяти»?
13. Как размер входных данных алгоритма влияет на эффективность работы программы?
14. Например «Алгоритма линейного поиска элемента в массиве» рассчитайте количество операций алгоритма в наихудшем, наилучшем и среднем случае.
15. Как проводится анализ сложности алгоритмов?
16. Что такое «асимптотические обозначения» и как их применяют для анализа эффективности алгоритмов?
17. Чем отличаются асимптотические обозначения:  $O$  (о большое),  $\Omega$  (омега большое),  $\Theta$  (тета большое)?
18. Опишите расчет алгоритмической сложности в наихудшем, наилучшем и среднем случае для  $O$ -большого?
19. Опишите расчет алгоритмической сложности в наихудшем, наилучшем и среднем случае для омега большое?
20. Опишите расчет алгоритмической сложности в наихудшем, наилучшем и среднем случае для тета большое?
21. Каковы свойства асимптотических функций и как их используют для расчета алгоритмической сложности?
22. На примере объясните этапы асимптотического анализа?
23. Что такое временная сложность и пространственные сложности алгоритмов?
24. Какие функции называют рекурсивными?
25. На конкретном примере алгоритма расскажите какие типы рекурсии бывают?
26. Что такое системный стек, и какие ошибки могут возникнуть при использовании рекурсивных алгоритмов?
27. Объясните как провести анализ сложности рекурсивных алгоритмов?
28. В чем состоит метод математической индукции в рамках анализ сложности рекурсивных алгоритмов?
29. Рассчитайте временную сложность алгоритма вычисления факториала числа?
30. В чем состоит общий метод решения рекуррентных соотношений?
31. Рассчитайте временную сложность алгоритма бинарного поиска.
32. Рассчитайте временную сложность алгоритма сортировки слиянием.
33. В чем состоят основные задачи сортировки?
34. Что такое устойчивая и неустойчивая сортировки?
35. Чем отличаются алгоритмы сортировок? Какова их вычислительная сложность?
36. Объясните алгоритма пузырьковой сортировки и вычислите ее временную сложность.
37. Что такое «метод грубой силы»?
38. Какие методы уменьшения размера задачи вы знаете?
39. Объясните алгоритма сортировки вставкой и вычислите ее временную сложность.

40. В чем состоит метод «разделяй и властвуй»?
41. Объясните, как работает метод сортировки слиянием?
42. Объясните, как работает метод быстрой сортировки? В чем особенность выбора опорного элемента?
43. Рассчитайте временную сложность алгоритма быстрой сортировки.
44. Проанализируйте временную сложность алгоритма сортировки подсчетом.
45. Дайте понятие структуры данных. Опишите особенности работы с линейными структурами данных.
46. Опишите на примере основные методы работы с массивами.
47. Опишите на примере основные методы работы со строками.
48. Что такое коллекции? Особенности и методы работы со стеком, очередью, словарем. В чем особенность работы с этими структурами данных?
49. Объясните внутреннюю структуру хранения данных в списках. Какие типы списков вы знаете?
50. В чем отличие списков от одномерных массивов? В каких случаях можно использовать списки, одномерные массивы?
51. Опишите особенности структуры данных «очередь с приоритетами».
52. Что такое Хэш-набор? Какие методы работы с этой структурой данных?
53. Что такое обобщенные типы данных? Их реализация.
54. Что такое графы, какие типы графов бывают?
55. Типы представления графов.
56. Что такое матрица смежности и списки смежных вершин? В каких случаях используют эти структуры хранения данных?
57. Взвешенные графы, особенности структуры хранения данных?
58. Что такое пути и циклы? Связанность, ориентированность.
59. Объясните, какие структуры данных называю деревья, и каковы их свойства
60. Приведите псевдокод алгоритма: «Поиск в глубину» и объясните его работу.
61. Приведите псевдокод алгоритма: «Поиск в ширину» и объясните его работу.
62. Что такое «жадные алгоритмы»? Для чего используются жадные алгоритмы?
63. Для чего в программировании используются остовные (покрывающие) деревья?
64. Что такое минимальные остовные деревья?
65. В чем состоит алгоритм Крускала. Приведите его псевдокод.
66. Расскажите, какие алгоритмы поиска кратчайшего пути в графе вы знаете?
67. В чем состоит алгоритм Прима. Приведите его псевдокод.
68. Объясните работу алгоритма Дейкстры? Приведите его псевдокод.
69. В чем состоит особенность алгоритма Беллмана Форда?
70. Как работает стадия релаксации в алгоритме Беллмана Форда?
71. Как работает алгоритм кодировки Хаффмана? Объясните на конкретном примере.
72. Объясните логику жадных алгоритмов на примере алгоритма подсчет выдачи сдачи.

## **5.2. Перечень работ, выполняемых по дисциплине (модулю, практике, НИР) - эссе, рефераты, практические и расчетно-графические работы, курсовые работы, проекты и др.**

Примерные задания используемые для отработки навыков в этом курсе

1 семестр

1. Консольный ввод и вывод (2 часа)

Задачи:

1. Программа запрашивает два целых числа и выводит их сумму, разность, произведение и частное (с проверкой деления на ноль).
2. Программа запрашивает имя пользователя и возраст, выводит приветствие с учётом склонения слова «год».
3. Вычисление значения выражения с вещественными числами (например,  $(a + b) / (c - d)$ ) с контролем корректности ввода.

2. Работа с целыми, вещественными числами, символами и логическими переменными (2 часа)

Задачи:

1. Определение чётности/нечётности числа и вывод соответствующего сообщения.
2. Нахождение максимального из трёх введенных чисел с использованием условного оператора.
3. Проверка, является ли год високосным (делится на 4, но не на 100, или делится на 400).
4. Преобразование введенного символа: если это латинская буква, вывести её в верхнем регистре; иначе вывести код символа.

3. Работа со строками (4 часа)

Задачи:

1. Подсчёт количества слов в строке (слова разделены пробелами).
2. Проверка, является ли строка палиндромом (без учёта регистра и пробелов).
3. Замена всех пробелов в строке на символ подчёркивания.
4. Разбиение строки на слова и вывод их в обратном порядке.

4. Массивы (6 часов)

Задачи:

1. Одномерные массивы:

- o Заполнение массива случайными числами, вывод на экран.
- o Нахождение минимального элемента и его индекса.
- o Сортировка массива методом пузырька (или выбором).

2. Двумерные массивы (матрицы):

- o Создание матрицы заданного размера, заполнение случайными числами, вывод в виде таблицы.
- o Вычисление суммы элементов главной диагонали.

о Транспонирование матрицы.

3. Ступенчатые массивы (зубчатые):

о Создание массива массивов разной длины, заполнение, вывод суммы элементов каждой строки.

5. Функции (2 часа)

Задачи:

1. Написание функции для вычисления факториала числа (нерекурсивно).

2. Функция, принимающая массив целых чисел и возвращающая его сумму.

3. Перегруженные функции для вычисления площади круга (по радиусу) и прямоугольника (по сторонам).

4. Функция с параметрами по умолчанию и передачей параметров по ссылке (ref/out).

6. Рекурсия (2 часа)

Задачи:

1. Вычисление факториала числа с помощью рекурсии.

2. Вычисление n-го числа Фибоначчи рекурсивным способом.

3. Нахождение наибольшего общего делителя (НОД) двух чисел по алгоритму Евклида (рекурсивная версия).

2 семестр

7. Использование LINQ (9 часов)

Задачи:

1. Из массива целых чисел с помощью LINQ выбрать все чётные, отсортировать по убыванию и вывести.

2. Из списка строк отобрать те, которые начинаются с заданной буквы, и преобразовать в верхний регистр.

3. Вычисление среднего арифметического, минимального и максимального значения в коллекции с использованием агрегатных функций LINQ.

4. Работа с коллекцией объектов:

о Создать класс Student (поля: Name, Age, Grade).

о Сформировать список студентов.

о С помощью LINQ выполнить группировку по возрасту и подсчитать средний балл в каждой группе.

5. Применение LINQ к матрице: выбрать все положительные элементы и преобразовать в одномерный массив.

8. Реализация алгоритмов сортировки (9 часов)

Задачи:

1. Реализация «пузырьковой» сортировки (Bubble Sort).

2. Реализация сортировки вставками (Insertion Sort).

3. Реализация сортировки выбором (Selection Sort).

4. Реализация быстрой сортировки (Quick Sort) — рекурсивная версия.

5. Реализация сортировки слиянием (Merge Sort).

6. Сравнение времени работы реализованных алгоритмов на массивах разного размера (генерация случайных данных, замер времени).

Примеры домашних заданий

Домашняя работа №1. Введение в алгоритмизацию и линейные алгоритмы

Темы: консольный ввод/вывод, типы данных, условный оператор, простые вычисления.

Задачи:

1. Калькулятор времени.

Написать программу, которая запрашивает у пользователя количество секунд и выводит это же время в формате «часы:минуты:секунды».

Пример: ввод 3665 → вывод 1:1:5.

2. Корни квадратного уравнения.

Решить квадратное уравнение с коэффициентами, введёнными пользователем. Программа должна обрабатывать все случаи: два корня, один корень, отсутствие действительных корней. Результат выводить с точностью до 3 знаков.

3. Високосный год.

Написать программу, которая проверяет, является ли год високосным. Учитывать, что год високосный, если он делится на 4, но не делится на 100, за исключением годов, кратных 400. Вывести сообщение и количество дней в году (365 или 366).

4. Сравнение строк (без учёта регистра).

Программа запрашивает две строки и определяет, равны ли они без учёта регистра. Вывести результат сравнения (равны / не равны). Использовать стандартные методы строк.

Домашняя работа №2. Циклы и массивы

Темы: циклы (for, while, do-while), одномерные массивы, двумерные массивы.

Задачи:

1. Числа Фибоначчи.

Вывести первые N чисел Фибоначчи, где N задаёт пользователь. Программа должна использовать цикл (без рекурсии).

2. Статистика массива.

Заполнить одномерный массив случайными целыми числами от –50 до 50. Размер массива вводится пользователем.

Найти:

о сумму положительных элементов;

о количество отрицательных элементов;

о среднее арифметическое всех элементов;

о максимальный элемент и его индекс.

3. Магический квадрат (упрощённый вариант).

Проверить, является ли квадратная матрица (размер  $3 \times 3$ ) магическим квадратом (суммы по строкам, столбцам и двум диагоналям равны). Матрицу либо ввести вручную, либо сгенерировать случайно. Вывести результат и саму матрицу.

#### 4. Удаление дубликатов.

Дан одномерный массив целых чисел. Создать новый массив, содержащий только уникальные элементы исходного (порядок сохранить). Реализовать без использования HashSet.

#### Домашняя работа №3. Функции и рекурсия

Темы: функции, параметры, перегрузка, рекурсия.

Задачи:

##### 1. Простые числа.

Написать функцию IsPrime(int n), возвращающую true, если число простое, и false в противном случае. Использовать её для вывода всех простых чисел в заданном диапазоне [a, b].

##### 2. Функции для работы с массивами.

Реализовать набор функций:

- o FillRandom(int[] arr, int min, int max) – заполняет массив случайными числами;
- o PrintArray(int[] arr) – выводит массив в строку;
- o SortArray(int[] arr) – сортирует массив по возрастанию (любой метод);
- o ReverseArray(int[] arr) – переворачивает массив.

В Main() продемонстрировать их работу.

##### 3. Рекурсивные вычисления.

Написать рекурсивную функцию для вычисления:

- o числа Фибоначчи;
- o факториала;
- o суммы цифр числа.

Сравнить время выполнения с итеративными версиями для больших значений.

##### 4. Перегрузка функций.

Создать перегруженные функции для вычисления площади:

- o прямоугольника (две стороны);
- o круга (радиус);
- o треугольника (три стороны по формуле Герона).

В Main() запросить у пользователя фигуру и вызвать соответствующую функцию.

#### Домашняя работа №4. Основы объектно-ориентированного программирования

Темы: классы, объекты, свойства, методы, конструкторы, инкапсуляция.

Задачи:

##### 1. Класс Person.

Создать класс Person с полями:

- o string name;
- o int age;
- o string gender.

Добавить:

- o конструктор с параметрами;
- o свойства (проверка возраста на неотрицательность);
- o метод DisplayInfo() для вывода информации.

В Main() создать массив из 3–5 человек и вывести всех.

##### 2. Класс BankAccount.

Реализовать класс BankAccount с полями:

- o string accountNumber;
- o string owner;
- o double balance.

Методы:

- o Deposit(double amount) – пополнение;
- o Withdraw(double amount) – снятие с проверкой достаточности средств;
- o Transfer(BankAccount target, double amount) – перевод на другой счёт.

Создать два счёта, выполнить несколько операций, вывести остатки.

##### 3. Класс Fraction (дробь).

Создать класс Fraction, хранящий числитель и знаменатель. Реализовать:

- o конструктор, сокращающий дробь;
- o перегрузку операторов +, -, \*, /;
- o перегрузку ToString() для отображения в виде числитель/знаменатель.

В Main() продемонстрировать арифметические операции с дробями.

#### Семестр 2

#### Домашняя работа №5. Работа с коллекциями и LINQ

Темы: List<T>, Dictionary<TKey, TValue>, LINQ-запросы.

**Задачи:****1. Анализ текста.**

Загрузить текст из файла (например, «Война и мир» – небольшой отрывок). Используя LINQ, подсчитать:

- о количество слов в тексте;
- о частоту каждого слова (с игнорированием регистра и знаков препинания);
- о вывести топ-10 самых часто встречающихся слов.

**2. Студенты.**

Создать класс Student с полями: string Name, int Age, string Group, double AverageGrade. Сформировать список из 10–15 студентов (можно сгенерировать случайно). С помощью LINQ:

- о выбрать студентов, у которых средний балл  $\geq 4.5$ ;
- о сгруппировать по группам и вычислить средний балл по группе;
- о найти самого старшего студента;
- о вывести список студентов, отсортированный по убыванию среднего балла.

**3. Работа со словарём.**

Создать Dictionary<string, double>, где ключ – название товара, значение – цена. Написать программу, которая:

- о добавляет новые товары;
- о обновляет цены;
- о выводит все товары с ценой выше заданной.

Использовать LINQ для фильтрации.

Домашняя работа №6. Алгоритмы сортировки и рекурсивные алгоритмы

Темы: быстрая сортировка, сортировка слиянием, анализ сложности, рекурсия на графах.

**Задачи:****1. Реализация алгоритмов сортировки.**

Написать функции для сортировки:

- о быстрая сортировка (QuickSort);
- о сортировка слиянием (MergeSort).

Для каждого алгоритма:

- о реализовать рекурсивно;
- о замерить время выполнения для массивов размером 1000, 10000, 100000 (сгенерированных случайно);
- о сравнить с встроенной Array.Sort().

**2. Рекурсивный обход каталогов.**

Написать программу, которая рекурсивно обходит указанную папку и выводит структуру каталогов (дерево) с указанием размера каждого файла. Использовать классы DirectoryInfo, FileInfo. Результат сохранять в текстовый файл.

Поиск в глубину (DFS) на графе.

Реализовать класс Graph с представлением в виде списка смежности. Добавить методы:

- о добавление ребра;
- о рекурсивный DFS, выводящий все достижимые вершины из заданной;
- о поиск пути между двумя вершинами (с возвращением списка вершин).

Протестировать на небольшом графе (можно задать в коде).

### 5.3. Оценочные материалы, используемые для экзамена (описание билетов, тестов и т.п.)

По дисциплине промежуточная аттестация предусмотрена в форме экзамена и зачета.

Экзаменационные билеты состоит из двух теоретических вопросов и одного практического. Билеты хранятся на кафедре.

### 5.4. Методика оценки освоения дисциплины (модуля, практики. НИР)

**ШКАЛА ОЦЕНИВАНИЯ ДОМАШНИХ ЗАДАНИЙ:**

Оценка "зачтено" - задания выполнены полностью.

Оценка "не зачтено" - задания выполнены не в полном объеме, допущены ошибки.

**ШКАЛА ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ:**

«отлично» – студент показывает глубокие, исчерпывающие знания в объеме пройденной программы, уверенно действует по применению полученных знаний на практике, грамотно и логически стройно излагает материал при ответе, умеет формулировать выводы из изложенного теоретического материала, знает дополнительно рекомендованную литературу;

«хорошо» – студент показывает твердые и достаточно полные знания в объеме пройденной программы, допускает незначительные ошибки при освещении заданных вопросов, правильно действует по применению знаний на практике, четко излагает материал;

«удовлетворительно» – студент показывает знания в объеме пройденной программы, ответы излагает хотя и с ошибками, но уверенно исправляемыми после дополнительных и наводящих вопросов, правильно действует по применению знаний на практике;

«неудовлетворительно» – студент допускает грубые ошибки в ответе, не понимает сущности излагаемого вопроса, не умеет применять знания на практике, дает неполные ответы на дополнительные и наводящие вопросы.

**Шкала оценивания рефератов:**

«Зачтено» - Студент владеет теоретическим материалом (возможно на минимально допустимом уровне), отсутствуют ошибки при описании теории, но возможно испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.

«Не зачтено» - Студент не владеет теоретическим материалом, допуская ошибки по сущности рассматриваемых

(обсуждаемых) вопросов, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допускает ошибки при ответе на дополнительные вопросы.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

### 6.1. Рекомендуемая литература

#### 6.1.1. Основная литература

|      | Авторы, составители                 | Заглавие                                                               | Библиотека           | Издательство, год                     |
|------|-------------------------------------|------------------------------------------------------------------------|----------------------|---------------------------------------|
| Л1.1 | Крапухина Н.В.,<br>Светозарова Г.И. | Информатика. Основы алгоритмизации и программирования: Учебное пособие | Методические пособия | Москва, 2005                          |
| Л1.2 | Павловская Т.А.<br>Павловская Т.А.  | С# Программирование на языке высокого уровня:: учебник                 | Электронный каталог  | Питер СПб:Питер, 2013                 |
| Л1.3 | Робертсон Л.Н. Л.А.<br>Робертсон    | Программирование - это просто. Пошаговый подход: учебное пособие       | Электронный каталог  | Москва БИНОМ. Лаборатрия знаний, 2012 |
| Л1.4 | Незнанов А.А.                       | Программирование и алгоритмизация: учебник                             | Электронный каталог  | Москва Изд.-й центр "Академия", 2010  |

### 6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

|    |                                                                                               |                                                         |
|----|-----------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Э1 | Научная электронная библиотека <a href="https://elibrary.ru">https://elibrary.ru</a>          | <a href="https://elibrary.ru/">https://elibrary.ru/</a> |
| Э2 | Электронная библиотека МИСиС <a href="http://lib.misis.ru">http://lib.misis.ru</a>            | <a href="http://lib.misis.ru">http://lib.misis.ru</a>   |
| Э3 | ЭБС Университетская библиотека онлайн <a href="http://biblioclub.ru">http://biblioclub.ru</a> | <a href="http://biblioclub.ru">http://biblioclub.ru</a> |

### 6.3 Перечень программного обеспечения

|     |                    |
|-----|--------------------|
| П.1 | Microsoft Office   |
| П.2 | Visual Studio 2019 |

### 6.4. Перечень информационных справочных систем и профессиональных баз данных

## 7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

| Ауд. | Назначение                               | Оснащение                                                                                   |
|------|------------------------------------------|---------------------------------------------------------------------------------------------|
| 11   | Основы алгоритмизации и программирования | 30 посадочных мест, лингафонное оборудование, 15 компьютеров для студентов. 1 компьютер для |

## 8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

1. Посещать все виды занятий.
  2. Своевременно зарегистрироваться на рекомендованные электронные ресурсы - LMS Canvas и MS Teams.
  3. При возникновении любых вопросов по содержанию курса и организации работы своевременно обращаться к преподавателю.
  4. Своевременно выполнять домашние задания.
- Качественное освоение дисциплины возможно только при систематической самостоятельной работе, что поддерживается системой текущей аттестации на LMS Canvas.