

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и компьютерных наук

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор



* Т.А. Хагуров

подпись

«30» мая 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.14 Технология программирования и работы на ЭВМ

Направление подготовки: 01.05.01 Фундаментальная математика и механика

Направленность (профиль): Фундаментальная математика и её приложения;
Фундаментальная математика и её приложения

Форма обучения: очная

Квалификация: Математик. Механик. Преподаватель

Краснодар 2022

Рабочая программа дисциплины Б1.О.14 ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ И РАБОТА НА ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНОЙ МАШИНЕ (ЭВМ) составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.05.01 Фундаментальные математика и механика

Программу составил(и):

Голуб М. В., заведующий кафедрой, д. ф.-м. н., доцент



Дорошенко О. В., доцент, к. ф.-м. н.



Рабочая программа дисциплины Б1.О.14 ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ И РАБОТА НА ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНОЙ МАШИНЕ (ЭВМ) утверждена на заседании кафедры ТЕОРИЯ ФУНКЦИИ протокол № 9 «12» апреля 2022 г. Заведующий кафедрой Голуб М. В.



Утверждена на заседании учебно-методической комиссии факультета математики и компьютерных наук протокол № 5 «5» мая 2022 г. Председатель УМК факультета/института Шмалько С. П.



Рецензенты:

Фоменко Сергей Иванович, канд. физ. - мат. наук,
старший научный сотрудник лаборатории волновых процессов

Лепетухин Михаил Викторович,
председатель правления КПК «Кубанский капитал»

1 Цели и задачи изучения дисциплины

1.1 Цель освоения дисциплины

Целью изучения дисциплины является подготовка в области применения современной вычислительной техники для практического решения прикладных и теоретических математических задач.

1.2 Задачи дисциплины

- ознакомить студентов с основными структурами данных и методами структурного программирования, современными информационными технологиями и системами программирования;
- научить применять современные информационные технологии для решения различных задач;
- обучить основам программирования на алгоритмических языках высокого уровня;
- научить использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных математических задач.

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Технология программирования и работы на ЭВМ» относится к базовой части Блока 1 "Дисциплины (модули)" учебного плана.

Для её успешного изучения необходимы знания и умения, приобретенные в результате освоения школьного курса математики и информатики, а также некоторых разделов математического анализа и алгебры.

Изучение дисциплины «Технологии программирования и работы на ЭВМ» позволит студентам проводить научные исследования в различных областях математики, а также качественно оформлять свои учебные и научные работы. Полученные знания необходимы для изучения последующих дисциплин, таких как Б1.О.13 «Численные методы», Б1.О.27 «Алгоритмы статистического анализа данных», Б1.В.02 «Теоретико-числовые методы криптографии», Б1.В.06 «Практикум на ЭВМ» и практически всех дисциплин по выбору.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

| Код и наименование индикатора достижения компетенции | Результаты обучения по дисциплине |
|--|---|
| УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий | |
| ИУК-1.2. Осуществляет поиск, отбор и систематизацию информации для определения альтернативных вариантов стратегических решений в проблемной ситуации и обоснования выбора оптимальной стратегии с учетом поставленной цели, рисков и возможных последствий | Знает основные подходы к алгоритмизации решения математических и практических задач |
| | Умеет разрабатывать, анализировать, адаптировать и совершенствовать алгоритмы |
| | Владеет навыками практического использования математических алгоритмов |
| ПК-4 способен разрабатывать программное обеспечение для решения прикладных задач в сфере профессиональной деятельности | |
| ИПК-1.2. Имеет навыки использования современных языков программирования для разработки программного обеспечения | Знает основные приемы используемые при разработке программного обеспечения на современных языках программирования |
| | Умеет реализовывать алгоритмы с использованием современных языков программирования |
| | Владеет навыками практического программирования |
| ОПК-3 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности | |

| Код и наименование индикатора достижения компетенции | Результаты обучения по дисциплине |
|--|---|
| ИОПК-3.1. Имеет представление о принципах работы современных информационных технологий | Знает базовые принципы работы современных информационных технологий, основные методы математического и алгоритмического моделирования |
| | Умеет выбирать подходящие программные и технические средства для решения конкретных задач |
| | Владеет базовыми инструментами, используемыми в современных информационных технологиях |
| ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | |
| ИОПК-5.2. Реализует алгоритмы с использованием современных средств разработки прикладного программного обеспечения | Знает основные концепции, используемые при программной реализации алгоритмов |
| | Умеет выбирать корректные методы при программной реализации решения математических задач |
| | Владеет навыками применения современных средств разработки прикладного программного обеспечения |

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 16 зачетных единиц (576 часов), их распределение по видам работ представлено в таблице

| Виды работ | Всего часов | Форма обучения | | | | | |
|---|--------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | | очная | | | | | |
| | | 1 семестр (часы) | 2 семестр (часы) | 3 семестр (часы) | 4 семестр (часы) | 5 семестр (часы) | 6 семестр (часы) |
| Контактная работа, в том числе: | 375,2 | 72,2 | 54,2 | 54,2 | 66,2 | 72,2 | 56,2 |
| Аудиторные занятия (всего): | | 68 | 48 | 52 | 64 | 68 | 52 |
| занятия лекционного типа | | 34 | 16 | 18 | 16 | 34 | 18 |
| лабораторные занятия | | 34 | 32 | 34 | 48 | 34 | 34 |
| Иная контактная работа: | | 4,2 | 6,2 | 2,2 | 2,2 | 4,2 | 4,2 |
| Контроль самостоятельной работы (КСР) | | 4 | 6 | 2 | 2 | 4 | 4 |
| Промежуточная аттестация (ИКР) | | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 |
| Самостоятельная работа, в том числе: | 200,8 | 35,8 | 53,8 | 17,8 | 5,8 | 35,8 | 51,8 |
| <i>Контрольная работа</i> | | | | | | | |
| <i>Самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т. д.)</i> | | | | | | | |
| Подготовка к текущему контролю | – | – | – | – | – | – | – |
| Контроль: | – | – | – | – | – | – | – |
| Подготовка к экзамену | – | – | – | – | – | – | – |
| час. | 576 | 108 | 108 | 72 | 72 | 108 | 108 |

| | | | | | | | | |
|-------------------------------|--|--------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Общая трудоемкость | в том числе контактная работа | 375,2 | 72,2 | 54,2 | 54,2 | 66,2 | 72,2 | 56,2 |
| | зач. ед | 16 | 3 | 3 | 2 | 2 | 3 | 3 |

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 1 семестре

| № | Наименование разделов (тем) | Количество часов | | | | |
|----|---------------------------------------|------------------|-------------------|----|----|----------------------|
| | | Всего | Аудиторная работа | | | Внеаудиторная работа |
| | | | Л | ПЗ | ЛР | |
| 1. | Алгоритмы и структуры данных | 11 | 6 | | 2 | 3 |
| 2. | Функции алгебры логики | 23 | 8 | | 6 | 9 |
| 3. | Массивы, циклы и сортировки | 30 | 8 | | 12 | 10 |
| 4. | Функции | 22 | 4 | | 10 | 8 |
| 5. | Конечные автоматы | 17,8 | 8 | | 4 | 5,8 |
| | <i>ИТОГО по разделам дисциплины</i> | 103,8 | 34 | | 34 | 35,8 |
| | Контроль самостоятельной работы (КСР) | 4 | | | | |
| | Промежуточная аттестация (ИКР) | 0,2 | | | | |
| | Подготовка к текущему контролю | – | | | | |
| | Общая трудоемкость по дисциплине | 108 | | | | |

Разделы (темы) дисциплины, изучаемые во 2 семестре

| № | Наименование разделов (тем) | Количество часов | | | | |
|----|---------------------------------------|------------------|-------------------|----|----|----------------------|
| | | Всего | Аудиторная работа | | | Внеаудиторная работа |
| | | | Л | ПЗ | ЛР | |
| 6. | Строки | | 4 | | 10 | 23,8 |
| 7. | Модули и пакеты в Python | | 4 | | 6 | 8 |
| 8. | Файлы | | 4 | | 8 | 8 |
| 9. | Работа с графикой | | 4 | | 8 | 14 |
| | <i>ИТОГО по разделам дисциплины</i> | 101,8 | 16 | | 32 | 53,8 |
| | Контроль самостоятельной работы (КСР) | 6 | | | | |
| | Промежуточная аттестация (ИКР) | 0,2 | | | | |
| | Подготовка к текущему контролю | – | | | | |
| | Общая трудоемкость по дисциплине | 108 | | | | |

Разделы (темы) дисциплины, изучаемые в 3 семестре

| № | Наименование разделов (тем) | Количество часов | | | | |
|-----|---------------------------------------|------------------|-------------------|----|----|----------------------|
| | | Всего | Аудиторная работа | | | Внеаудиторная работа |
| | | | Л | ПЗ | ЛР | |
| 10. | Библиотека для анализа данных NumPy | | 10 | | 20 | 10 |
| 11. | Библиотека для анализа данных pandas. | | 8 | | 14 | 7,8 |
| | <i>ИТОГО по разделам дисциплины</i> | 69,8 | 18 | | 34 | 17,8 |
| | Контроль самостоятельной работы (КСР) | 2 | | | | |
| | Промежуточная аттестация (ИКР) | 0,2 | | | | |
| | Подготовка к текущему контролю | – | | | | |
| | Общая трудоемкость по дисциплине | 72 | | | | |

Разделы (темы) дисциплины, изучаемые в 4 семестре

| № | Наименование разделов (тем) | Количество часов | | | | |
|-----|--|------------------|-------------------|----|----|----------------------|
| | | Всего | Аудиторная работа | | | Внеаудиторная работа |
| | | | Л | ПЗ | ЛР | |
| 12. | Объектно-ориентированное программирование. Классы и методы | | 8 | | 28 | 2,8 |
| 13. | Наследование, полиморфизм и инкапсуляция | | 8 | | 20 | 3 |

| | | | | | | |
|--|---------------------------------------|------|----|--|----|-----|
| | <i>ИТОГО по разделам дисциплины</i> | 69,8 | 16 | | 48 | 5,8 |
| | Контроль самостоятельной работы (КСР) | 2 | | | | |
| | Промежуточная аттестация (ИКР) | 0,2 | | | | |
| | Подготовка к текущему контролю | – | | | | |
| | Общая трудоемкость по дисциплине | 72 | | | | |

Разделы (темы) дисциплины, изучаемые в 5 семестре

| № | Наименование разделов (тем) | Количество часов | | | | |
|-----|--|------------------|-------------------|----|-----------------------------|------|
| | | Всего | Аудиторная работа | | Внеаудиторная работа СРС | |
| | | | Л | ПЗ | | ЛР |
| 14. | Основы построения параллельных вычислительных систем | | 8 | | 4 | 5,8 |
| 15. | Параллельные вычисления | | 14 | | 16 | 16 |
| 16. | Параллельная обработка больших массивов данных | | 12 | | 14 | 14 |
| | <i>ИТОГО по разделам дисциплины</i> | 103,8 | 34 | | 34 | 35,8 |
| | Контроль самостоятельной работы (КСР) | 4 | | | | |
| | Промежуточная аттестация (ИКР) | 0,2 | | | | |
| | Подготовка к текущему контролю | – | | | | |
| | Общая трудоемкость по дисциплине | 108 | | | | |

Разделы (темы) дисциплины, изучаемые в 6 семестре

| № | Наименование разделов (тем) | Количество часов | | | | |
|-----|---|------------------|-------------------|----|-----------------------------|------|
| | | Всего | Аудиторная работа | | Внеаудиторная работа СРС | |
| | | | Л | ПЗ | | ЛР |
| 17. | Распределенные системы и вычисления | | 8 | | 14 | 23,8 |
| 18. | Технологии распределенного программирования | | 10 | | 20 | 30 |
| | <i>ИТОГО по разделам дисциплины</i> | 103,8 | 18 | | 34 | 53,8 |
| | Контроль самостоятельной работы (КСР) | 4 | | | | |
| | Промежуточная аттестация (ИКР) | 0,2 | | | | |
| | Подготовка к текущему контролю | – | | | | |
| | Общая трудоемкость по дисциплине | 108 | | | | |

Разделы (темы) дисциплины, изучаемые в 3 семестре

Примечание: Л – лекции, ПЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

| № | Наименование раздела (темы) | Тематика занятий/разбор | Форма текущего контроля |
|----|------------------------------|--|-------------------------|
| 1. | Алгоритмы и структуры данных | Классификация ЯП. Знакомство с Python. Типы данных и переменные. Базовые операции. Интерпретация ошибок. РЕР8. Алгоритмизация. Языки программирования и алгоритмы. Математические множества и типы данных. Структуры данных: множества, связные списки, стеки, дек, очереди, словари, хэш-таблицы, двоичные и префиксные деревья, графы. | К |
| 2. | Функции алгебры логики | Функции алгебры логики. Логический тип. Операции отношения. Логические операции. Условные операторы. Составной оператор. Метки. Отладка программ в Python. | К |
| 3. | Функции | Функции. Исключения и их обработка. Именные функции, инструкция def. Аргументы функции. Анонимные функции, инструкция lambda. Рекурсия. Прямой и обратный ход рекурсии. | Т |
| 4. | Массивы, циклы и сортировки | Массивы. Операторы цикла. Списки, кортежи и словари. Операторы общие для всех типов последовательностей. | Т |

| | | | |
|-----|--|--|----------|
| | | Строки. Итераторы, генераторы, генераторы списков. Индексы и срезы. Методы сортировки. Пузырьковая сортировка и её улучшения. Сортировка перемешиванием (шейкерная сортировка). Сортировка расчёской. Сортировка вставками. Сортировка выбором. | |
| 5. | Конечные автоматы | Детерминированные и недетерминированные конечные автоматы. Таблица переходов. Конечные автоматы с магазинной памятью. | <i>T</i> |
| 6. | Строки | Срезы строк и списков. Присваивание в срез списка. Стандартные функции len, max, min, sum. Методы split и join. Структура данных Куча/Пирамида (Heap) Расстояние Левенштейна. Проверка равенства строк. Наивный поиск подстроки в строке. Алгоритм Кнута-Морриса-Пратта. | <i>K</i> |
| 7. | Модули и пакеты в Python | Основные стандартные модули и пакеты в Python. Импортное модулей. Создание собственных модулей и их импортное. Специализированные модули и приложения. | <i>T</i> |
| 8. | Файлы | Ввод-вывод информации. Система управления вводом-выводом. Методы доступа. Метки файлов. Открытие и закрытие файлов. Логические и физические файлы. Общие операции над файлами. Типизированные файлы. Бестиповые файлы. Файлы прямого доступа. Текстовые файлы. | <i>K</i> |
| 9. | Работа с графикой | Библиотека Matplotlib. Контейнер Figure. Область рисования. Мультиоконные рисунки. Координатные оси. Деления координатных осей. Специальные элементы рисунка в matplotlib. Особенности координатных осей. Графики в полярной системе координат. Реализация собственных функций. | <i>T</i> |
| 10. | Библиотека для анализа данных NumPy | Создание массивов NumPy. Арифметические операции над массивами NumPy. Индексация массива NumPy. Агрегирование в NumPy. Операции над матрицами и векторами в NumPy. Индексация матрицы NumPy. Агрегирование матриц NumPy. Транспонирование и изменение формы матриц в NumPy. Таблицы NumPy. Аудио и временные ряды в NumPy Обработка изображений и текста в NumPy. | <i>T</i> |
| 11. | Библиотека для анализа данных pandas. | Библиотека pandas. Работа с Датафреймами (таблицами) в pandas. Фильтрация наблюдений. Группировка и агрегирование. Графики и pandas. Визуализация. Регулярные выражения. | <i>T</i> |
| 12. | Объектно-ориентированное программирование. Классы и методы | Преимущества и недостатки ООП. Классы. Объекты. Атрибуты класса. Атрибуты класса против атрибутов экземпляров. Методы. Статические методы. Возврат множественных значений из метода. Метод str. Конструкторы. Локальные и глобальные переменные. Модификаторы доступа. | <i>K</i> |
| 13. | Наследование, полиморфизм и инкапсуляция | Наследование. Абстрактные классы. Уровень абстракции. Множественное наследование. Полиморфизм. Перегрузка метода. Переопределение метода. Инкапсуляция. Аксессуары (геттеры и сеттеры). Интерфейсы. | <i>K</i> |
| 14. | Основы построения параллельных вычислительных систем | Архитектура параллельных вычислительных систем. Статическая и динамическая конвейеризация. Векторные вычисления. Систематика Флинна. Примеры параллельных вычислений. Параллелизм. Ускорение. Эффективность. Масштабируемость. Оценка | <i>K</i> |

| | | | |
|-----|--|--|----------|
| | | <p>эффективности разработки. Ограничения сети. Методы параллелизма. Процессы и потоки.</p> <p>Типичные ошибки многопоточного программирования. Взаимное исключение и условная синхронизация. Модель памяти и низкоуровневые примитивы синхронизации. Альтернативные подходы к реализации одновременных программ.</p> | |
| 15. | Параллельные вычисления | <p>Области применения и проблематика. Современные параллельные вычислительные системы. Теоретические основы параллельных вычислений. Показатели качества параллельного алгоритма. Принципы разработки и типовые структуры параллельных алгоритмов. Методология РСАМ. Системы параллельного программирования. Типовые модели программирования и шаблоны. Основы параллельного программирования на системах с общей памятью на примере технологии OpenMP. Основы параллельного программирования на системах с распределенной памятью на примере технологии MPI.</p> | <i>T</i> |
| 16. | Параллельная обработка больших массивов данных | <p>Феномен Big Data. Модель программирования MapReduce. Принципы параллельной реализации вычислений. Область применения и примеры задач. Принципы распределенной реализации MapReduce на кластерных системах. Платформа Apache Hadoop. Интерфейсы прикладного программирования и реализация программ для Hadoop. Локальная отладка и запуск программ на кластере. Приемы и стратегии реализации MapReduce-программ. Высокоуровневые языки и инструментарии для работы с Hadoop. Практические примеры использования MapReduce. Ограничения модели MapReduce, расширения и альтернативные подходы.</p> | <i>K</i> |
| 17. | Распределенные системы и вычисления | <p>Области применения, характерные особенности и виды распределенных систем. Проблемы построения распределенных систем. Теоретические основы распределенных вычислений, примеры распределенных алгоритмов. Способы взаимодействия распределенных процессов, сетевые протоколы.</p> | <i>T</i> |
| 18. | Технологии распределенного программирования | <p>Технологии распределенного программирования. Распределенные системы хранения данных, репликация данных, NoSQL-системы. Технологии распределенных вычислений, гриды, добровольные вычисления. Облачные вычислительные системы.</p> | <i>T</i> |

2.3.2 Занятия семинарского типа (практические / семинарские занятия/ лабораторные работы)

| № | Наименование раздела (темы) | Тематика занятий/работ | Форма текущего контроля |
|----|------------------------------|---|-------------------------|
| 1. | Алгоритмы и структуры данных | <p>Классификация ЯП. Знакомство с Python. Типы данных и переменные. Базовые операции. Интерпретация ошибок. PEP8. Алгоритмизация.</p> <p>Языки программирования и алгоритмы. Математические множества и типы данных.</p> <p>Структуры данных: множества, связные списки, стеки, дек, очереди, словари, хэш-таблицы, двоичные и префиксные деревья, графы.</p> | <i>KP</i> <i>РГЗ</i> |
| 2. | Функции алгебры логики | <p>Функции алгебры логики. Логический тип. Операции отношения. Логические операции. Условные операторы. Составной оператор. Метки.</p> <p>Отладка программ в Python.</p> | <i>РГЗ</i> |
| 3. | Функции | <p>Функции. Исключения и их обработка.</p> <p>Именные функции, инструкция def. Аргументы функции.</p> <p>Анонимные функции, инструкция lambda.</p> <p>Рекурсия. Прямой и обратный ход рекурсии.</p> | <i>РГЗ</i> |

| | | | |
|-----|--|--|-------------------|
| 4. | Массивы, циклы и сортировки | Массивы. Операторы цикла. Списки, кортежи и словари. Операторы общие для всех типов последовательностей. Строки. Итераторы, генераторы, генераторы списков. Индексы и срезы. Методы сортировки. Пузырьковая сортировка и её улучшения. Сортировка перемешиванием (шейкерная сортировка). Сортировка расчёской. Сортировка вставками. Сортировка выбором. | <i>КР РГЗ</i> |
| 5. | Конечные автоматы | Детерминированные и недетерминированные конечные автоматы. Таблица переходов. Конечные автоматы с магазинной памятью. | <i>РГЗ</i> |
| 6. | Строки | Срезы строк и списков. Присваивание в срез списка. Стандартные функции len, max, min, sum. Методы split и join. Структура данных Куча/Пирамида (Heap) Расстояние Левенштейна. Проверка равенства строк. Наивный поиск подстроки в строке. Алгоритм Кнута-Морриса-Практа. | <i>КР</i> |
| 7. | Модули и пакеты в Python | Основные стандартные модули и пакеты в Python. Импортирование модулей. Создание собственных модулей и их импортирование. Специализированные модули и приложения. | <i>РГЗ</i> |
| 8. | Файлы | Ввод-вывод информации. Система управления вводом-выводом. Методы доступа. Метки файлов. Открытие и закрытие файлов. Логические и физические файлы. Общие операции над файлами. Типизированные файлы. Бестиповые файлы. Файлы прямого доступа. Текстовые файлы. | <i>КР</i> |
| 9. | Работа с графикой | Библиотека Matplotlib. Контейнер Figure. Область рисования. Мультиоконные рисунки. Координатные оси. Деления координатных осей. Специальные элементы рисунка в matplotlib. Особенности координатных осей. Графики в полярной системе координат. Реализация собственных функций. | <i>РГЗ</i> |
| 10. | Библиотека для анализа данных NumPy | Создание массивов NumPy. Арифметические операции над массивами NumPy. Индексация массива NumPy. Агрегирование в NumPy. Операции над матрицами и векторами в NumPy. Индексация матрицы NumPy. Агрегирование матриц NumPy. Транспонирование и изменение формы матриц в NumPy. Таблицы NumPy. Аудио и временные ряды в NumPy. Обработка изображений и текста в NumPy. | <i>РГЗ</i> |
| 11. | Библиотека для анализа данных pandas. | Библиотека pandas. Работа с Датафреймами (таблицами) в pandas. Фильтрация наблюдений. Группировка и агрегирование. Графики и pandas. Визуализация. Регулярные выражения. | <i>РГЗ</i> |
| 12. | Объектно-ориентированное программирование. Классы и методы | Преимущества и недостатки ООП. Классы. Объекты. Атрибуты класса. Атрибуты класса против атрибутов экземпляров. Методы. Статические методы. Возврат множественных значений из метода. Метод str. Конструкторы. Локальные и глобальные переменные. Модификаторы доступа. | <i>КР</i> |
| 13. | Наследование, полиморфизм и инкапсуляция | Наследование. Абстрактные классы. Уровень абстракции. Множественное наследование. Полиморфизм. Перегрузка метода. Переопределение метода. Инкапсуляция. Аксессуары (геттеры и сеттеры). Интерфейсы. | <i>РГЗ</i> |
| 14. | Основы построения параллельных | Архитектура параллельных вычислительных систем. Статическая и динамическая конвейеризация. Векторные вычисления. Систематика Флинна. | <i>РГЗ</i> |

| | | | |
|-----|--|---|-----|
| | вычислительных систем | Примеры параллельных вычислений. Параллелизм. Ускорение. Эффективность. Масштабируемость. Оценка эффективности разработки. Ограничения сети. Методы параллелизма. Процессы и потоки. Типичные ошибки многопоточного программирования. Взаимное исключение и условная синхронизация. Модель памяти и низкоуровневые примитивы синхронизации. Альтернативные подходы к реализации одновременных программ. | |
| 15. | Параллельные вычисления | Области применения и проблематика. Современные параллельные вычислительные системы. Теоретические основы параллельных вычислений. Показатели качества параллельного алгоритма. Принципы разработки и типовые структуры параллельных алгоритмов. Методология РСАМ. Системы параллельного программирования. Типовые модели программирования и шаблоны. Основы параллельного программирования на системах с общей памятью на примере технологии OpenMP. Основы параллельного программирования на системах с распределенной памятью на примере технологии MPI. | КР |
| 16. | Параллельная обработка больших массивов данных | Феномен Big Data. Модель программирования MapReduce. Принципы параллельной реализации вычислений. Область применения и примеры задач. Принципы распределенной реализации MapReduce на кластерных системах. Платформа Apache Hadoop. Интерфейсы прикладного программирования и реализация программ для Hadoop. Локальная отладка и запуск программ на кластере. Приемы и стратегии реализации MapReduce-программ. Высокоуровневые языки и инструментарии для работы с Hadoop. Практические примеры использования MapReduce. Ограничения модели MapReduce, расширения и альтернативные подходы. | РГЗ |
| 17. | Распределенные системы и вычисления | Области применения, характерные особенности и виды распределенных систем. Проблемы построения распределенных систем. Теоретические основы распределенных вычислений, примеры распределенных алгоритмов. Способы взаимодействия распределенных процессов, сетевые протоколы. | РГЗ |
| 18. | Технологии распределенного программирования | Технологии распределенного программирования. Распределенные системы хранения данных, репликация данных, NoSQL-системы. Технологии распределенных вычислений, гриды, добровольные вычисления. Облачные вычислительные системы. | РГЗ |

Защита лабораторной работы (ЛР), выполнение курсового проекта (КП), курсовой работы (КР), расчетно-графического задания (РГЗ), написание реферата (Р), эссе (Э), коллоквиум (К), тестирование (Т) и выполнение контрольной работы (КР).

При изучении дисциплины применяется электронное обучение (проектор и ЭВМ), дистанционные образовательные технологии в соответствии с ФГОС ВО.

2.3.3 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены учебным планом.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

| № | Вид СРС | Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы |
|---|---------|---|
|---|---------|---|

| | | |
|---|---|---|
| 1 | Проработка и повторение лекционного материала и материала учебников и учебных пособий | <i>Онлайн-курс «Алгоритмы и структуры данных на Python». – МФТИ. – URL: https://mipt.lectoriy.ru/course/PythonAlgorithms/lectures</i> <i>Онлайн-курс лекций «Алгоритмы и структуры данных». – МФТИ. – URL: https://mipt.ru/online/algorithmov-i-tekhnologiy/algorithmy-struktury.php</i> |
| 2 | Подготовка к практическим занятиям | Онлайн-курс «Основы программирования на Python». – НИУ ВШЭ. – URL: https://ru.coursera.org/learn/python-osnovy-programmirovaniya |
| 3 | Подготовка к коллоквиуму | <i>Спиричева Н.Р. Структура данных и основные алгоритмы: Учебное пособие. - Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2004. - 92 с. — Текст: электронный// Информационная система "Единое окно доступа к образовательным ресурсам" — URL: http://window.edu.ru/resource/573/28573</i> |
| 4 | Выполнение контрольных и расчетно-графических заданий и контрольных работ | <i>Методические указания по выполнению расчетно-графических заданий, утвержденные на заседании кафедры теории функций факультета математики и компьютерных наук ФГБОУ ВО «КубГУ», протокол № 7 от 08.04.2021 г.</i> |

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В ходе изучения дисциплины предусмотрено использование следующих образовательных технологий: лекции, практические занятия, подготовка письменных аналитических работ, самостоятельная работа студентов.

Компетентностный подход в рамках преподавания дисциплины реализуется в использовании интерактивных технологий и активных методов (проектных методик, разбора конкретных ситуаций) в сочетании с внеаудиторной работой.

Информационные технологии, применяемые при изучении дисциплины: использование информационных ресурсов, доступных в информационно-телекоммуникационной сети Интернет.

Адаптивные образовательные технологии, применяемые при изучении дисциплины – для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Технология программирования и работы на ЭВМ».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме *тестовых заданий, разноуровневых заданий, отчетов по индивидуальным расчетно-графическим и проектно-групповым заданиям* и **промежуточной аттестации** в форме вопросов и заданий к зачету.

Структура оценочных средств для текущей и промежуточной аттестации

| № п/п | Код и наименование индикатора (в соответствии с п. 1.4) | Результаты обучения (в соответствии с п. 1.4) | Наименование оценочного средства | |
|-------|---|---|---|-------------------------------|
| | | | Текущий контроль | Промежуточная аттестация |
| 1 | <i>ИУК-1.2. Осуществляет поиск, отбор и систематизацию информации для определения альтернативных вариантов стратегических решений в проблемной ситуации и обоснования выбора оптимальной стратегии с учетом поставленной цели, рисков и возможных последствий</i> | Знает основные подходы к алгоритмизации решения математических и практических задач | <i>Вопросы на коллоквиуме 1-10, КР,</i> | <i>Вопросы на зачете 1-10</i> |
| 2 | <i>ИУК-1.2. Осуществляет поиск, отбор и систематизацию информации для определения альтернативных вариантов стратегических решений в проблемной ситуации и обоснования выбора оптимальной стратегии с учетом поставленной цели, рисков и возможных последствий</i> | Умеет разрабатывать, анализировать, адаптировать и совершенствовать алгоритмы | <i>РГЗ</i> | <i>Задание на зачете</i> |
| 3 | <i>ИУК-1.2. Осуществляет поиск, отбор и систематизацию информации для определения альтернативных вариантов стратегических решений в проблемной ситуации и обоснования выбора оптимальной стратегии с учетом поставленной цели, рисков и возможных последствий</i> | Владеет навыками практического использования математических алгоритмов | <i>РГЗ</i> | <i>Задание на зачете</i> |
| 4 | <i>ИОПК-3.1. Имеет представление о принципах работы современных информационных технологий</i> | Знает базовые принципы работы современных информационных технологий, основные методы математического и алгоритмического моделирования | <i>Вопросы на коллоквиуме КР,</i> | <i>Вопросы на зачете</i> |
| 5 | <i>ИОПК-3.1. Имеет представление о принципах работы современных информационных технологий</i> | Умеет выбирать подходящие программные и технические средства для решения конкретных задач | <i>РГЗ</i> | <i>Задание на зачете</i> |
| 6 | <i>ИОПК-3.1. Имеет представление о принципах работы современных информационных технологий</i> | Владеет базовыми инструментами, используемыми в современных информационных технологиях | <i>Вопросы на коллоквиуме КР,</i> | <i>Вопросы на зачете</i> |

| | | | | |
|----|---|--|-----------------------------------|--------------------------|
| 7 | <i>ИОПК-5.2. Реализует алгоритмы с использованием современных средств разработки прикладного программного обеспечения</i> | Знает основные концепции, используемые при программной реализации алгоритмов | <i>Вопросы на коллоквиуме РГЗ</i> | <i>Вопросы на зачете</i> |
| 8 | <i>ИОПК-5.2. Реализует алгоритмы с использованием современных средств разработки прикладного программного обеспечения</i> | Умеет выбирать корректные методы для программной реализации решения математических задач | <i>РГЗ, КР</i> | <i>Задание на зачете</i> |
| 9 | <i>ИОПК-5.2. Реализует алгоритмы с использованием современных средств разработки прикладного программного обеспечения</i> | Владет навыками применения современных средств разработки прикладного программного обеспечения | <i>РГЗ</i> | <i>Задание на зачете</i> |
| 10 | <i>ИПК-1.2. Имеет навыки использования современных языков программирования для разработки программного обеспечения</i> | Знает основные приемы, используемые при разработке программного обеспечения на современных языках программирования | <i>Вопросы на коллоквиуме КР,</i> | <i>Вопросы на зачете</i> |
| 11 | <i>ИПК-1.2. Имеет навыки использования современных языков программирования для разработки программного обеспечения</i> | Умеет реализовывать алгоритмы с использованием современных языков программирования | <i>РГЗ</i> | <i>Задание на зачете</i> |
| 12 | <i>ИПК-1.2. Имеет навыки использования современных языков программирования для разработки программного обеспечения</i> | Владет навыками практического программирования | <i>КР</i> | <i>Задание на зачете</i> |

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Примерный перечень индивидуальных заданий

1-й семестр

1. Факториалом числа натурального числа n называется произведение чисел от 1 до n включительно. Факториалом нуля называют единицу. Написать программу нахождения факториала данного числа. Реализовать через рекурсию и без рекурсии. Вывести на экран факториалы десяти первых чисел.
2. Написать программу нахождения факториала данного числа с проверкой на максимально возможное значение, которое может быть найдено. В случае переполнения памяти выдавать сообщение о невозможности нахождения факториала.
3. Число размещений без повторений A_n^k и число сочетаний без повторений C_n^k могут быть найдены соответственно по формулам $A_n^k = n!/(n-k)!$ и $C_n^k = n!/(k!(n-k)!)$. Напишите программу для нахождения данных величин при известных n и k .
4. Последовательность 1,1,2,3,5,8,... состоит из чисел Фибоначчи. Каждый элемент, начиная с третьего, равен сумме двух предыдущих. Найдите n -е число Фибоначчи. Реализовать вариант с рекурсией и вариант без рекурсии. Вывести на экран n строк из символов "*". Количество символов в строке с номером k равно k -ому числу Фибоначчи. Вывести на экран все числа Фибоначчи, меньшие данного числа.
5. Найти приближительные значения чисел e и π с, используя разложение в ряд.

6. Найти квадрат данного натурального числа, если квадрат любого натурального числа n равен сумме n первых нечетных чисел. Составить таблицу квадратов натуральных чисел от 1 до 10000. Куб любого натурального числа n равен сумме n нечетных чисел, следующих по порядку за числами, сумма которых составила куб числа $n-1$. С помощью этого факта составьте таблицу кубов натуральных чисел от 1 до 100.
7. Написать программу нахождения целой части и остатка кубического корня из натурального числа (целочисленное извлечение кубического корня). Обобщить на корень произвольной натуральной степени, большей двух.
8. Написать несколько вариантов целочисленного извлечения квадратного корня из натурального числа. Сравнить способы по времени выполнения, количеству выполненных операций и диапазону применения.
9. Арифметический квадрат. Заполнить квадратную матрицу $n \times n$ так, чтобы все числа первого столбца и первой строки равны 1, а каждое из оставшихся чисел равно сумме верхнего и левого соседей. Вывести на экран матрицу данного размера.
10. Треугольник Паскаля. Вывести на экран треугольник Паскаля из n строк. Придумать структуру данных для хранения треугольника Паскаля (например, стандартная матрица, что, однако, не экономно). Реализовать показ треугольника по данным из этой структуры.
11. Даны три числа. Определите, можно ли из отрезков с такими длинами составить треугольник. Определите вид треугольника (прямоугольный, тупоугольный, остроугольный), если он существует. Даны числа \min и \max . Найдите все треугольники с целочисленными длинами сторон от \min до \max включительно. Напишите программу, которая определяет, можно ли из четырех отрезков с данными длинами a , b , c и d составить прямоугольник.
12. Пифагоровы числа. Три натуральных числа a , b и c образуют пифагорову тройку, если $c^2 = a^2 + b^2$. Пифагорова тройка называется основной, если наибольший общий делитель ее чисел равен единице. Например, 3, 4, 5 - основная тройка, 6, 8, 10 - производная тройка. Найдите все основные пифагоровы тройки, числа в которых не превышают данное число \max . Напишите программу нахождения всех решений на отрезке $[2; 100]$ уравнения $x^2 + y^2 = zn$, n - известное натуральное число. Треугольники, у которых длины сторон и площадь являются натуральными числами, называются треугольниками Герона. Например, площадь треугольника со сторонами 13, 14 и 15 равна 84. Найдите n треугольников Герона. Найдите n треугольников Герона, длины сторон каждого из которых являются последовательными числами (например, 13, 14, 15). Найдите n треугольников Герона, у которых площадь равна периметру.
13. Дан прямоугольник, длины сторон которого выражаются целыми числами. Найдите количество квадратов (длины сторон каждого квадрата целые), на которые можно разрезать данный прямоугольник при условии, что при разрезании каждый раз отрезается квадрат наибольшей площади со стороны, общей стороне текущего прямоугольника.
14. Найти все прямоугольники с целыми длинами сторон данной целочисленной площади. Например, для площади, равной 12, подходят три прямоугольника 1 x 12, 2 x 6, 3 x 4. Напишите программу для нахождения всех различных прямоугольных параллелепипедов, объем которых равен V , а ребра выражены натуральными числами. Параллелепипеды, получающиеся один из другого, если поменять ребра местами, считаются одинаковыми.
15. Напишите программу бинарного поиска по ключу в массиве целых чисел. Решение
16. Оптический танк. Танк должен выехать с базы, пересечь пустынную и болотистую местность и прибыть на пост. Препятствий по пути нет, танк может двигаться в любом направлении. Известно, что прямая, соединяющая базу и пост, проходит по

- обеим территориям. Определите путь, по которому танк приедет на пост быстрее всего.
17. Вход. Первая строка текста содержит два числа - скорости танка по пустыне и болоту (м/с). Вторая строка содержит координаты базы, третья - координаты поста. Известно, что ось Ox разделяет пустынную и болотистую территории (пустынная наверху, болотистая внизу), координата y базы положительна, y поста - отрицательна. Все числа действительные.
 18. Выход. Вывести два числа: абсциссу координаты места, в котором танк пересекает границу пустынной и болотистой местности, и время движения от базы до поста. Оба числа выводить как дробные, с возможной ошибкой не более 10^{-5} .
 19. Написать слияние двух упорядоченных массивов, то есть итоговый массив должен быть упорядоченным.
 20. Слияние файлов. В тексте записана неубывающая последовательность положительных чисел. Два таких текста нужно слить в один, то есть получить новый текст, содержащий все исходные числа в порядке неубывания. Вход (каждый текст состоит из нескольких строк или является пустым. Числовые константы могут находиться в нескольких строках и внутри строки разделены пробелами). Выход. Строка текста, в которой константы разделены пробелами.
 21. Решить задачу 20 при условии, что задан список файлов с текстами, количество которых может достигать девяти.
 22. Выполнить сортировку пузырьком целочисленного массива.
 23. Выполнить сортировку выбором целочисленного массива.
 24. Реализуйте сортировку слиянием для массива целых чисел двумя способами (рекурсивный и без рекурсии)
 25. Реализуйте алгоритм быстрой сортировки для массива целых чисел.
 26. Реализуйте пирамидальную сортировку целочисленного массива.
 27. Строка состоит из клеток, пронумерованных от 1 до n . Состояние клетки можно изменить - если она пуста, поставить в нее шашку (занять ее), иначе убрать из нее шашку (освободить ее). Вначале строка пуста. Нужно занять все клетки, соблюдая следующее правило. Изменение клетки допустимо, если она имеет номер 1 или расположена непосредственно после занятой клетки, имеющей минимальный номер среди занятых клеток.
 28. Вход. Целое $n, 1 \leq n \leq 15$.
 29. Выход. Последовательность элементов вида $+i$ или $-i$, обозначающих соответственно занять клетку i и освободить клетку i .
 30. Пример. При $n=3$ выход имеет вид $+1+2-1+3+1$.
 31. Даны a и N . Вычислите a^N без использования логарифма и экспоненты.
 32. Написать программу, рисующую снежинку Коха порядка n .
 33. Написать программу, рисующую треугольник Серпинского порядка n .
 34. Напишите программу построения драконовой ломаной порядка n .
 35. С помощью рекурсии найдите сумму цифр данного натурального числа.
 36. Напишите рекурсивную функцию, печатающую цифра натурального числа в а) прямом; б) обратном порядке.
 37. Известно, что $F(n)=n-10$ при $n>100$ и $F(n)=F(F(n+11))$ при $n \leq 100$. Найдите $F(n)$ при $n < 200$.
 38. Заданы две строки a и b с длинами m и n . Найдите максимальную длину их общей подпоследовательности символов. Например, для "орел" и "осел" это "оел" длины 3.
 39. Дано число. Если оно больше 10, то увеличьте его на 100, иначе уменьшите на 30.
 40. Дано натуральное число. Если оно четное, то уменьшите его в 2 раза, иначе увеличьте в 3 раза.

41. Дано число. Если оно не меньше 50, то выведите квадрат этого числа, если же это число больше 10 и меньше 30, то выведите ноль, в остальных случаях выведите слово "Ошибка"
42. Дано два числа. Вывести наибольшее из них.
43. Дано два числа. Вывести 'Да', если они отличаются на 100, иначе вывести 'Нет'
44. Дано два числа. Вывести 'Да', если они отличаются не более чем на 20, иначе вывести 'Нет'.
45. Дан номер месяца. Вывести название поры года (весна, лето и так далее) или слово 'Ошибка', если месяца с таким номером не существует.

2-й семестр

1. Дана строка. Если она начинается на 'abc', то заменить их на 'www', иначе добавить в конец строки 'zzz'.
2. Дана строка. Если ее длина больше 10, то оставить в строке только первые 6 символов, иначе дополнить строку символами 'o' до длины 12.
3. В данной строке найти количество цифр.
4. Замените в строке все вхождения 'word' на 'letter'.
5. Удалите в строке все буквы 'x', за которыми следует 'abc'.
6. Найдите количество вхождения 'aba' в строку.
7. Дана строка, состоящая из слов, разделенных символами, которые перечислены во второй строке. Показать все слова.
8. Дана строка. Преобразовать все символы кроме первого в нижний регистр.
9. Даны два файла со словами, разделенными пробелами. Создать новый файл, содержащий: а) строки, которые встречаются только в первом файле; б) строки, которые встречаются в обоих файлах; в) строки, которые встречаются в каждом файле более двух раз.
10. Даны два файла, состоящие из предложений. Создать третий файл, содержащий все предложения, которые есть хотя бы в одном из файлов. Повторы не добавлять в третий файл.
11. Дан файл со словами. Упорядочить слова по алфавиту.
12. Дан файл. Каждая строка содержит имя, пароль и email, разделенные символами ':' (двоеточие). Удалить строки с повторами email. Удалить строки, в которых имена совпадают.
13. Написать функцию, которая будет показывать список всех файлов в данной папке (поиск файлов происходит и во всех вложенных уровнях).
14. Пользователю предлагается ввести имя каталога в локальной файловой системе сервера. Сценарий RHP выводит содержимое этого каталога в следующем формате: пиктограмма, указывающая на тип файла, имя файла, размер (или специальная пометка, если файл является каталогом), дата и время последней модификации.
15. Дан файл с html-кодом. Показать все ссылки из этого кода.
16. Дан файл с текстом. Показывать содержимое файла по страницам с навигацией. На каждую страницу помещается заданное пользователем количество символов.
17. Сделать тестирующую программу. Программа должна состоять из четырёх страниц по два вопроса на каждой. Варианты ответов обозначить радиальными переключателями – по три варианта на каждый вопрос. Список номеров правильных ответов хранится в текстовом файле, номера разделены пробелами. На пятой странице программы выводится количество правильных ответов и содержится кнопка «Пройти заново», которая возвращает пользователя на первую страницу.

18. Пользователь загружает текстовый файл со списком ссылок. Добавить в базу (файл на сервере) из этого файла только те ссылки, которых нет ни в базе, ни в файле с запрещенными ссылками.
19. Написать гостевую книгу. Пользователи указывают свой email и текст сообщения. При отображении на сайте сообщений пользователей указывается дата написания текста. Для хранения сообщений использовать файлы.
20. Написать доску объявлений. Пользователь указывает свой email, текст объявления, заголовок объявления и выбирает категорию объявления. Для хранения объявлений использовать файлы.
21. Создать личную галерею изображений. Возможности: загрузка фото, указание комментариев к фото и выбор категории.
22. В некотором каталоге на сервере содержатся файлы и папки. Папки, в свою очередь, могут содержать вложенные папки и файлы. Уровень вложенности неограничен. Необходимо построить дерево каталогов в виде многоуровневого списка. При этом маркером перед именем папки должен быть значок папки Windows, а маркером перед именем файла – значок программы, которая этот файл открывает по умолчанию. В каталогах могут содержаться файлы со следующими расширениями: doc, xls, htm, html, txt, css.
23. Робот Работа в интерактивном режиме. Команды Робота. Лесенка для робота. Алгебра логики на практике. Коридорные задачи: на циклы и условия. Вложенные циклы: сложные коридоры. Считаем шаги. Задачи на подсчет шагов. Ищем выход из лабиринта.
24. Черепаха Черепашка Python Turtle. Черепаха как объект. Свойства черепахи. Задачи на циклы: многоугольники, звезды. Вложенные циклы. Написание функций. Функции без параметров и с параметрами. Списки в Python. Операции со списками. Генераторы списков. Кортежи в Python. Список кортежей. Случайное движение черепахи, модуль random. Заливка полигонов.
25. Картина GitHub: регистрация и основы использования. Библиотека graphics Дж.Зелле. Геометрические примитивы. Создание картины по образцу.
26. Рефакторинг картины Структурное программирование. Проектирование сверху-вниз. Рефакторинг.
27. Анимация картины Проработка интерфейсов функций. Документация функций.
28. Игра «Поймай шарик» Введение в Tkinter. Событийная модель построения приложения. Виджеты, события и обработчики событий. Свойства и упаковка виджетов. Tkinter Canvas: методы, идентификаторы и теги. Создание простой игры: кликать мышкой на случайно появляющихся шариках. Виджет кнопки для перезапуска игры.
29. Игра «Поймай шарик» Возбуждение событий по времени. Движение шариков с отражением от стен. Подсчет игровых очков. Переменные с обратной связью в tkinter. Ползунок, управляющий скоростью движения шариков.
30. Игра «Пушка» Простое объектно-ориентированное программирование. Создание и инициализация класса. Моделирование кинематики материальной точки (снаряда в поле силы тяжести).
31. Игра «Пушка». Модульный рефакторинг Модульное программирование. Создание модулей и пакетов. Документация проекта.
32. Игра «Пушка» Введение в ООП проектирование. Проработка интерфейсов и контрактов функций и ответственности классов. Отношения между классами. Наследование и композиция. Ассоциация. Диаграмма классов UML.

33. Игра «Артиллерия» Проектирование игры в стиле artillery game. Командная работа. Учимся взаимодействовать через GitHub. Осваиваем: git branch, git merge, pull request. Механизм review.
34. Игра «Артиллерия»

3-й семестр

1. Задача связности. Предположим, что имеется последовательность пар целых чисел (пара p - q интерпретируется в значении " p связано с q "). Если p связано с q , а q связано с r , то p связано с r . Задача состоит в написании программы для исключения лишних пар из набора: когда программа получает очередную пару p - q , она должна добавлять эту пару только в том случае, если из предыдущих пар не следует, что p связано с q . Пример: 3-4, 4-9, 8-0, 2-3, 5-6. К данному списку 2-9 не добавляем, так как 2-3-4-9.
2. Создать список из заданного количества элементов. Выполнить циклический сдвиг этого списка на N элементов вправо или влево.
3. Реализовать с помощью однонаправленного линейного списка операцию умножения длинного целого числа на N (целую цифру, то есть целое число от 0 до 9).
4. Исходный файл содержит два набора положительных значений, между наборами стоит отрицательное значение. Построить два списка $S1$ и $S2$, элементы которых содержат значение из наборов 1 и 2 соответственно. Элементы расположить по возрастанию. Выполнить соединение списков.
5. Создать два стека и поменять информацию местами.
6. Исходный файл содержит наименование некоторых объектов. Построить список, элементы которого содержат наименование и шифры данных объектов, причем элементы списка должны быть упорядочены по возрастанию значений шифров. Выполнить «сжатие» списка, удалив продублированные наименования объектов.
7. Задано множество точек на двухмерной плоскости. Найти три разные точки, которые составляют треугольник наибольшего периметра. Структуру данных "множество" реализовать в виде класса.
8. Задано множество точек на двухмерной плоскости. Найти такую точку, от которой сумма расстояния к другим точкам наименьшая. Структуру данных "множество" реализовать в виде класса.
9. На плоскости заданы N множеств по M точек в каждой. Найти среди точек первого множества такую точку, которая принадлежит наибольшему количеству множеств. Структуру данных "множество" реализовать в виде класса.
10. Создать список из стихов одного автора. Выполнить сортировку списка по возрастанию размера стихов.
11. Ввести некоторое число и записать его цифры в стек. Вывести число, у которого цифры идут в обратном порядке. Предусмотреть возможность введения произвольного количества чисел.
12. Система состоит из трех процессоров $P1$, $P2$, $P3$, очереди F , стека S и распределителя задач R . В систему поступают запросы на выполнение задач трёх типов – $T1$, $T2$ и $T3$, каждая для своего процессора. Поступающие запросы ставятся в очередь. Если в начале очереди находится задача T_i и процессор P_i свободен, то распределитель R ставит задачу на выполнение в процессор P_i , а если процессор P_i занят, то распределитель R отправляет задачу в стек и из очереди извлекается следующая задача. Если в вершине стека находится задача, процессор которой в данный момент свободен, то эта задача извлекается и отправляется на выполнение. Осуществить моделирование работы данной системы.
13. Задано два списка, которые содержат результаты N -измерений тока (I) и напряжения (U) на неизвестном сопротивлении (R). Найти приближенное значение R методом наименьших квадратов.

14. Организовать хеш-таблицу с открытой адресацией, используя процедуру поиска и вставки по ключу. Для формирования начального хеш-адреса использовать метод деления, а затем при возникновении коллизии процедуру квадратичного исследования.

15. Написать программу, определяющую является ли двоичное дерево деревом поиска.

16. Имеется дерево, корень которого соответствует основателю рода. Сыновья каждой вершины задают сыновей и дочерей соответствующего человека. Указываются имена двух человек (например, А и В) . Сообщить, какая из следующих ситуаций имеет место:

- 1) А предок В;
- 2) В предок А;
- 3) А и В имеют ближайшего общего предка С.

17. При запуске программы в консоли должно открыться меню с выбором "Работа со списками" или "Работа с деревьями". При выборе какого-либо из пунктов должно открыться подменю. Подменю "Работа со списками" должно включать в себя пункты: "Включить новый элемент после i -ого по номеру элемента", "Исключить элемент после i -ого по номеру элемента", "Поменять местами i -ый и i -ый+1 по номерам элементы". Подменю "Работа с деревьями" должно выполнять следующие действия: создание бинарного дерева (дерева поиска) в диалоге с пользователем (способ формирования дерева: с учётом значений ключа), обход бинарного дерева каждым из трёх способов с выдачей на экран содержимого информационных полей, включение элемента в бинарное дерево (согласно алгоритму формирования дерева), удаление заданного узла из дерева (без поддерев), удаление дерева с освобождением памяти, вывод дерева на экран. А так же определить количество листьев на заданном уровне дерева, определить количество узлов (не листьев) бинарного дерева.

18. Составить программу для построения и обработки дерева общего вида или упорядоченного двоичного дерева, содержащего узлы целого типа. После того как дерево, создано с ним можно выполнять следующие действия:

– добавление нового узла (для двоичного дерева положение нового узла определяется в соответствии с требованием сохранения порядка, для дерева общего вида должен задаваться путь до добавляемого узла, добавляемый узел становится младшим сыном).

– текстовая визуализация дерева (значение каждого узла выводится на отдельной строке, с отступом пропорциональным глубине узла (шаг отступа равен двум пробелам), в порядке старшинства узлов).

– удаление узла (двоичное дерево перестраивается в соответствии с требованием сохранения целостности и порядка; для дерева общего вида удаляется все поддерево, исходящее из удаляемого узла; должно быть предусмотрено корректное освобождение памяти). Определение числа вершин, степень которых совпадает со степенью дерева.

19. Задан граф (орграф) в виде матрицы смежности. Составить программу

- а) проверки, есть ли в графе петли;
- б) поиска в графе изолированной вершины (не смежной с другими);
- в) определения степени графа;
- г) получения последовательности ребер.

20. Запрограммировать структуры данных и операции над ними для представления графа в виде

- а) последовательности ребер (дуг),
- б) векторов смежности,
- в) матрицы смежности,
- г) списков смежности,
- д) матрицы весов,
- е) сети,

ж) матрицы инцидентности,

з) списковой структуры.

Необходимые детали представления уточнить самостоятельно.

21. Дан граф (орграф). Составить описание данных для его представления и фрагмент программы

а) получения кратчайших путей и расстояний между всеми вершинами по алгоритму Флойда и вывода кратчайшего пути от вершины А до вершины В.

б) получения матрицы связности по алгоритму Уоршалла и вывода номеров вершин каждой компоненты связности графа.

22. Разработать программу, выполняющую обработку данных на основе рекурсивных алгоритмов.

23. Дано N городов и известны расстояния между ними. Не все города связаны друг с другом дорогой. Найти все возможные маршруты из города А в город В (ни в один город не заходить дважды). Определить самый длинный и самый короткий маршрут.

24. Система двусторонних дорог такова, что для любой пары городов можно указать соединяющий их путь. Найдите такой город, сумма расстояний от которого до остальных городов минимальна.

25. Для заданной системы двусторонних дорог определите, можно ли, закрыв какие-нибудь три дороги, запретить перемещение из города А в город Б.

26. Дана система двусторонних дорог. N-периферией называется множество городов, расстояние от которых до выделенного города (столицы) больше N. Для данного N определите N-периферию.

27. Определите, можно ли в данной системе двусторонних дорог проехать из города А в город В так, чтобы посетить город С и не проезжать ни по какой дороге более одного раза.

28. За проезд по каждой дороге взимается некоторая пошлина. Найдите путь из города А в город В с минимальной величиной $S+P$, где S - сумма длин дорог пути, а P - сумма пошлин проезжаемых дорог.

29. Найдите медиану графа, то есть такую его вершину, что сумма расстояний от нее до остальных вершин минимальна.

30. Разработать класс В-дерево для хранения объектов, обладающих ключом с операцией сравнения. При реализации следует пользоваться статическим полиморфизмом на основе шаблонов.

31. Компьютер загадывает число от 1 до n. У пользователя k попыток отгадать. После каждой неудачной попытки компьютер сообщает меньше или больше загаданное число. В конце игры текст с результатом (или “Вы угадали”, или “Попытки закончились”).

32. Найти количество различных элементов массива. Пример: для 1 4 5 1 1 3 ответ 4.

33. Дан массив. Перемешать его элементы случайным образом так, чтобы каждый элемент оказался на новом месте.

34. Куб состоит из n^3 прозрачных и непрозрачных элементарных кубиков. Имеется ли хотя бы один просвет по каждому из трех измерений? Если это так, вывести координаты каждого просвета. Куб задается трехмерной матрицей из 0 и 1.

35. Дано предложение. Заменить группы пробелов одиночными, крайние пробелы удалить. Все слова перевести в нижний регистр, первые буквы сделать заглавными.

36. Дана строка. Заменить все ссылки и email на ***** (количество звездочек равно длине заменяемого фрагмента). Примеры ссылок: www.site.com, <http://site.com> и т.п. Решить двумя способами: с использованием регулярных выражений и без. Сравнить скорости работы.

37. Пользователь вводит формулу вида $\max(a,b)$ или $\min(a,b)$, где a,b - целые числа или аналогичные выражения $\min()$, $\max()$. Найти значение выражения. Примеры: $\max(1,5) = 5$, $\max(4,\min(3,6)) = 4$, $\min(\max(1,\max(5,3)),\min(9,0)) = 0$.

38. Уроки по математике проводятся через день. Если урок попадает на воскресенье, то переносится на понедельник. Пользователь вводит дату первого урока. Сгенерируйте расписание из дат на текущий месяц.

39. Дан файл с логинами и паролями. Найдите топ10 самых популярных паролей.

40. Дан файл с пунктами меню (id, название, id родителя). Если id родителя равно 0, то родителя не существует. Показать полное меню с отступами. Пользователь вводит id пункта. Показать цепочку из пунктов меню до этого пункта. Уровней вложенности в меню может быть любое количество.

41. Дан каталог книг. Про книгу известно: уникальный номер, автор, название, год издания. Реализовать CRUD, показ всех книг на экран и поиск по каждому из полей. Сделать меню для работы с каталогом.

4-й семестр

1. Создайте структуру с именем student, содержащую поля: фамилия и инициалы, номер группы, успеваемость (массив из пяти элементов). Создать массив из десяти элементов такого типа, упорядочить записи по возрастанию среднего балла. Добавить возможность вывода фамилий и номеров групп студентов, имеющих оценки, равные только 4 или 5.

2. Создайте структуру с именем train, содержащую поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа train, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массив по пункту назначения, причем поезда с одинаковыми пунктами назначения должны быть упорядочены по времени отправления.

3. Создать класс с двумя переменными. Добавить функцию вывода на экран и функцию изменения этих переменных. Добавить функцию, которая находит сумму значений этих переменных, и функцию, которая находит наибольшее значение из этих двух переменных.

4. Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние. Написать программу, демонстрирующую все возможности класса.

5. Создать класс с двумя переменными. Добавить конструктор с входными параметрами. Добавить конструктор, инициализирующий члены класса по умолчанию. Добавить деструктор, выводящий на экран сообщение об удалении объекта.

6. Создать класс, содержащий динамический массив и количество элементов в нем. Добавить конструктор, который выделяет память под заданное количество элементов, и деструктор. Добавить методы, позволяющие заполнять массив случайными числами, переставлять в данном массиве элементы в случайном порядке, находить количество различных элементов в массиве, выводить массив на экран.

7. Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть контроль выхода за пределы массива, возможность обращения к отдельным строкам массива по индексам, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, а также вывод на экран элемента массива по заданному индексу и всего массива.

8. Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения,

вычитания и умножения многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена.

9. Построить три класса (базовый и 3 потомка), описывающих некоторых хищных животных (один из потомков), всеядных (второй потомок) и травоядных (третий потомок). Описать в базовом классе абстрактный метод для расчета количества и типа пищи, необходимого для пропитания животного в зоопарке.

10. а) Упорядочить всю последовательность животных по убыванию количества пищи. При совпадении значений – упорядочивать данные по алфавиту по имени. Вывести идентификатор животного, имя, тип и количество потребляемой пищи для всех элементов списка.

b) Вывести первые 5 имен животных из полученного в пункте а) списка.

с) Вывести последние 3 идентификатора животных из полученного в пункте а) списка.

d) Организовать запись и чтение коллекции в/из файл.

e) Организовать обработку некорректного формата входного файла.

11. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям.

12. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников.

13. Создать класс для хранения комплексных чисел. Реализовать операции над комплексными числами: сложение, вычитание, умножение, деление, сопряжение, возведение в степень, извлечение корня. Предусмотреть возможность изменения формы записи комплексного числа: алгебраическая форма, тригонометрическая форма, экспоненциальная форма.

14. Составить описание класса для представления времени. Предусмотреть возможности установки времени и изменения его отдельных полей (час, минута, секунда) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Создать методы изменения времени на заданное количество часов, минут и секунд.

15. Составить описание класса для вектора, заданного координатами его концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

16. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, вычисления площади, периметра и точки пересечения медиан. Описать свойства для получения состояния объекта.

17. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник) со своими методами вычисления площади и периметра. Создать массив n фигур и вывести полную информацию о фигурах на экран.

18. Класс Покупатель: Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета; Конструктор; Методы: установка значений атрибутов, получение значений атрибутов, вывод информации. Создать массив объектов данного класса. Вывести список покупателей в алфавитном порядке и список покупателей, у которых номер кредитной карточки находится в заданном диапазоне.

19. Класс Абонент: Идентификационный номер, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Дебет, Кредит, Время междугородных и городских переговоров; Конструктор; Методы: установка значений атрибутов, получение значений атрибутов, вывод информации. Создать массив объектов данного класса. Вывести сведения относительно абонентов, у которых время городских переговоров превышает заданное. Сведения относительно абонентов, которые пользовались междугородной связью. Список абонентов в алфавитном порядке.

20. Задача на взаимодействие между классами. Разработать систему «Автобаза». Диспетчер распределяет заявки на Рейсы между Водителями и назначает для этого Автомобиль. Водитель может сделать заявку на ремонт. Диспетчер может отстранить Водителя от работы. Водитель делает отметку о выполнении Рейса и состоянии Автомобиля.

21. Задача на взаимодействие между классами. Разработать систему «Железнодорожная касса». Пассажир делает заявку на станцию назначения, время и дату поездки. Система регистрирует Заявку и осуществляет поиск соответствующего Поезда. Пассажир делает выбор Поезда и получает Счет на оплату. Кассир вводит номера Поездов, промежуточные и конечные станции, цены.

22. Задача на взаимодействие между классами. Разработать систему «Интернет-магазин». Товаровед добавляет информацию о Товаре. Клиент делает и оплачивает Заказ на Товары. Товаровед регистрирует Продажу и может занести неплательщика в «черный список».

23. Задача на взаимодействие между классами. Разработать систему «Платежи». Клиент имеет Счет в банке и Банковскую карту (КК). Клиент может оплатить Заказ, сделать платеж на другой Счет, заблокировать КК и аннулировать Счет. Администратор может заблокировать КК за превышение платежа.

24. Задача на взаимодействие между классами. Разработать систему «Вступительные экзамены». Абитуриент регистрируется на Факультет, сдает Экзамены. Преподаватель выставляет Оценку. Система подсчитывает средний бал и определяет Абитуриента, зачисленного в учебное заведение.

25. Разработать класс «Калькулятор логарифмов» с возможностью сложения, вычитания, умножения, деления, возведения в степень и перехода к другому основанию. Программа должна выполнять ввод данных, проверку правильности введенных данных, выдачу сообщений в случае ошибок. Протокол работы калькулятора записать в файл. Предусмотреть возможность просмотра файла из программы.

26. Аквариум

1) Определите объект TFish - аквариумная рыбка. Рыбка имеет координаты, скорость, размер, цвет, направление движения. Методами объекта являются:

Init - устанавливает значения полей объекта и рисует рыбу на экране методом Draw.
Draw - рисует рыбу (виртуальный метод).

Look - проверяет несколько точек на линии движения рыбы. Если хоть одна из них отличается по цвету от воды, возвращается её цвет и расстояние до рыбы.

Run - перемещает рыбу в текущем направлении на расстояние, зависящее от текущей скорости рыбы. Иногда случайным образом меняет направление движения рыбы. Если рыба видит препятствие, направление движения меняется, пока препятствие не исчезнет из поля зрения рыбы.

2) Определите объект Taquarium, который является местом обитания рыб. Он представляет собой область экрана, наполненную водой.

Методы:

Init - включает графический режим, заполняет аквариум водой, скалами и рыбами.
Run - организует бесконечный цикл, в котором выполняется метод Run всех обитателей аквариума.

Done - выключает графический режим.

3) Определите два объекта *Trike* и *Tkarp*, которые наследуют объект *Tfish*.

4) Карпы и щуки должны быть объединены в стаи. Стая - это связанный список рыб в динамической памяти. Для связи в объектах *Trike* и *Tkarp* используйте поле *Next* - указатель на следующую рыбу в стае. Аквариум должен быть владельцем двух стай. Пользователь может пополнять стаи, вводя рыб с клавиатуры.

5) Щуки должны поедать карпов, как только они их увидят. Необходимо установить, какого именно карпа видит щука (ближайший по координатам карп к щуке, найденный карп удаляется из стаи).

27. Разработка приложения для предметной области «Учёт товаров в магазине»: разработать приложение, позволяющее собирать и накапливать сведения о поступлении и реализации товаров некоторого магазина. Структура приложения обязательно должна включать следующие классы: товар, производитель, документ, поступление товара, реализация товара и др.

28. Разработка приложения для предметной области «Организация учебного процесса в ВУЗе»: разработать приложение, позволяющее собирать и накапливать сведения об организации и диспетчеризации учебного процесса в ВУЗе. Структура приложения обязательно должна включать следующие классы: академическая группа, специальность, дисциплина, аудитория, преподаватель и др.

29. Транспортное агентство: разработайте программу, имитирующую работу трансагентства. Трансагентство имеет сеть филиалов в нескольких городах. Транспортировка грузов осуществляется между этими городами тремя видами транспорта: автомобильным, железнодорожным и воздушным. Любой вид транспортировки имеет стоимость единицы веса на единицу пути и скорость доставки. Воздушный транспорт можно использовать только между крупными городами, этот вид самый скоростной и самый дорогой. Кроме того, воздушный транспорт зависит от погоды. Доставить груз воздушным путем можно только при условии хорошей погоды одновременно в городах отправки и назначения. Хорошая или плохая погода задается случайным образом. Железнодорожный транспорт можно использовать между крупными и средними городами, этот вид самый дешевый. Автомобильный транспорт можно использовать между любыми городами. Заказчики через случайные промежутки времени обращаются в один из филиалов трансагентства с заказом на перевозку определенной массы груза и возможным пожеланием о скорости/цене доставки. Трансагентство организует отправку грузов одним из видов транспорта с учетом пожеланий клиента. Оплату трансагентство получает только после успешной доставки груза. Между некоторыми городами для железнодорожного и/или автомобильного транспорта имеются скоростные магистрали, на которых скорость соответствующего вида транспорта увеличивается с заданным коэффициентом. При перевозке грузов могут происходить аварии, при этом вероятность аварии на автотранспорте больше, чем на железнодорожном транспорте, а авиатранспорт имеет аварийность очень низкую. На скоростных магистралях вероятность аварии меньше, чем на обычных дорогах. При аварии трансагентство возвращает заказчику двойную стоимость перевозки.

Процесс имитации может быть остановлен пользователем программы для просмотра параметров объектов:

Доход трансагентства, в том числе с разбивкой по видам транспорта и городам.

Среднее время доставки груза, в том числе с разбивкой по видам транспорта и городам.

Потери, связанные с плохой погодой.

Потери, связанные с аварийностью, в том числе с разбивкой по видам транспорта и по видам дорог.

Доход на тонно-километр скоростных магистралей в сравнении с таким же доход на обычных дорогах.

Список исполняемых заказов с возможностью сортировки по городам, видам транспорта, стоимости перевозки.

Список задерживаемых заказов в связи с плохой погодой.

(Разработать и реализовать классы, которые являются основными в задании. Для сдачи лабораторной работы необходимо создать проект, в котором демонстрируется работа объектов созданных классов. Должны быть продемонстрированы выполнение конструктора, всех доступных методов и деструктора.)

30. Класс Дробное число со знаком (Fractions). Число должно быть представлено двумя полями: целая часть - длинное целое со знаком, дробная часть - беззнаковое короткое целое. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения. В функции main проверить эти методы.

31. Класс Деньги для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа unsigned char - для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. В функции main проверить эти методы.

32. Класс Равнобочная трапеция, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура равнобочной трапецией; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N трапеций, найти количество трапеций, у которых площадь больше средней площади.

33. Описать базовый класс Строка. Обязательные поля класса: указатель pChar хранит адрес динамически выделенной памяти для размещения символов строки;

значение типа word хранит длину строки в байтах.

Реализовать обязательные методы следующего назначения:

-конструктор без параметров;

-конструктор, принимающий в качестве параметра строковый литерал; п конструктор, принимающий в качестве параметра символ;

-метод получения длины строки;

-метод очистки строки (делает строку пустой);

-деструктор.

Описать производный от Строка класс Комплексное число.

Строки данного класса должны состоять из двух полей разделочных символом i. Первое поле задает значение действительной части числа, а второе - значение мнимой. Каждое из полей может содержать только символы десятичных цифр и символы - и +, определяющие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будет встречен любой символ, отличный от допустимых, объект класса Комплексное число должен принимать нулевое значение. Примеры строк: 33i12, -7U00, +5i-21.

Для класса Комплексное_число определить следующие методы:

-проверка на равенство;

-сложение чисел;

-умножение чисел.

Написать тестовую программу, которая демонстрирует работу методов базового и производного классов.

Описать шаблонный класс List для работы с однонаправленными списками в динамической памяти.

Для объектов класса List определить операции проверки списка на пустоту, добавления элемента в начало списка, в конец списка, подсчет числа вхождений элемента в список, удаление элемента из списка.

Продемонстрировать работу с шаблонным классом для списка с целыми элементами и с элементами-строками.

5-й семестр

1. В директории лежат входные текстовые файлы, проименованные следующим образом: `in_<N>.dat`, где N - натуральное число. Каждый файл состоит из двух строк. В первой строке - число, обозначающее действие, а во второй - числа с плавающей точкой, разделенные пробелом.

Действия могут быть следующими:

- 1 - сложение
- 2 - умножение
- 3 - сумма квадратов

Необходимо написать многопоточное приложение, которое выполнит требуемые действия над числами и сумму результатов запишет в файл `out.dat`. Название рабочей директории передается в виде аргумента рабочей строки. В реализации приветствуется использование полиморфизма и паттернов проектирования.

2. Разработайте многопоточное приложение, выполняющее вычисление произведения матриц $A (m \times n)$ и $B (n \times k)$. Элементы c_{ij} матрицы произведения $C = A \times B$ вычисляются параллельно p однотипными потоками. Если некоторый поток уже вычисляет элемент c_{ij} матрицы C , то следующий приступающий к вычислению поток выбирает для расчета элемент c_{ij+1} , если $j < k$, и c_{i+1k} , если $j = k$. Выполнив вычисление элемента матрицы-произведения, поток проверяет, нет ли элемента, который еще не рассчитывается. Если такой элемент есть, то приступает к его расчету. В противном случае отправляет (пользовательское) сообщение о завершении своей работы и приостанавливает своё выполнение. Главный поток, получив сообщения о завершении вычислений от всех потоков, выводит результат на экран и запускает поток, записывающий результат в конец файла-протокола. В каждом потоке должна быть задержка в выполнении вычислений (чтобы дать возможность поработать всем потокам). Синхронизацию потоков между собой организуйте через критическую секцию или мьютекс.

3. Задача в терминах C++. При нажатии кнопки "Start" диалоговое приложение запускает консольное приложение. Последующие нажатия кнопки "Start" должны привести к созданию в консольном приложении N новых рабочих потоков, где N - значение из поля с числовым счетчиком.

Нажатие кнопки "Stop" должно привести к закрытию последнего созданного рабочего потока в консольном приложении, а в случае отсутствия рабочих потоков - к его завершению. Консольное приложение также должно завершиться и при завершении диалогового. Диалоговое приложение должно отслеживать, закрыл ли пользователь консольное приложение самостоятельно. Следующее после закрытия консоли нажатие "Start" возобновляет рабочий цикл.

6-й семестр

После запуска консоли выпадающий список содержит строки "Все потоки" и "Главный поток", по мере создания новых потоков в него добавляются строки, содержащие их номера (разумеется, при удалении потоков удаляются и соответствующие им строки). Корректировка выпадающего списка осуществляется лишь после получения подтверждения от консоли об успешном создании потоков.

Взаимодействие между приложениями реализовать с помощью объектов ядра "события" (events).

Добавить в приложения передачу информации с помощью файла, отображаемого в память (memory mapped file). Реализовать транспортные функции в виде динамически подключаемой библиотеки с неявной загрузкой.

При нажатии кнопки "Send" диалоговое приложение пересылает консоли текст, введенный в поле ввода. Выпадающий список задает поток-адресат. При выборе строки "Все потоки" сообщение отправляется всем потокам, каждый поток считывает информацию независимо.

Главный поток выводит полученный текст на стандартный вывод (stdout), рабочие потоки создают текстовые файлы с именами <свой номер>.txt и дописывают в них полученный текст. Файл, отображаемый в память, защищается от совместного использования с помощью объекта синхронизации mutex.

Все действия подтверждаются сообщениями от консоли.

Начальная параллельная программа (печать идентификаторов процессов) - запуск (локальный, распределенный). Оценка времени выполнения программы, синхронизация, коллективные операции.

Численное интегрирование- создание проекта на основе готового исходного текста программы, настройка опций, компиляция, запуск.

Скалярное произведение векторов.

Умножение матриц, ленточный алгоритм (самостоятельное задание 2).

Умножение матриц, блочные схемы распределения данных (алгоритмы Фокса и Кеннона).

Умножение матриц при блочном способе распределения данных" (алгоритмы Фокса и Кеннона): схема реализации, теоретическая оценка эффективности, вычислительные эксперименты.

Скалярное произведение векторов. Глобальные и локальные данные потоков. Критические секции доступа к разделяемым данным. Синхронизация.

Умножение матриц (варианты распараллеливания вложенных циклов, самостоятельное задание 3).

Параллельная сортировка: алгоритмы пузырьковой сортировки, сортировки Шелла и быстрой сортировки (выполнение задания под руководством преподавателя).

Задачи обработки графов: построение минимального охватывающего дерева, поиск кратчайших путей.

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

1-й семестр

Для получения необходимой суммы баллов необходимо решить индивидуальные задания по следующим темам (для каждой темы указано количество баллов за задание):

1. Функции алгебры логики, условный оператор. (15 баллов)
2. Списки. (15 баллов)
3. Последовательности. (15 баллов)
4. Функции и рекурсия. (15 баллов)
5. Массивы. (15 баллов)
6. Сортировки. (20 баллов)

2-й семестр

Для получения необходимой суммы баллов необходимо решить индивидуальные задания по следующим темам (для каждой темы указано количество баллов за задание):

1. Работа со строками. (15 баллов)
2. Обработка запросов для текстов (15 баллов)
3. Создание собственных модулей. (20 баллов)
4. Файлы. (15 баллов)
5. Библиотека Matplotlib. Функции одной переменной. (15 баллов)
6. Библиотека Matplotlib. Функции двух переменных. (15 баллов)

3-й семестр

Для получения необходимой суммы баллов необходимо решить индивидуальные задания по следующим темам (для каждой темы указано количество баллов за задание):

1. Матричные и векторные операции. (20 баллов)
2. Работа с временными рядами (15 баллов)
3. Обработка изображений и текста (20 баллов)
4. Работа с датафреймами. (20 баллов)
5. Визуализация данных. (15 баллов)

4-й семестр

Для получения зачета необходимо подготовить 2 индивидуальных проекта с использованием всех основных возможностей объектно-ориентированного программирования (45 баллов за полностью готовый проект).

5-й семестр

Для получения зачета необходимо подготовить 2 индивидуальных проекта с использованием параллельных вычислений (45 баллов за полностью готовый проект).

6-й семестр

Для получения зачета необходимо подготовить 2 индивидуальных проекта с использованием распределенных вычислений (45 баллов за полностью готовый проект).

Критерии оценивания результатов обучения

Критерии оценивания по зачету:

Зачеты по дисциплине преследуют цель оценить работу студента за курс, получение теоретических и практических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом задач за семестр, индивидуального задания / проекта и ответов на вопросы зачета. За решение каждого индивидуального задания дается количество баллов, указанное в разделе «Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)». Если студент предоставляет программу, которая имеет недочеты, то есть проходит большинство тестов, но не все, то снимается от 1 до 5 баллов.

«зачтено»: студент набрал 80 баллов и более; студент владеет теоретическими знаниями по данному разделу и решил все индивидуальные задачи и индивидуальные задания.

«не зачтено»: студент набрал менее 80 баллов; материал не усвоен или усвоен частично.

Кроме того, за ответы на вопросы на зачете дается до 20 баллов.

0 баллов: непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы.

0-10 баллов: знание и понимание основных вопросов программы, студент указал направление решения индивидуальной задачи; частично ответил на два вопроса или достаточно полно ответил хотя бы на один вопрос.

11-15 баллов: твердые и достаточно полные знания всего программного материала, последовательные, правильные, конкретные ответы на поставленные вопросы при свободном реагировании на замечания по отдельным вопросам; достаточно полно ответил на два вопроса.

16-20 баллов: глубокие исчерпывающие знания всего программного материала, логически последовательные, полные, грамматически правильные и конкретные ответы на вопросы.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1. Учебная литература

1. Окулов, С. М. Программирование в алгоритмах: учебное пособие / С. М. Окулов. — 7-е изд. — Москва: Лаборатория знаний, 2021. — 386 с. — ISBN 978-5-93208-521-9. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/172252>

2. Окулов, С. М. Алгоритмы обработки строк: учебное пособие / С. М. Окулов. — 4-е изд. — Москва: Лаборатория знаний, 2020. — 258 с. — ISBN 978-5-00101-658-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/135553>

3. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2021. — 210 с. — (Высшее образование). — ISBN 978-5-534-14638-7. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/478098>

4. Зыков, С. В. Программирование. Функциональный подход: учебник и практикум для вузов / С. В. Зыков. — Москва: Издательство Юрайт, 2021. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470387>

5. Зыков, С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. — Москва: Издательство Юрайт, 2021. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470281>

5.2. Периодическая литература

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» www.biblioclub.ru
3. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных:

1. Web of Science (WoS) <http://webofscience.com/>
2. Scopus <http://www.scopus.com/>
3. ScienceDirect www.sciencedirect.com
4. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
5. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
6. "Лекториум ТВ" <http://www.lektorium.tv/>
7. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Информационные справочные системы:

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

Ресурсы свободного доступа:

1. КиберЛенинка (<http://cyberleninka.ru/>);
2. Курсы ведущих вузов России" <http://www.openedu.ru/>;
3. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
4. Онлайн-курсы и сертификаты от ведущих вузов мира <https://ru.coursera.org/>.

Собственные электронные образовательные и информационные ресурсы

КубГУ:

1. Среда модульного динамического обучения <http://moodle.kubsu.ru>
2. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://mschool.kubsu.ru/>
3. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
4. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
5. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

6. Методические указания для обучающихся по освоению дисциплины (модуля)

– *Общие рекомендации по самостоятельной работе обучающихся.*

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301).

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений".

Положение о самостоятельной работе студентов (утверждено приказом № 272 КубГУ от 03 марта 2016 г.).

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Материально-техническое обеспечение по дисциплине (модулю)

| Наименование специальных помещений | Оснащенность специальных помещений | Перечень лицензионного программного обеспечения |
|---|---|---|
| Учебные аудитории для проведения занятий лекционного типа | Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер | Microsoft Windows Microsoft Office Professional Plus |
| Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации | Мебель: учебная мебель | |

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

| Наименование помещений для самостоятельной работы обучающихся | Оснащенность помещений для самостоятельной работы обучающихся | Перечень лицензионного программного обеспечения |
|---|---|---|
| Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки) | Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное | |

| | | |
|---|---|--|
| | оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi) | |
| Помещение для самостоятельной работы обучающихся (ауд. _____) | <p>Мебель: учебная мебель</p> <p>Комплект специализированной мебели: компьютерные столы</p> <p>Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)</p> | |