

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**
(БГТУ им. В.Г. Шухова)



РАБОЧАЯ ПРОГРАММА
дисциплины

Основы программирования

направление подготовки:

09.03.01 «Информатика и вычислительная техника»

Направленность программы (профиль):

Вычислительные машины, комплексы, системы и сети

Квалификация

Бакалавр

Форма обучения

Очная

Институт энергетики, информационных технологий и управляющих систем

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

| Категория (группа) компетенций | Код и наименование компетенции | Код и наименование индикатора достижения компетенции | Наименование показателя оценивания результата обучения по дисциплине |
|----------------------------------|---|---|--|
| Разработка алгоритмов и программ | ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения | ОПК-8.1 Применяет методы алгоритмизации, языки и технологии программирования при решении задач профессиональной деятельности | Знания, умения. |
| | | ОПК-8.2 Программирует, отлаживает и тестирует прототипы программно-технических комплексов, пригодные для практического применения | Знания, умения, навыки. |

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

- 1. Компетенция ОПК-8.** Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

Данная компетенция формируется следующими дисциплинами.

| Стадия | Наименования дисциплины |
|--------|--|
| 1. | Основы программирования |
| 2. | Основы алгоритмизации |
| 3. | Учебная ознакомительная практика |
| 4. | Производственная технологическая (проектно-технологическая) практика |

3. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 10 зач. единиц, 360 часов.

Форма промежуточной аттестации зачёт, экзамен.

| Вид учебной работы | Всего часов | Семестр № 1 | Семестр № 2 | Семестр № 3 |
|---|-------------|-------------|-------------|-------------|
| Общая трудоемкость дисциплины, час | 360 | 144 | 144 | 72 |
| Контактная работа (аудиторные занятия), в т.ч.: | 145 | 73 | 55 | 17 |
| лекции | 51 | 34 | 17 | |
| лабораторные | 68 | 34 | 34 | |
| практические | 17 | | | 17 |
| групповые консультации в период теоретического обучения и промежуточной аттестации | 9 | 5 | 4 | |
| Самостоятельная работа студентов, включая индивидуальные и групповые консультации, в том числе: | 215 | 71 | 89 | 55 |
| Курсовой проект | | | | |
| Курсовая работа | 36 | | | 36 |
| Расчетно-графическое задание | | | | |
| Индивидуальное домашнее задание | | | | |
| Самостоятельная работа на подготовку к аудиторным занятиям (лекции, практические занятия, лабораторные занятия) | 107 | 35 | 53 | 19 |
| Экзамен | 72 | 36 | 36 | |

4.СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Наименование тем, их содержание и объем

Курс 1 Семестр 1

| № п/п | Наименование раздела (краткое содержание) | Объем на тематический раздел по видам учебной нагрузки, час | | | |
|---|--|---|----------------------|----------------------|------------------------|
| | | Лекции | Практические занятия | Лабораторные занятия | Самостоятельная работа |
| 1. Алгоритмы. Языки программирования. | | | | | |
| | Программы, как способ записи алгоритмов на языке программирования. История, тенденции развития и классификация языков программирования, их свойства | 2 | | | 1 |
| 2. Введение в язык C. | | | | | |
| | Структура программы. Подключение библиотек. Сборка программы на языке программирования C. Ввод вывод в языке программирования C. | 2 | | 2 | 3 |
| 3. Типы данных языка программирования C. | | | | | |
| | Понятие «типа данных», классификация типов данных, согласно стандарту C90. Тип данных bool, определенный в <stdbool.h>. | 2 | | | 1 |
| 4. Целочисленные типы данных. Вещественные типы данных. Указатели. | | | | | |
| | Представление целочисленных типов данных в памяти ЭВМ, вычисление диапазонов, принимаемых тем или иных типов данных. Представление вещественных типов данных в памяти ЭВМ. Проблемы вещественной арифметики. Операции, определенные над указателями. | 2 | | | 1 |
| 5. Линейные алгоритмы. Разветвляющиеся алгоритмы. | | | | | |
| | Реализация линейных / разветвляющихся алгоритмов в языке программирования C. Знакомство с простейшими операциями. | 2 | | 10 | 6 |
| 6. Арифметические и побитовые операции языка программирования C. | | | | | |
| | Понятие «операция». Типы операций. Арифметические операции: унарные плюс и минус, сложение и вычитание, умножение, деление и остаток от деления, приоритеты арифметических операций. Логические побитовые операции, операции сдвига. | 2 | | | 1 |
| 7. Логические операции и операции сравнения. Приведение типов. | | | | | |
| | Логические операции, операции сравнения и их приоритеты. Приоритеты операций. Приведение типов. | 2 | | | 1 |
| 8. Циклические алгоритмы: цикл <i>for</i> . | | | | | |
| | Описание и использование цикла <i>for</i> . Рассмотрение решаемых при помощи цикла задач. | 2 | | 4 | 2 |
| 9. Циклические алгоритмы: циклы <i>while</i> , <i>do-while</i> . Операторы <i>break</i> , <i>goto</i> , <i>continue</i> . | | | | | |
| | Описание и использование циклов <i>while</i> , <i>do-while</i> . Рассмотрение решаемых при помощи циклов задач. Рассмотрение аспектов использования операторов <i>break</i> , | 2 | | 2 | 2 |

| | | | | | |
|--|---|----|--|----|----|
| | <i>goto, continue.</i> | | | | |
| 10. Функции. | | | | | |
| | Понятие «Функция». Объявление и определение функций. Формальные и фактические параметры. Действия, производимые при вызове функций, возвращаемое значение функций. Передача аргументов. | 4 | | 2 | 2 |
| 11. Алгоритмы обработки одномерных массивов. | | | | | |
| | Алгоритмы обработки одномерных массивов, независимые от упорядоченности. | 2 | | 4 | 4 |
| 12. Алгоритмы обработки одномерных массивов. Алгоритмы поиска. | | | | | |
| | Алгоритмы, зависящие от упорядоченности. Алгоритмы поиска: линейный поиск, бинарный поиск, блочный поиск, тернарный поиск. Применение бинарного поиска в задачах оптимизации. | 2 | | 4 | 4 |
| 13. Рекурсивные функции. | | | | | |
| | Решение задач при помощи рекурсивных функций. Стек и рекурсия. | 2 | | 2 | 2 |
| 14. Алгоритмы сортировок. | | | | | |
| | Сортировки выбором, вставками, обменная сортировка, сортировка расческой. Сортировки подсчетом, цифровая, слиянием, кучей, быстрая. | 4 | | 2 | 2 |
| 15 Структуры. | | | | | |
| | Объявление структур. Передача структур в функции. Массивы структур. | 2 | | 2 | 3 |
| | ВСЕГО | 34 | | 34 | 35 |

Курс1 Семестр 2

| № п/п | Наименование раздела (краткое содержание) | Объем на тематический раздел по видам учебной нагрузки, час | | | |
|--|---|---|----------------------|----------------------|------------------------|
| | | Лекции | Практические занятия | Лабораторные занятия | Самостоятельная работа |
| 1. Структуры данных на одномерных массивах: множества. | | | | | |
| | Представление множеств в памяти ЭВМ. | 1 | | 2 | 3 |
| 2. Структуры данных на одномерных массивах: вектор. | | | | | |
| | Работа с динамической памятью в языке программирования С. Обзор механизма работы вектора (саморасширяющегося массива). | 1 | | 2 | 3 |
| 3. Алгоритмы обработки многомерных массивов. | | | | | |
| | Представления многомерных массивов в памяти ЭВМ (схемы размещения). Функции, используемые для обработки многомерных массивов. | | | 4 | 4 |
| 4. Строки. | | | | | |
| | Алгоритмы обработки строк стандартной библиотеки. Реализация пользовательских функций обработки. | 1 | | 4 | 5 |
| 5. Введение в С++: ссылки, изменения в управляющих конструкциях. | | | | | |
| | Ссылочный тип. Ссылки на константные и неконстантные данные. <i>Range-based for</i> . Инициализация локальных пе- | 2 | | 2 | 4 |

| | | | | | |
|-----|--|----|--|----|----|
| | ременных в <i>if</i> и <i>switch</i> . | | | | |
| 6. | Изменения в типах данных, функциях. | | | | |
| | Ключевое слово <i>auto</i> . Перегрузки функций, параметры функций по умолчанию, <i>lambda</i> -функции, введение в шаблоны. | 2 | | | 2 |
| 7. | Структуры в C++. | | | | |
| | Причины появления в языках программирования, ограничение доступа к полям структуры, константность функций-членов, конструкторы, инициализация полей структур и классов, деструкторы. | 2 | | 2 | 4 |
| 8. | Обзор контейнеров языка C++. | | | | |
| | Массивы, векторы, дек, односвязный список, двусвязный список, стек, очередь, приоритетная очередь, отображения, множества, мультимножества, неупорядоченные отображения и множества. | 2 | | 4 | 8 |
| 9. | Итераторы. Алгоритмы на паре итераторов. | | | | |
| | Подход без использования итераторов. Определение «итератор». Рассмотрение алгоритмов на паре итераторов. | 2 | | 2 | 4 |
| 10. | Работа с файловыми потоками. | | | | |
| | Потоки ввода/вывода. Перенаправление потоков. | 2 | | 4 | 6 |
| 11. | Динамическое программирование. | | | | |
| | Решение задач с подходов динамического программирования. Мемоизация и табуляция. | 2 | | 4 | 6 |
| 12. | Реализация структур данных с использованием ООП. | | | | |
| | Реализация вектора, односвязного списка, несбалансированных бинарных деревьев, с использованием объектно-ориентированного подхода в программировании. | | | 4 | 4 |
| | ВСЕГО | 17 | | 34 | 53 |

Курс 2 Семестр 3

| № п/п | Наименование раздела (краткое содержание) | Объем на тематический раздел по видам учебной нагрузки, час | | | |
|-------|---|---|----------------------|----------------------|------------------------|
| | | Лекции | Практические занятия | Лабораторные занятия | Самостоятельная работа |
| 1. | Вектора | | | | |
| | Реализация СД для хранения произвольного типа. | | 3 | | 3 |
| 2. | Длинная арифметика | | | | |
| | Способы представления чисел в памяти ЭВМ для решения задач длинной арифметики. | | 3 | | 3 |
| 3. | Линейная алгебра. | | | | |
| | Разработка модуля программных объектов для решения задач линейной алгебры (матрицы целочисленных дробей). | | 3 | | 3 |
| 4. | Аналитическая геометрия | | | | |

| | | | | | |
|--------------------------------|---|--|----|--|----|
| | Разработка модуля программных объектов для решения задач аналитической геометрии на плоскости | | 3 | | 4 |
| 5. Модульное программирование. | | | | | |
| | Системы сборки проектов (Make, CMake). | | 5 | | 6 |
| | ВСЕГО | | 17 | | 19 |

4.2. Содержание практических (семинарских) занятий

| № п/п | Наименование раздела дисциплины | Тема практического (семинарского) занятия | К-во часов | Самостоятельная работа на подготовку к аудиторным занятиям |
|-------------|---------------------------------|---|------------|--|
| семестр № 3 | | | | |
| 1 | Вектора | Разработка модуля программных объектов для работы с векторами | 3 | 3 |
| 2 | Длинная арифметика | Разработка модуля программных объектов для работы с длинными целыми | 3 | 3 |
| 3 | Линейная алгебра. | Разработка модуля программных объектов для решения задач линейной алгебры | 3 | 3 |
| 4 | Аналитическая геометрия | Разработка модуля программных объектов для решения задач линейной алгебры (матрицы целочисленных дробей). | 3 | 4 |
| 5 | Модульное программирование. | Системы сборки проектов (Make, CMake). | 5 | 6 |
| | ВСЕГО | | 17 | 19 |

4.3. Содержание лабораторных занятий

| № п/п | Наименование раздела дисциплины | Тема лабораторного занятия | К-во часов | Самостоятельная работа на подготовку к аудиторным занятиям |
|------------|---|--|------------|--|
| семестр №1 | | | | |
| 1 | Введение в язык С. | Ввод/вывод в языке программирования С. | 2 | 2 |
| 2 | Линейные алгоритмы. Разветвляющиеся алгоритмы. | Программирование алгоритмов разветвляющейся структуры. | 10 | 8 |
| 3 | Циклические алгоритмы: цикл <i>for</i> . Циклические алгоритмы: циклы <i>while</i> , <i>do-while</i> . Операторы <i>break</i> , | Программирование алгоритмов циклической структуры. | 6 | 4 |

| | | | | |
|------------|---|--|----|----|
| | <i>goto, continue</i> | | | |
| 4 | Функции. | Введение в функции. | 2 | 2 |
| 5 | Алгоритмы обработки одномерных массивов. | Алгоритмы обработки одномерных массивов. | 4 | 2 |
| 6 | Алгоритмы обработки одномерных массивов. | Бинарный поиска при решении задач оптимизации. | 4 | 2 |
| 7 | Рекурсивные функции. | Рекурсия. | 2 | 2 |
| 8 | Алгоритмы сортировок. | Анализ алгоритмов сортировки. | 2 | 2 |
| 9 | Структуры | Структуры. Функции для работы со структурами. | 2 | 2 |
| ИТОГО: | | | 34 | 26 |
| семестр №2 | | | | |
| 1 | Структуры данных на одномерных массивах: множества. | Множества | 2 | 3 |
| 2 | Структуры данных на одномерных массивах: вектор. | Реализация структуры данных «Вектор». | 2 | 2 |
| 3 | Алгоритмы обработки многомерных массивов. | Алгоритмы обработки многомерных массивов. | 4 | 4 |
| 4 | Строки. | Строки. | 4 | 3 |
| 5 | Введение в C++. | Введение в C++. | 2 | 2 |
| 6 | Структуры в C++. | Структуры в C++. | 2 | 2 |
| 7 | Обзор контейнеров языка C++. | Обзор контейнеров языка C++. | 4 | 4 |
| 8 | Итераторы. Алгоритмы на паре итераторов | Итераторы. Алгоритмы на паре итераторов | 2 | 2 |
| 9 | Работа с файловыми потоками. | Работа с файловыми потоками. | 4 | 4 |
| 10 | Динамическое программирование. | Динамическое программирование. | 4 | 4 |
| 11 | Реализация структур данных с использованием ООП. | Реализация структур данных с использованием ООП. | 4 | 3 |
| ИТОГО: | | | 34 | 33 |
| ВСЕГО: | | | 68 | 59 |

4.4. Содержание курсовой работы

Целью выполнения курсовой работы является углубление и закрепление студентами знаний основных приемов, методов и принципов работы при решении на ЭВМ задач с использованием языков высокого уровня, соблюдая принципы структурного программирования. Для выполнения работы достаточно знаний основных разделов дисциплин «Основы программирования», «Математический анализ», «Алгебра и геометрия», «Дискретная математика». Курсовая работа за-

ключается в разработке модулей программных объектов и приложений для решения задач заданного класса.

Курсовая работа оформляется в виде пояснительной записки (15-20 с.), в которой отражаются все этапы создания программного продукта.

В процессе выполнения курсовой работы осуществляется контактная работа обучающегося с преподавателем. Консультации проводятся в аудитории. Примеры тем курсовых работ.

1. Точное решение системы линейных уравнений произвольного порядка с рациональными коэффициентами.
2. Вычисления значения арифметического выражения, заданного строкой.
3. Решение головоломки «Судоку».
4. Решение головоломки «Какуро».
5. Обзор инструментов тестирования в языке программирования C.
6. Нахождение рациональных корней многочлена с рациональными коэффициентами.
7. Реализация игры «Морской бой».
8. Обзор алгоритмов обработки строк.
9. Реализация красно-черных и AVL-деревьев.
10. Разработка редактора кода в консоли.
11. Реализация алгоритма генерации уровней в игре.

Пояснительная записка оформляется в произвольной форме в зависимости от выбранной предметной области, однако обязательно включает в себя анализ предметной области, обзор имеющихся решений, этапы проектирования разработанного решения, результаты работы разработанного ПО.

4.5. Содержание расчетно-графического задания, индивидуальных домашних заданий

Не предусмотрено учебным планом.

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

5.1. Реализация компетенций

1. Компетенция ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения

| Наименование индикатора достижения компетенции | Используемые средства оценивания |
|---|--|
| ОПК-8.1 Применяет методы алгоритмизации, языки и технологии программирования при решении задач профессиональной деятельности | защита лабораторных работ, зачет, экзамен |
| ОПК-8.2 Программирует, отлаживает и тестирует прототипы программно-технических комплексов, пригодные для практического применения | защита лабораторных работ, дифференцированный зачет при защите курсовой работы |

5.2. Типовые контрольные задания для промежуточной аттестации

5.2.1. Перечень контрольных вопросов (типовых заданий) для экзамена

Курс 1 Семестр 1

| № п/п | Наименование раздела дисциплины | Содержание вопросов (типовых заданий) |
|-------|---|---|
| 1 | Алгоритмы. Языки программирования. | <ol style="list-style-type: none"> 1. Дайте определение термину 'алгоритм'. Перечислите свойства алгоритмов с их определениями. 2. Что является входными и выходными данными? 3. Перечислите способы описания алгоритмов. Приведите примеры каждого способа. От чего зависит выбор способа записи алгоритма? |
| 2 | Введение в язык C. | <ol style="list-style-type: none"> 1. Перечислите основные достоинства и недостатки языка C. 2. Препроцессор, компилятор и компоновщик. Этапы компиляции. 3. При использовании директивы препроцессора <i>#include</i> в чем заключается разница между <code>< ></code> и <code>" "</code>. 4. Перечислите принципы структурного программирования. 5. Каскадная модель. Её особенности и этапы |
| 3 | Типы данных языка программирования C. | <ol style="list-style-type: none"> 1. Что определяет тип данных? Перечислите базовые типы данных C. |
| 4 | Целочисленные типы данных. Вещественные типы данных. Указатели. | <ol style="list-style-type: none"> 1. Целочисленный тип <i>char</i>. Вывод диапазона значений <i>char / unsigned char</i>. 2. Целочисленный тип <i>int</i>. Вывод диапазона значений. 3. Вещественный тип <i>float</i> и <i>double</i>. В чём разница между ними? 4. В каких системах счисления могут быть представлены целочисленные литералы? Приведите примеры. 5. Приведите примеры <i>escape</i>-последовательностей. 6. В каком случае инициализация переменной при ее объявлении будет излишней? 7. Массивы в C. Объявление. Инициализация. 8. Указатели в C. Операции взятия адреса и косвенного доступа. |

| | | |
|----|--|---|
| | | <p>Макрос <i>NULL</i>.</p> <p>9. В чем заключается разница между литералами 'x' и 'x'?"</p> <p>10. В чем заключается разница между объявлением констант при помощи макроса <i>#define</i> и при помощи ключевого слова <i>const</i>?</p> |
| 5 | Линейные алгоритмы. Разветвляющиеся алгоритмы. | <p>1. Что такое линейный алгоритм? Приведите примеры.</p> <p>2. Функция ввода <i>scanf</i> и её параметры.</p> <p>3. Функция вывода <i>printf</i> и её параметры.</p> <p>4. Подключение библиотеки в C. Какую библиотеку необходимо подключить для использования математических функций?</p> <p>5. Функции для генерации случайных чисел. Для чего используются функции <i>srand</i> и <i>rand</i>?</p> <p>6. Правило выполнения деления в C.</p> <p>7. Какие алгоритмы называются разветвляющимися?</p> <p>8. Какие имеются требования к тестовым данным для тестирования программ с ветвлением?</p> <p>9. Что представляет собой логическое выражение?</p> <p>10. Как организовать бинарное ветвление?</p> <p>11. Как организовать множественное ветвление?</p> <p>12. В каком случае лучше использовать тернарный оператор?</p> <p>13. К какому <i>if</i> относится <i>else</i>?</p> |
| 6 | Арифметические и побитовые операции языка программирования C. | <p>1. Дайте определение терминам 'операция' и 'операнд'. Перечислите виды операций в зависимости от количества операндов. Приведите примеры.</p> <p>2. Арифметические операции и их приоритеты</p> <p>3. Какие существуют способы в C увеличить или уменьшить значение переменной на единицу?</p> <p>4. В чем заключается разница между постфиксной и префиксной записи инкремента и декремента?</p> <p>5. Какая проблема может возникнуть при умножении чисел в C?</p> <p>6. Каким образом работает деление в языке программирования C.</p> <p>7. Побитовые операции и их приоритеты.</p> |
| 7 | Логические операции и операции сравнения. Приведение типов. | <p>1. Перечислите какие бывают приведения типов в C. В каких случаях они выполняются?</p> <p>2. Перечислите последствия при приведении типов. Приведите примеры.</p> <p>3. Логические операции и их приоритеты. Опишите как работает 'ленивая' схема вычислений.</p> |
| 8 | Циклические алгоритмы: цикл <i>for</i> . | <p>1. Какие алгоритмы называются циклическими?</p> <p>2. Секции цикла <i>for</i>. Какие из них могут быть опущены?</p> |
| 9 | Циклические алгоритмы: циклы <i>while</i> , <i>do-while</i> . Операторы <i>break</i> , <i>goto</i> , <i>continue</i> . | <p>1. В чем заключается разница между циклом <i>while</i> и циклом <i>do-while</i>?</p> <p>2. В каком случае цикл <i>while</i> может стать бесконечным?</p> <p>3. Сколько раз обязательно выполнится тело цикла <i>do-while</i>?</p> <p>4. Инструкции <i>break</i> и <i>goto</i>. Когда имеет смысл их применять?</p> <p>5. Как заменить цикл <i>for</i> 'эквивалентным' через цикл <i>while</i>.</p> <p>6. Инструкция <i>continue</i>. В чём разница применения <i>continue</i> для цикла <i>for</i> и <i>while</i>?</p> |
| 10 | Функции. | <p>1. В каких случаях целесообразно использовать функции?</p> <p>2. Принципы <i>DRY</i> и <i>WET</i>.</p> <p>3. В чём разница между объявлением и определением функции?</p> <p>4. Какие параметры называются фактическими, а какие формальными?</p> <p>5. Может ли в результате вызова функции измениться значение фактического параметра функции?</p> |

| | | |
|----|---|--|
| | | <p>6. Действия, производимые при вызове функций.</p> <p>7. Для чего используется квалификатор <i>const</i> для формальных параметров функции?</p> <p>8. В чем заключается побочный эффект функции?</p> <p>9. Как изменить аргументы, передаваемые в функцию?</p> |
| 11 | Алгоритмы обработки одномерных массивов. | <p>1. Как описываются массивы в С?</p> <p>2. Как осуществляется ввод и вывод одномерных массивов?</p> <p>3. Какими способами может быть осуществлен поиск элемента в упорядоченном и неупорядоченном массиве?</p> <p>4. Опишите алгоритм однопроходного алгоритма удаления из последовательности членов, удовлетворяющих заданному условию.</p> <p>5. Как передавать в функцию адрес некоторой отличной от начальной части массива? Приведите пример.</p> <p>6. В чем заключается принцип работы стека?</p> <p>7. В чем заключается принцип работы кучи?</p> <p>8. В каких случаях используются динамические массивы?</p> <p>9. Какие функции отвечают в С за работу с динамической памятью? Как работают данные функции?</p> <p>10. Передача функции в функцию. Для чего необходим данный прием? Приведите примеры.</p> <p>11. Опишите последовательность решения задачи на одномерные массивы с использованием принципа пошаговой детализации.</p> |
| 12 | Алгоритмы обработки одномерных массивов. Алгоритмы поиска. | <p>1. Опишите алгоритм линейного поиска.</p> <p>2. Опишите алгоритм блочного поиска. Каков оптимальный размер блока при блочном поиске? Приведите доказательство.</p> <p>3. Бинарный поиск в двух вариациях. Различия в реализации.</p> |
| 13 | Рекурсивные функции. | <p>1. Опишите действия, производимые на стеке вызовов при работе с рекурсивными функциями.</p> <p>2. Опишите виды рекурсии, приведите примеры.</p> <p>3. Подходы к решению задач с использованием рекурсии.</p> |
| 14 | Алгоритмы сортировок. | <p>1. Реализация сортировки пузырьком.</p> <p>2. Реализация сортировки выбором.</p> <p>3. Реализация сортировки вставками.</p> <p>4. Реализация сортировки расческой.</p> <p>5. Реализация сортировки Шелла.</p> <p>6. Реализация пирамидальной сортировки.</p> <p>7. Реализация быстрой сортировки.</p> <p>8. Реализация цифровой сортировки.</p> |
| 15 | Структуры. | <p>1. Объявление структуры.</p> <p>2. Инициализация структуры.</p> <p>3. Создание массива структур.</p> <p>4. Создание динамического массива структур.</p> <p>5. Передача структур в функцию.</p> <p>6. Сортировка функций. Использование компаратора при сортировке.</p> <p>7. Битовые поля.</p> |

Курс 1 Семестр 2

| | | |
|----|---|---|
| 16 | Структуры данных на одномерных массивах: множества. | <p>1. Способ представления множества на битах целого числа.</p> <p>2. Способ представления множества на неупорядоченном массиве.</p> <p>3. Способ представления множества на упорядоченном массиве.</p> |
| 17 | Структуры данных на | <p>1. Политики расширения динамического массива при его запол-</p> |

| | | |
|----|---|---|
| | одномерных массивах: вектор. | нении. 2. Реализация функций для работы с векторами. |
| 18 | Алгоритмы обработки многомерных массивов. | 1. Способы размещения матрицы в динамической памяти. 2. Схемы размещения. |
| 19 | Строки. | 1. Стандартная библиотека <string.h>. Функции. Механизмы работы. 2. Ввод и вывод строк в языке программирования C. |
| 20 | Введение в C++: ссылки, изменения в управляющих конструкциях. | 1. Виды ссылок: ссылки на неконстантные данные, ссылки на константные данные. 2. <i>Range-based for</i> . 3. Вариации <i>range-based for</i> . 4. Ключевое слово <i>auto</i> и <i>decltype</i> . |
| 21 | Изменения в типах данных, функциях. | 1. <i>lambda</i> -функции. 2. Шаблоны функций. 3. Перегрузки функций. 4. Параметры функции по умолчанию. |
| 22 | Структуры в C++. | 1. Причины появления в языках программирования 2. Ограничение доступа к полям структуры. 3. Константность функций-членов. 4. Конструкторы. 5. Инициализация полей структур. 6. Деструкторы. |
| 23 | Обзор контейнеров языка C++. | 1. Массивы. 2. Векторы. 3. Дек. 4. Односвязный список. 5. Двусвязный список. 6. Стек. 7. Очередь. 8. Приоритетная очередь. 9. Отображения. 10. Множества. 11. Мультимножества. 12. Неупорядоченные отображения и множества. |
| 24 | Итераторы. Алгоритмы на паре итераторов. | 1. Итераторы. Определение. Предпосылки к созданию. 2. Алгоритмы на паре итераторов. |
| 25 | Работа с файловыми потоками. | 1. Понятия «файл», «поток». 2. Функции для работы с файлами. |
| 26 | Динамическое программирование. | 1. Динамическое программирование. Этапы решения задач с использованием динамического программирования. 2. Задачи одномерного динамического программирования. 3. Задачи многомерного динамического программирования. 4. Техники кэширования при решении задач на динамическое программирование: мемоизация и табуляция. |
| 27 | Реализация структур данных с использованием ООП. | 1. Структуры и классы. 2. Конструктор без параметров / с параметрами. 3. Деструктор. 4. Конструкторы копирования и перемещения. 5. Перегрузка операций. |

5.2.2. Перечень контрольных материалов для защиты курсовой работы

1. Формулировка принципов структурного программирования
2. Какие программные объекты можно описывать в заголовочных файлах, а какие не рекомендуется?
3. По каким критериям выполняется выделение модулей при разработке программы?
4. Какими соображениями руководствуются при создании наборов тестовых данных?

5.3. Типовые контрольные задания (материалы) для текущего контроля в семестре

Текущий контроль осуществляется в течение семестра в форме защиты лабораторных работ.

В методических указаниях к выполнению лабораторных работ по дисциплине представлен перечень лабораторных работ, обозначены цель и задачи, представлены варианты заданий.

Защита лабораторной работы проводится в форме устного опроса студента и направлена на проверку степени усвоения материала и понимания теоретических сведений, используемых в процессе выполнения работы. Для защиты необходимо представить в печатной (рукописной) форме отчет по лабораторной работе, выполненный самостоятельно и в соответствии со всеми требованиями, приведёнными в методических указаниях к выполнению лабораторных работ. Примерный перечень контрольных вопросов для защиты лабораторных работ приведен в таблице:

| Тематика лабораторной работы | Контрольные вопросы |
|---|--|
| Лабораторная работа №1 Ввод/вывод в языке программирования C. | <ol style="list-style-type: none">1. Функция ввода <i>printf</i> и её параметры.2. Функция вывода <i>scanf</i> и её параметры.3. Буферизация вывода. Очистка потока вывода.4. Спецификаторы преобразования для функций ввода и вывода.5. Ввод и вывод строк. Строковые литералы.6. Типы данных языка C:<ul style="list-style-type: none">• Диапазоны значений базовых типов.• Количество значений, представимых переменными базового типа.• Литералы базовых типов. |
| Лабораторная работа №2а. Программирование алгоритмов разветвляющейся структуры. | <ol style="list-style-type: none">1. Способы организации ветвления в языке программирования C.2. Условный оператор <i>if</i>. |
| Лабораторная работа №2б. Программирование алгоритмов разветвляющейся структуры. | <ol style="list-style-type: none">3. Определение «логическое выражение».4. Оператор множественного ветвления <i>switch</i>. |

| | |
|---|---|
| Лабораторная работа №2с. Программирование алгоритмов разветвляющейся структуры. | |
| Лабораторная работа №3а Разработка алгоритмов циклической структуры. Введение в тестирование. | <ol style="list-style-type: none"> 1. Циклы языка программирования C. 2. Описание и механизм работы цикла <i>for</i>. 3. Описание и механизм работы цикла <i>while</i>. 4. Описание и механизм работы цикла <i>do</i>. 5. Оператор <i>break</i>. 6. Оператор <i>goto</i>. 7. Оператор <i>continue</i>. 8. В каких случаях оправдано использование <i>break</i>, <i>goto</i>, <i>continue</i>. |
| Лабораторная работа №3б Разработка алгоритмов циклической структуры. | <ol style="list-style-type: none"> 6. Оператор <i>goto</i>. 7. Оператор <i>continue</i>. 8. В каких случаях оправдано использование <i>break</i>, <i>goto</i>, <i>continue</i>. |
| Лабораторная работа №4 Введение в функции. | <ol style="list-style-type: none"> 1. В каких случаях целесообразно использовать функции? 2. Принципы <i>DRY</i> и <i>WET</i>. 3. В чём разница между объявлением и определением функции? 4. Какие параметры называются фактическими, а какие формальными? 5. Может ли в результате вызова функции измениться значение фактического параметра функции? 6. Действия, производимые при вызове функций. 7. Для чего используется квалификатор <i>const</i> для формальных параметров функции? 8. В чем заключается побочный эффект функции? 9. Как изменить аргументы, передаваемые в функцию? |
| Лабораторная работа №5 Алгоритмы обработки одномерных массивов. | <ol style="list-style-type: none"> 1. Опишите алгоритм ввода / вывода массива. 2. Поиск позиции элемента, удовлетворяющего условию. 3. Поиск количества элементов, удовлетворяющих условию. 4. Поиск максимального количества подряд идущих элементов, удовлетворяющих условию. 5. Однопроходный алгоритм удаления. 6. Вставка элемента с сохранением порядка остальных элементов. 7. Вставка элемента без сохранения порядка остальных элементов. 8. Удаление элемента с сохранением порядка остальных элементов. 9. Удаление элемента без сохранения порядка остальных элементов. 10. Передача функций, как параметров. |
| Лабораторная работа №6 Бинарный поиска при решении задач оптимизации. | <ol style="list-style-type: none"> 1. Бинарный поиск в двух вариациях. Различия в реализации. 2. Определение сложности алгоритмов бинарного поиска. |
| Лабораторная работа №7 Рекурсия. | <ol style="list-style-type: none"> 1. Опишите действия, производимые на стеке вызовов при работе с рекурсивными функциями. 2. Опишите виды рекурсии, приведите примеры. 3. Подходы к решению задач с использованием рекурсии. |
| Лабораторная работа №8 Алгоритмы сортировок. | <ol style="list-style-type: none"> 1. Сортировка выбором. 2. Сортировка вставками. 3. Обменная сортировка (пузырьковая). 4. Сортировка расческой. 5. Сортировка слиянием. 6. Быстрая сортировка. 7. Поразрядная сортировка. 8. Сортировка подсчётом. |

| | |
|---|--|
| | 9. Цифровая сортировка. |
| Лабораторная работа №9 Структуры. Функции для работы со структурами. | 1. Объявление структуры. 2. Инициализация структуры. 3. Создание массива структур. 4. Создание динамического массива структур. 5. Передача структур в функцию. 6. Сортировка функций. Использование компаратора при сортировке. 7. Битовые поля. |
| Лабораторная работа №10 Множества. | 1. Способ представления множества на битах целого числа. 2. Способ представления множества на неупорядоченном массиве. 3. Способ представления множества на упорядоченном массиве. |
| Лабораторная работа №11 Реализация структуры данных «Вектор». | 1. Политики расширения динамического массива при его заполнении. 2. Функции для работы с векторами. |
| Лабораторная работа №12 Работа с многомерными массивами. | 1. Способы размещения матрицы в динамической памяти. 2. Схемы размещения. |
| Лабораторная работа №13 Алгоритмы обработки строк. | 1. Стандартная библиотека <string.h>. Функции. Механизмы работы. 2. Ввод и вывод строк в языке программирования C. |
| Лабораторная работа №14 Введение в C++: ссылки, изменения в управляющих конструкциях. | 1. Виды ссылок: ссылки на неконстантные данные, ссылки на константные данные. 2. <i>Range-based for</i> . 3. Вариации <i>range-based for</i> . 4. Ключевое слово <i>auto</i> и <i>decltype</i> . |
| Лабораторная работа №15 Структуры в C++. | 1. Причины появления в языках программирования 2. Ограничение доступа к полям структуры. 3. Константность функций-членов. 4. Конструкторы. 5. Инициализация полей структур. 6. Деструкторы. |
| Лабораторная работа №16 Обзор контейнеров языка C++. | 1. Массивы. 2. Векторы. 3. Дек. 4. Односвязный список. 5. Двусвязный список. 6. Стек. 7. Очередь. 8. Приоритетная очередь. 9. Отображения. 10. Множества. 11. Мультимножества. 12. Неупорядоченные отображения и множества. |
| Лабораторная работа №17 Итераторы. Алгоритмы на паре итераторов. | 1. Итераторы. Определение. Предпосылки к созданию. 2. Алгоритмы на паре итераторов. |
| Лабораторная работа | 1. Понятия «файл», «поток». |

| | |
|--|---|
| №18 Работа с файловыми потоками. | 2. Функции для работы с файлами. |
| Лабораторная работа №19 Динамические переменные | 1. Динамическое программирование. Этапы решения задач с использованием динамического программирования. 2. Задачи одномерного динамического программирования. 3. Задачи многомерного динамического программирования. 4. Техники кэширования при решении задач на динамическое программирование: мемоизация и табуляция. |
| Лабораторная работа №20 Реализация структур данных с использованием ООП. | 1. Структуры и классы. 2. Конструктор без параметров / с параметрами. 3. Деструктор. 4. Конструкторы копирования и перемещения. 5. Перегрузка операций. |

Критерии оценки лабораторной работы: лабораторная работа считается защищенной, если студент выполнил задание к работе полностью и во время устного опроса по работе правильно ответил на заданные преподавателем дополнительные вопросы.

5.4. Описание критериев оценивания компетенций и шкалы оценивания

При промежуточной аттестации в форме экзамена и дифференцированного зачёта при защите курсовой работы используется следующая шкала оценивания: 2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично.

Критериями оценивания достижений показателей являются:

| Наименование показателя оценивания результата обучения по дисциплине | Критерий оценивания |
|--|--|
| Знания | Знание синтаксиса и семантики языка программирования |
| | Объем освоенного материала |
| | Полнота ответов на вопросы |
| | Четкость изложения и интерпретации знаний |
| Умения | Умение разрабатывать алгоритмы и программы с применением принципов структурного программирования |
| | Умение проверять решение и анализировать результаты |
| Навыки | Владение навыками отладки и тестирования программ |
| | Самостоятельность выполнения отладки и тестирования программ |

Оценка преподавателем выставляется интегрально с учётом всех показателей и критериев оценивания.

Оценка сформированности компетенций по показателю Знания.

| Критерий | Уровень освоения и оценка | | | |
|--|--|--|--|--|
| | 2 | 3 | 4 | 5 |
| Знание синтаксиса и семантики языка программирования | Не знает синтаксиса и семантики языка программирования | Знает основные синтаксические конструкции языка программирования | Знает синтаксис языка программирования | Знает синтаксис и семантику языка программирования |
| Объем освоенного материала | Не знает значительной части материала | Знает только основной материал | Знает материал дисциплины в до- | Обладает твердым и полным знанием ма- |

| | | | | |
|---|--|---|---|---|
| | ла дисциплины | дисциплины, не усвоил его деталей | статочном объеме | териала дисциплины, владеет дополнительными знаниями |
| Полнота ответов на вопросы | Не дает ответы на большинство вопросов | Дает неполные ответы на все вопросы | Дает ответы на вопросы, но не все - полные | Дает полные, развернутые ответы на поставленные вопросы |
| Четкость изложения и интерпретации знаний | Излагает знания без логической последовательности | Излагает знания с нарушениями в логической последовательности | Излагает знания без нарушений в логической последовательности | Излагает знания в логической последовательности, самостоятельно их интерпретируя и анализируя |
| | Не иллюстрирует изложение поясняющими схемами, рисунками и примерами | Выполняет поясняющие схемы и рисунки небрежно и с ошибками | Выполняет поясняющие рисунки и схемы корректно и понятно | Выполняет поясняющие рисунки и схемы точно и аккуратно, раскрывая полноту усвоенных знаний |
| | Неверно излагает и интерпретирует знания | Допускает неточности в изложении и интерпретации знаний | Грамотно и по существу излагает знания | Грамотно и точно излагает знания, делает самостоятельные выводы |

Оценка сформированности компетенций по показателю Умения.

| Критерий | Уровень освоения и оценка | | | |
|--|--|---|---|---|
| | 2 | 3 | 4 | 5 |
| Умение разрабатывать алгоритмы и программы с применением принципов структурного программирования | Не умеет разрабатывать алгоритмы и программы с применением принципов структурного программирования | Допускает неточности в разработке алгоритмов и программ с применением принципов структурного программирования | Умеет разрабатывать алгоритмы и программы с применением принципов структурного программирования | Безошибочно разрабатывает алгоритмы и программы с применением принципов структурного программирования |
| Умение проверять решение и анализировать результаты | Не умеет проверять решение и анализировать результаты | Умеет проверять решение некоторых задач | Умеет проверять решение некоторых задач и анализировать результаты | Умеет проверять решение и анализировать результаты |

Оценка сформированности компетенций по показателю Навыки.

| Критерий | Уровень освоения и оценка | | | |
|--|---|---|--|--|
| | 2 | 3 | 4 | 5 |
| Владение навыками отладки и тестирования программ | Не владеет навыками отладки и тестирования программ | Не достаточно хорошо владеет навыками отладки и тестирования программ | Владеет навыками отладки и тестирования программ | Профессионально владеет навыками отладки и тестирования программ |
| Самостоятельность выполнения отладки и тестирования программ | Не может самостоятельно выполнять исследования объектов профессиональной деятельности | Выполняет исследования объектов профессиональной деятельности с посторонней помощью | При выполнении исследования объектов профессиональной деятельности иногда требуется посторонняя помощь | Самостоятельно выполняет исследования объектов профессиональной деятельности |

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

6.1. Материально-техническое обеспечение

| № | Наименование специальных помещений и помещений для самостоятельной работы | Оснащенность специальных помещений и помещений для самостоятельной работы |
|----|---|--|
| 1. | Учебная аудитория для проведения лекционных занятий | Специализированная мебель. Мультимедийная установка, экран, доски |
| 2. | Учебная аудитория для проведения лабораторных занятий | Специализированная мебель. Компьютеры на базе процессоров Intel или AMD. |
| 3. | Читальный зал библиотеки для самостоятельной работы | Специализированная мебель. Компьютерная техника, подключенная к сети интернет и имеющая доступ в электронно-образовательную среду |

6.2. Лицензионное и свободно распространяемое программное обеспечение

| № | Перечень лицензионного программного обеспечения. | Реквизиты подтверждающего документа |
|----|--|--|
| 1. | Microsoft Windows 10 Корпоративная | (Соглашение Microsoft Open Value Subscription V6328633 Соглашение действительно с 02.10.2017 по 23.10.2023). Договор поставки ПО 0326100004117000038-0003147-01 от 06.10.2017. |
| 2. | Microsoft Office Professional Plus 2016 | (Соглашение Microsoft Open Value Subscription V6328633 Соглашение действительно с 02.10.2017 по 23.10.2023). Договор поставки ПО 0326100004117000038-0003147-01 от 06.10.2017. |
| 3. | Среды программирования Microsoft Visual Studio Code, Dev C++ или CodeBlocks. | Свободно распространяемое ПО согласно условиям лицензионного соглашения |

6.3. Перечень учебных изданий и учебно-методических материалов

1. Керниган, Б. Язык программирования Си: Пер. с англ./ Б. Керниган, Д. Ритчи.– 2-е изд., испр.– М.: Вильямс, 2017 .– 288 с.
2. Брусенцева В. С. Язык программирования Си. Учебное пособие по дисциплине "Основы программирования" для студентов направлений подготовки 09.01.01 – Информатика и вычислительная техника и 09.01.04 – Программная инженерия, Белгород: Изд-во БГТУ им. В.Г. Шухова, 2018. – 64 с
3. Брусенцева В. С. Методические указания к выполнению курсовых работ по дисциплине «Основы программирования» для студентов, направлений бакалавриата 09.03.01 – Информатика и вычислительная техника, 09.03.04 – Программная инженерия. Белгород: Изд-во БГТУ им. В.Г. Шухова, 2014. – 11 с
4. Плаксин М.А. Тестирование и отладка программ для профессионалов будущих и настоящих [Электронный ресурс]/ Плаксин М.А.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2013.— 167 с.

<http://www.iprbookshop.ru/20704.html>

5. Лоспинозо Д. С++ для профи. — СПб.: Питер, 2021. — 816 с.: ил. ISBN 978-5-4461-1730-7.
6. Липпман С., Лажойе Ж. Язык программирования С++. Полное руководство / Пер. с англ. — 2-е изд. — Саратов: Профобразование, 2019. — 1104 с.: ил. ISBN 978-5-4488-0136-5.
7. Мартин Р. Чистый код: создание, анализ, рефакторинг. — СПб.: Питер, 2019. — 464с.:ил. ISBN 978-5-4461-0960-9.
8. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр. : ил. ISBN 978-5-7502-0064-1.

6.4. Перечень интернет-ресурсов, профессиональных баз данных, информационно-справочных систем

1. Электронная библиотека (на базе ЭБС «БиблиоТех») — Режим доступа: <http://ntb.bstu.ru>.
2. Электронно-библиотечная система IPRbooks — Режим доступа: <http://www.iprbookshop.ru>.
3. Электронно-библиотечная система «Университетская библиотека ONLINE» — Режим доступа: <http://www.biblioclub.ru/>.