МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра компьютерной безопасности и математических методов обработки информации

УТВЕРЖДАЮ
Декан математического факультета

Нестеров П.Н.
20 июня 2023 г.

Рабочая программа дисциплины

Основы программирования

Направление подготовки (специальности)
02.03.01 Математика и компьютерные науки

Направленность (профиль)
«Программирование, алгоритмы и анализ данных»

Форма обучения очная

Программа рассмотрена
на заседании кафедры
от 14 апреля 2023 г., протокол № 8

Программа одобрена НМК
математического факультета
протокол № 9 от 3 мая 2023 г.

Ярославль

1. Цели освоения дисциплины

Дисциплина "Основы программирования" нацелена на подготовку студентов к деятельности, связанной с разработкой и использованием математических алгоритмов для решения профессиональных задач, и, является одним из основных предметов, способствующих развитию алгоритмической культуры и компьютерной грамотности студентов.

Курс способствует формированию фундаментальной базы, необходимой для успешного освоения как общепрофессиональных, так и специальных дисциплин, изучение которых связано с применением современных вычислительных систем для различных предметных областей.

Целью изучения данной дисциплины является ознакомление студентов с понятием алгоритма, способами и средствами их представления, их анализом и оценкой трудоемкости, различными парадигмами программирования, классификацией и эволюцией языков программирования, и современными тенденциями их развития, знакомство с современными вычислительными системами, изучение математических алгоритмов, применение их на практике, а также детальное изучение одного из языков программирования высокого уровня (язык C++).

2. Место дисциплины в структуре образовательной программы

Дисциплина «Основы программирования» относится к обязательной части образовательной программы и входит в модуль «Программирование I».

Знания и навыки, полученные при изучении дисциплины «Основы программирования", используются учащимися при изучении последующих общепрофессиональных и специальных дисциплин компьютерного цикла, в частности дисциплин «Методы трансляции», «Пакеты математических программ и математическое моделирование», «Компьютерные технологии в математических дисциплинах», а также «Алгоритмы на графах» и «Быстрые алгоритмы».

Между вторым и третьим семестрами студенты проходят научно-исследовательскую практику, в рамках которой изучают объектно-ориентированное программирование.

Знания и практические навыки, полученные в результате освоения дисциплины, используются студентами при разработке курсовых и дипломных работ, в научно-исследовательской работе.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	**Индикатор достижения компетенции (код и формулировка)**	**Перечень планируемых результатов обучения**
Общепрофессиональные компетенции		
ОПК-4 Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	**ИД-ОПК-4.1** Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	***Знать:*** − общие принципы построения и использования современных языков программирования высокого уровня; − базовые структуры данных; – принципы процедурного и объектно-ориентированного программирования; – принципы разработки программ и отдельных программных модулей и комплексов, принципы оценки их качества; – основы программирования на языке С++. ***Владеть:*** – навыками алгоритмического мышления. – методиками современной технологии программирования.
	ИД-ОПК-4.2 Умеет использовать этот математический аппарат в профессиональной деятельности	***Знать:*** – средства описания алгоритмов; – различные парадигмы программирования. ***Уметь:*** – формализовать поставленную задачу; – проводить оценку сложности алгоритмов; – пользоваться инструментальными средствами для разработки прикладных приложений. - тестировать и отлаживать программы; ***Владеть навыками:*** – разработки алгоритмов решения типовых профессиональных задач;

	ИД-ОПК-4.3 Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	***Знать:*** – современные технологии программирования ***Уметь:*** – составлять программы на языке программирования высокого уровня для решения вычислительных задач и задач обработки информации; – работать с современными системами программирования, включая объектно-ориентированные; – оформлять программную документацию – оценивать качество готового программного продукта ***Владеть навыками:*** – использования инструментальных средств для создания, отладки и тестирования программ и отдельных модулей, библиотек; – работы с научно-технической литературой и технической документацией по программному обеспечению.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 9 зачетных единиц, 324 акад. часа.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации *(по семестрам)* Формы ЭО и ДОТ *(при наличии)*
			Контактная работа					самостоятельная работа	
			лекции	практические	лабораторные	консультации	аттестационные испытания		
1.	Тема 1. Введение. Алгоритмы и их представление.	1	1					1	
2.	Тема 2. Языки программирования. Классификация, особенности, история развития.	1	1					1	
3.	Тема 3. Введение в язык С++. Структура программы, основные типы данных.	1	1		0,5			1,5	Лабораторная работа 1
4.	Тема 4. Простейшие средства ввода и вывода.	1	1		0,5			1,5	Лабораторная работа 1
5.	Тема 5. Операции, выражения и операторы в языке С++.	1	1		0,5			2	Лабораторная работа 1
6.	Тема 6. Операторы ветвления и выбора в языке С++.	1	1		0,5			3	Контест № 1 Условия и циклы Лабораторная работа 1
7.	Тема 7. Программирование циклических процессов.	1	2			2		4	Контрольная работа № 1 Контест № 2 Циклы и последовательности
8.	Тема 8. Указатели. Работа с динамической памятью. Арифметика указателей. Ссылки.	1	1					2	
9.	Тема 9. Одномерные массивы: статические, динамические. Контейнер vector(STL)	1	4		2			4	Контест № 3 Массивы. Линейный поиск Лабораторная работа 2
10.	Тема 10.	1	2					4	Контест № 4 Бинарный поиск

	Линейный и бинарный поиск. Бинарный поиск по ответу.								
11.	Тема 11. Алгоритмы сортировки: подсчетом, поразрядные, квадратичные, быстрые.	1	4		2	1		6	Контест № 5 Массивы. Сортировка Лабораторная работа 2
12.	Тема 12. Функции, способы передачи параметров. Указатель на функцию. Шаблоны. Аргументы командной строки.	1	2					2	Контест № 6 Массивы. Разное
13.	Тема 13. Рекурсия. Типы рекурсии. Оптимизация.	1	2		2			4	Лабораторная работа 3
14.	Тема 14. Символьная информация. Способы представления строк. Функции обработки строк.	1	3					4	Контест № 7 Разбор строк
15.	Тема 15. Средства потокового ввода и вывода. Файлы.	1	2		4	1		6	Контест № 8 Массивы строк Лабораторная работа 4
16.	Тема 16. Многомерные массивы. Способы представления.	1	4		4	2		8	Контрольная работа № 2 Контест № 9 Двумерные массивы Контест № 10 Двумерные массивы Лабораторная работа 5
	Итого за 1 семестр 108 часов		***32***		***16***	***6***		***54***	
17.	Тема 17. Использование структур для определения пользовательских типов данных.	2	2		2	1		1	Контест № 11 Геометрические задачи Лабораторная работа 6
18.	Тема 18. Использование библиотеки OpenGL.	2	3		2	1		2	Лабораторная работа 7
19.	Тема 19. Жадные алгоритмы.	2	2		1			2	Контест № 12 Жадные алгоритмы
20.	Тема 20. Волновой алгоритм.	2	2		1			1	
21.	Тема 21. Динамическое программирование. Одномерная динамика. НОП, НВП.	2	2		2			1	Контест № 13 Динамическое программирование Лабораторная работа 8
22.	Тема 22. Двумерная динамика. Расстояние Левенштейна	2	2		2			1	Контест № 13 Динамическое программирование Лабораторная работа 8

23.	Тема 23. Динамика по профилю.	2	2					1	Контест № 13 Динамическое программирование
24.	Тема 24. Однонаправленные, двунаправленные, циклические списки. Реализации на базе массива и указателей.	2	2		1	1		1	Контест № 14 Стеки, очереди, деревья Контрольная работа № 3 Лабораторная работа 9
25.	Тема 25. Другие последовательные динамические структуры данных. Стеки, деки, очереди.	2	2		1	1		1	Контест № 14 Стеки, очереди, деревья Лабораторная работа 9 Контрольная работа № 4
26.	Тема 26. Куча (пирамида). Пирамидальная сортировка.	2	2		1			1	Лабораторная работа 10
27.	Тема 27. Бинарные деревья. Деревья поиска. Сбалансированный деревья.	2	4		2			1	Контест № 14 Стеки, очереди, деревья Лабораторная работа 10 Контрольная работа № 4
28.	Тема 28. Деревья общего вида.	2	2		1	1		1	Лабораторная работа 10
29.	Тема 29. Основы объектно-ориентированного подхода.	2	4			1		3	Контест № 15 ООП Контрольная работа № 4
30.	Тема 30. Статические и динамические библиотеки.	2	1					1	
						2	***0,5***	***33,5***	***Экзамен***
	Итого за 2 семестр 108 часов		***32***		***16***	***8***	***0,5***	***51,5***	
31.	Тема 31. Арифметические алгоритмы. Проверка на простоту. Решено Эратосфена. Функция Эйлера.	3	2		1			1	Лабораторная работа 11
32.	Тема 32. НОД, алгоритм Евклида, расширенный алгоритм Евклида. Их приложения.	3	2		1			1	Лабораторная работа 11
33.	Тема 33. Системы счисления. Алгоритмы перевода целых и дробных чисел. Специальные возможности языка.	3	2		1			1	Лабораторная работа 11
34.	Тема 34. Битовые операции. Перебор всех подмасок.	3	2		1	2		2	Лабораторная работа 11

35.	Тема 35. Комбинаторные алгоритмы. Перебор подмножеств. Перебор размещений. Рекурсивный и итеративный алгоритмы.	3	4		1			1	Лабораторная работа 12
36.	Тема 36. Перебор сочетаний с повторениями и без.	3	2		1			1	Лабораторная работа 12
37.	Тема 37. Разбиение на слагаемые, разложение на множители.	3	2		1			1	Лабораторная работа 12
38.	Тема 38. Перебор перестановок. Алгоритм Нарайаны.	3	1		1			1	Лабораторная работа 12
39.	Тема 39. Перестановки. Циклы. Порядок. Таблица инверсий.	3	3		1	2		3	Лабораторная работа 12 Контрольная работа № 5
40.	Тема 40. Введение в вычислительную геометрию. Особенности. Задание геометрических объектов. Произведения векторов.	3	4		1			2	Лабораторная работа 13
41.	Тема 41. Взаимное расположение точек, прямых, лучей, отрезков, изотетичных прямоугольников.	3	2		1			1	Лабораторная работа 13
42.	Тема 42. Треугольник и окружность. Взаимное расположение с точками, прямыми, отрезками и лучами.	3	2		2			1	Лабораторная работа 14
43.	Тема 43. Выпуклые и невыпуклые многоугольники. Алгоритмы построения выпуклой оболочки. Точка и многоугольник. Площадь.	3	4		5	2		2	Лабораторная работа 15
						2	***0,5***	***33,5***	***Экзамен***
	Итого за 3 семестр 108 часов		***32***		***16***	***8***	***0,5***	***51,5***	
	ИТОГО **324 часа**		***96***		***48***	***22***	***1***	***157***	

Содержание разделов дисциплины:

1. **Введение. Алгоритмы и их представление**
   Задачи курса и порядок его изучения. Понятие алгоритма. Сущность алгоритмизации вычислительных процессов. Данные и алгоритмический процесс. Представление алгоритмов.
2. **Языки программирования. Классификация, особенности, история развития.**
   Понятие языка программирования. Классификация языков, история разработки. Общая характеристика языка программирования С ++. Процесс обработки программ на компьютере. Интерпретация и компиляция. Процесс отладки.
3. **Введение в язык С++. Структура программы, основные типы данных.**
   Введение в язык С++. Пример простой программы на С. Структура простой программы. Алфавит языка С++, идентификаторы. Использование комментариев. Данные в языке С++. Переменные и константы. Основные типы данных, их размер, машинное представление. Операция sizeof.
4. **Простейшие средства ввода и вывода.**
   Простейшие средства ввода-вывода. Потоки cin, cout. Заголовочный файл iomanip и функции форматированного вывода.
5. **Операции, выражения и операторы в языке С++.**
   Основные арифметические операции, порядок их выполнения. Операции ++ и --.
   Преобразование типов, операция приведения. Операции, выражения и операторы в языке С++. Составной оператор.
6. **Операторы ветвления и выбора в языке С++.**
   Логический тип данных. Преобразование к логическому типу. Логические операции. Оператор ветвления if в полной и сокращенной форме. Вложенные условные операторы. Тернарный условный оператор ?:.
   Множественный выбор: операторы switch и break.
7. **Программирование циклических процессов.**
   Программирование циклических процессов. Операторы циклов while, for, do-while. Ключевые слова break, continue. Операция запятая. Вложенные циклы. Вычисление функции заданной бесконечным рядом. Табулирование функции. Нахождение корня функции на отрезке.
8. **Указатели. Работа с динамической памятью. Арифметика указателей. Ссылки.**
   Структура памяти. Стек, куча. Функции динамического распределения и освобождения памяти. Нахождение адреса переменной: операция &. Первое знакомство с указателями. Типы указателей. Описание указателей. Арифметика указателей. Операция косвенной адресации *. Операторы new и delete.
9. **Одномерные массивы: статические, динамические. Контейнер vector(STL)**
   Статические массивы.
   Динамические массивы.
   Использование указателей для работы с массивами.
   Контейнер vector как динамический массив, методы работы.
10. **Линейный и бинарный поиск. Бинарный поиск по ответу.**
    Линейный и бинарный поиск. Бинарный поиск по ответу. Использование датчика псевдослучайных чисел. Оценка трудоемкости алгоритмов.
11. **Алгоритмы сортировки: подсчетом, поразрядные, квадратичные, быстрые.**
    Сортировка подсчетом и поразрядная сортировка.
    Квадратичные сортировки: пузырек, пузырек с флагом, шейкерная сортировка, сортировка вставками, сортировка выбором, сортировка Шелла.
    Быстрые сортировки: сортировка Хоара, сортировка слиянием.

Оценка трудоемкости алгоритмов, сравнение времени работы.

12. **Функции, способы передачи параметров. Указатель на функцию. Шаблоны. Аргументы командной строки.**

    Функции в языке C++. Объявление и определение. Формальные и фактические параметры. Способы передачи параметров. Возвращение значения функцией. Оператор return. Локальные переменные функции. Использование указателей и ссылок для передачи параметров между функциями. Механизм вызова функции, понятие стека. Структура программы, состоящей из нескольких функций.

    Шаблоны функций.

    Аргументы функции main().

    Указатели на функции. Передача указателей на функцию в качестве параметров.

13. **Рекурсия. Типы рекурсии. Оптимизация.**

    Рекурсия и ее использование. Восходящая и нисходящая рекурсия. Вложенная рекурсия. Переполнение стека вызовов. Оптимизация рекурсии. Классические задачи: НОД, быстрое возведение в степень, Ханойские башни.

14. **Символьная информация. Способы представления строк. Функции обработки строк.**

    Тип данных char. Кодировочные таблицы. Функции библиотеки cctype для категоризации и преобразования символов. Способы представления строк в языках программирования. Строки в стиле C. Библиотек cstring для работы с нуль-терминированными строками. Строки в стиле C++. Класс string и методы работы с ним.

15. **Средства потокового ввода и вывода. Файлы.**

    Текстовые и двоичные файлы. Потоки fstream, ifstream и ofstream. Порядок работы с файловыми потоками, обработка ошибок.

16. **Многомерные массивы. Способы представления.**

    Массивы и указатели. Указатели на указатели и многомерные массивы. Функции и многомерные массивы. Массивы указателей. Выделение и освобождение памяти. Утечки памяти. Вектор векторов. Передача в функцию.

17. **Использование структур для определения пользовательских типов данных.**

    Структуры, их описание, инициализация, доступ к элементам. Вложенные структуры. Указатели на структуры, их описание и использование.

18. **Использование библиотеки OpenGL.**

    Открытая библиотека OpenGL, базовые возможности. Подключение библиотеки, работа с цветом, основные графические примитивы. Основные шаги для построения минимальной программы. Обработка событий таймера, клавиатуры и мыши. Анимация.

19. **Жадные алгоритмы.**

    Понятие жадного алгоритма, применимость, понятие надежного шага. Задачи: покрытие точек отрезками, задача о выборе заявок, задача о непрерывном рюкзаке. Сжатие данных, алгоритм Хаффмана. Очередь с приоритетами, мин-, макс-кучи.

20. **Волновой алгоритм.**

    Определение волнового алгоритма. Примеры. Алгоритм Ли поиска кратчайшего пути на примере поиска пути в лабиринте (двумерной матрице).

21. **Динамическое программирование. Одномерная динамика. НОП, НВП.**

    Нисходящее, восходящее динамическое программирование. Одномерная динамика: наибольшая возрастающая подпоследовательность, восстановление решения.

22. **Двумерная динамика.**

Двумерная динамика: расстояние редактирования, взвешенное расстояние редактирования. Задача о рюкзаке с повторениями, рюкзак без повторений. Нахождение числа сочетаний.

23. **Динамика по профилю.**
    Задача о замощении доминошками. Задача о симпатичных узорах.
24. **Однонаправленные, двунаправленные, циклические списки. Реализации на базе массива и указателей.**
    Эффективная работа с памятью при реализации динамических структур данных. Минимизация фрагментации динамически распределяемой памяти. Односвязный, двусвязный линейные списки
25. **Другие последовательные динамические структуры данных. Стеки, деки, очереди.**
    Классические задачи: правильность расстановки скобок, вычисление выражения в обратной польской записи, минимум в скользящем окне.
26. **Куча. Пирамидальная сортировка.**
    Реализация кучи на массиве. Сортировка кучей.
27. **Бинарные деревья.**
    Бинарные деревья, деревья поиска. Обход дерева, поиск элемента, вставка, удаление элемента. Печать дерева. AVL-деревья. Малый и большой левые и правые повороты. Балансировка.
28. **Деревья общего вида.**
    Обход дерева, поиск элемента, вставка, удаление элемента. Печать дерева. Поиск элементов одного уровня.
29. **Основы объектно-ориентированного подхода.**
    Объектно-ориентированное программирование: реализация в языке инкапсуляции. Понятие класса, объекта. Конструкторы. Деструкторы. Описание своего класса и создание объектов. Друзья класса. Роль слова static. Методы класса. Модификаторы доступа. Вызов методов у экземпляров. Переопределение методов класса. Переопределение операторов.
30. **Статические и динамические библиотеки.**
    Препроцессор языка C++. Директива #define. Макроопределения с параметрами, их отличие от функций. Директива #include. Заголовочные файлы. Директивы #undef, #ifdef,#ifndef, #if, #else, #endif. Понятие условной компиляции. Библиотеки динамической компоновки (DLL), статические библиотеки, управляемые сборки. Примеры разработки статической и динамической библиотек. Явная загрузка динамической библиотеки.
31. **Арифметические алгоритмы. Проверка на простоту. Решено Эратосфена. Функция Эйлера.**
    Некоторые известные числовые алгоритмы. Быстрое возведение в степень. Алгоритм извлечения квадратного корня. Деление по модулю. Проверка числа на простоту. Решено Эратосфена. Функция Эйлера.
32. **НОД, алгоритм Евклида, расширенный алгоритм Евклида. Их приложения.**
    Алгоритм Евклида нахождения наибольшего общего делителя. Наименьшее общее кратное. Задачи, сводящиеся к вычислению НОД. Расширенный алгоритм Евклида. Диофантовы уравнения с двумя неизвестными. Задача о целочисленных точках отрезка. Задача о переливаниях.
33. **Системы счисления. Алгоритмы перевода целых и дробных чисел.** Алгоритмы перевода целых и дробных чисел из десятичной системы счисления в систему счисления с произвольным основанием и обратно. Специальные возможности языка

C++ для ввода, вывода и перевода целых чисел в системы счисления с различными основаниями.

34. Битовые операции. Перебор всех подмасок.

Битовые операции. Битовые операции: AND (&), OR (|), XOR (^), битовые сдвиги (<<, >>). Перебор подмножеств с использованием битовых масок. Перебор всех подмасок данной маски. Динамика по профилю с использованием битовых масок. Задача коммивояжера. Решение методом динамического программирования с использованием битовых масок.

35. Комбинаторные алгоритмы. Перебор размещений. Рекурсивный и итеративный алгоритмы.

Три алгоритма перебора всевозможных последовательностей заданной длины из нулей и единиц. Перебор подмножеств. Обобщение на перебор размещений с повторениями. Рекурсивный и итеративный алгоритмы.

36. Перебор сочетаний.

Рекурсивный и итеративный алгоритмы перебора сочетаний без повторений и сочетаний с повторениями.

37. Разбиение на слагаемые, разложение на множители.

Алгоритмы перебора всевозможных разбиений числа на натуральные слагаемые. Фиксированное количество неотрицательных слагаемых. Перебор всевозможных способов разложения числа на множители (>1).

38. Перебор перестановок. Алгоритм Нарайаны.

39. Перестановки. Циклы. Порядок. Таблица инверсий.

Алгоритмы нахождения обратной перестановки, разложения перестановки на независимые циклы, порядка перестановки, построения таблицы инверсий, восстановления перестановки по таблице инверсий.

40. Введение в вычислительную геометрию. Особенности. Задание геометрических объектов. Произведения векторов.

Погрешность вычислений и скорость работы. Способы задания геометрических объектов. Скалярное, косое, векторное и смешанное произведения, их применение к решению задач.

41. Взаимное расположение точек, прямых, лучей, отрезков, изотетичных прямоугольников.

Принадлежность точки прямой, лучу, отрезку. Расстояние от точки до прямой, луча, отрезка. Взаимное расположение двух прямых, отрезка и прямой, луча и прямой, двух лучей, двух отрезков. Пересечение и объединение нескольких отрезков на прямой. Пересечение и объединение изотетичных прямоугольников. Метод сканирующей прямой.

42. Треугольник и окружность. Взаимное расположение с точками, прямыми, отрезками и лучами.

Взаимное расположение точки и треугольника. Взаимное расположение окружности и точки, прямой, луча, отрезка, двух окружностей. Вписанные и описанные окружности.

43. Выпуклые и невыпуклые многоугольники. Алгоритмы построения выпуклой оболочки. Точка и многоугольник. Площадь.

Алгоритмы построения выпуклой оболочки: Джарвиса, Грэхема, Эндрю, Мелькмана, QuickHull. Принадлежность точки выпуклому многоугольнику. Принадлежность точки невыпуклому многоугольнику. Ориентированная площадь.

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция с элементами лекции-беседы – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

Лабораторное занятие – занятие в компьютерном классе, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

Консультация – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса используются:

– для формирования текстов материалов для промежуточной и текущей аттестации - приложение Microsoft Office;

- для разработки презентаций лекций - приложение Microsoft PowerPoint;

- для проведения лабораторных работ Microsoft Visual Studio 2017/2019;

- OpenGL программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. Имеет тип лицензий GNU-/EU/.

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

- Автоматизированная библиотечно-информационная система «БУКИ-NEXT» http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) основная литература

1. Основы программирования: учебное пособие / О. В. Власова, Н. П. Федотова, О. П. Якимова; Яросл. гос. ун-т им. П. Г. Демидова. - Ярославль: ЯрГУ, 2019. – 83

2. Основы программирования: учебное пособие / О. В. Власова, Н. П. Федотова, О. П. Якимова; Яросл. гос. ун-т им. П. Г. Демидова. - Ярославль: ЯрГУ, 2021. – 72

3. Павловская Т. А. C/C++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская; М-во образования РФ. - СПб.: Питер, 2010. - 460 с. - (Учебник для вузов).

б) дополнительная литература

1. Страуструп Б. Язык программирования C++. Стандарт C++11: краткий курс. / Б. Страуструп; перевод с англ. под ред. Н. Н. Мартынова - М.: БИНОМ, 2019. - 176 с.
2. Лафоре Р. Объектно-ориентированное программирование в C++. / Р. Лафоре; [пер. с англ. А. Кузнецова, М. Назарова, В. Шрага] - 4-е изд. - СПб.: Питер, 2016. - 923 с.
3. Кнут Д. Искусство программирования ЭВМ. Т.1, Основные алгоритмы- М.: Мир, 1976. – 712 с.
4. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. - М.: Мир, 1989.- 360с.
5. Ярославские олимпиады по информатике : сборник задач с решениями / С. Г. Волченков, П. А. Корнилов, Ю. А. Белов, Н. Л. Дашниц, В. А. Никулин, Н. И. Заводчикова, М., БИНОМ. Лаборатория знаний, 2010, 405с

в) ресурсы сети «Интернет»

1. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php).
2. Электронно-библиотечная система «Юрайт https://www.biblio-online.ru/
3. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru)
4. Электронно-библиотечная система «Лань» http://e.lanbook.com/
5. https://docs.microsoft.com/ru-ru/cpp/?view=msvc-170
6. https://www.cplusplus.com/

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:
-учебные аудитории для проведения занятий лекционного типа
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
-помещения для самостоятельной работы;
-помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде ЯрГУ.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока.

Автор(ы):

Доцент кафедры компьютерной безопасности и математических методов обработки информации, к.ф.-м.н. Федотова Н.П.

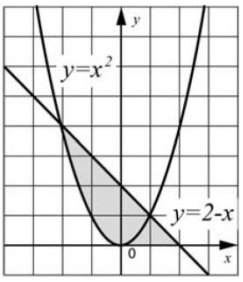
Приложение № 1 к рабочей программе дисциплины
«Основы программирования»
наименование дисциплины

Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине

1. Типовые контрольные задания и иные материалы, используемые в процессе текущего контроля успеваемости

Контрольная работа №1

1. Напишите программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы.



2. Назовём натуральное число N ($1000_8 \le N \le 777777_8$) счастливым, если суммы двух первых и двух последних цифр его восьмеричной записи равны. Найдите количество таких чисел.
   Для проверки числа разработайте функцию **`bool isHappy(int n);`**

3. Ниже записана программа. Получив на вход число x, эта программа печатает два числа, a и b. Укажите наименьшее из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 10.

```
int main()
{       int  x, a, b;
        cin >> x;
         a = 0; b = 1;
         while (x > 0)
                {       a ++;
                        b *= (x % 8);
                         x /= 8;
                }
        cout << a << ” “ << b ;
 return 0;
}
```

4. Вводится последовательность целых чисел (0 – конец последовательности), найти разность между наименьшим среди положительных и наибольшим среди отрицательных.

5. Дан целочисленный массив из N элементов, необходимо найти количество элементов в этом массиве, для которых последняя цифра в шестнадцатеричной записи и в восьмеричной записи одинаковая, и заменить все четные элементы массива на это количество. Гарантируется, что такой элемент есть. Напишите программу для решения этой задачи. В качестве результата программа должна вывести изменённый массив, по одному элементу в строке.
   Например, для исходного массива из 5 элементов
   `22 38 14 23 11`
   программа должна вывести числа **`3 3 3 23 11`**, по одному числу в строке.
   В программе разработайте функцию вычисляющую характеристику, функцию изменения массива и функцию печати массива.

Контрольная работа №2

1. Обработка двумерных массивов

Элементы исходной матрицы вводятся из текстового файла. Результаты выводить на экран и в результирующий текстовый файл. Матрицу выводить до и после преобразований.

Определить сумму и количество положительных чисел расположенных вне диагоналей матрицы B(n,n). Если нет положительных чисел, то поменять местами элементы главной и побочной диагоналей.

2. Строки

Описать функцию *IsIdent(S)* целого типа, проверяющую, является ли строка S допустимым идентификатором (набор букв и цифр, начинающийся с буквы). При утвердительном ответе возвращается 0. Если S является пустой строкой, то возвращается (–1), если строка начинается с цифры, то возвращается (–2). Если S содержит недопустимые символы, то возвращается номер первого недопустимого символа. Проверить с помощью этой функции пять данных строк.

Контрольная работа №3

1. Многочлен $P(x)=a_n x^n + a_{n-1}x^{n-1} +...+ a_1 x + a_0$ с целыми коэффициентами можно представить в виде линейного односвязного списка, причем если $a_i=0$, то соответствующее звено не включать в список. Определить логическую функцию *bool Equal*(*list p, list q*), проверяющие на равенство многочлены *p* и *q*
2. Даны натуральные ***n*** и ***k***. По кругу выписывают все натуральные числа от 1 до ***n***. Сначала отсчитывают ***k*** -ое число, начиная с первого, и удаляют его. Затем от него отсчитывают ***k*** чисел и ***k*** -ое удаляют, и т.д. Процесс останавливается, когда остаётся одно число. Требуется найти это число.

   Задача была поставлена Иосифом Флавием (Flavius Josephus) ещё в 1 веке (правда, в несколько более узкой формулировке: при ***k*** =2).

   Решать эту задачу моделированием на базе линейного односвязного списка.

Контрольная работа №4

1. Написать функцию, подсчитывающую количество вершин в дереве равных К.
   int CountVertex(Tree Top, int K);*
2. Выполнить преобразование дерева.(*void Transform (Tree* Top))*

   Дан указатель *Top* на корень непустого дерева. Для каждой вершины дерева, имеющей две дочерние вершины, поменять местами значения дочерних вершин (т. е. значения их полей *info*).

Контрольная работа №5

Дано натуральное число N. Рассмотрим его разбиение на натуральные слагаемые. Два разбиения, отличающихся только порядком слагаемых, будем считать за одно, поэтому можно считать, что слагаемые в разбиении упорядочены по невозрастанию. На вход подается единственное число $N \leq 40$. Необходимо вывести все разбиения числа N на натуральные слагаемые в лексикографическом порядке.

Пример. Для = 5 существуют такие способы разбиения на слагаемые:

1 + 1 + 1 + 1 + 1 = 2 + 1 + 1 + 1 = 2 + 2 + 1 = 3 + 1 + 1 = 3 + 2 = 4 + 1 = 5

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

1. Алгоритм. Свойства. Способы записи.
2. Машина Тьюринга. Примеры.
3. Архитектура компьютера.
4. Структура программы. Заголовочные файлы, файлы реализации. Препроцессор.
5. Понятие переменной. Типы данных. Области видимости. Время жизни.
6. Арифметические, логические операции. Приоритет.
7. Побитовые операции. Операции сдвига.
8. Арифметика указателей. Связь между указателями и массивами. Ссылки. Примеры.
9. Потоковый ввод-вывод (iostream). Использование манипуляторов. Примеры.
10. Условный оператор. Оператор выбора. Примеры.
11. Виды циклов. Примеры.
12. Псевдослучайные числа. Получение целого(вещественного) числа из заданного отрезка [a, b] Примеры.
13. Одномерные массивы. Статическая память. Динамическая память. Примеры.
14. Сортировка подсчетом, поразрядная сотрировка. Трудоемкость. Примеры.
15. Сортировка выбором. Трудоемкость. Примеры.
16. Сортировка обменом. Модификации. Трудоемкость. Примеры.
17. Сортировка вставками. Модификации. Трудоемкость. Примеры.
18. Сортировка Шелла. Трудоемкость. Примеры.
19. Быстрая сортировка. Сортировка Хоара. Трудоемкость. Примеры.
20. Быстрая сортировка. Сортировка слиянием. Трудоемкость. Примеры.
21. Задачи поиска минимума (максимума) и его номера в одномерном массиве. Примеры.
22. Поиск в одномерном массиве (Линейный, бинарный). Трудоемкость. Примеры.
23. Изменение массива (добавление, удаление элементов). Трудоемкость. Примеры.
24. Статические двумерные массивы. Примеры.
25. Способы создания динамических двумерных массивы. Примеры.
26. Двумерный массив в виде массива массивов. Примеры.
27. Функции. Способы передачи параметров. Примеры.
28. Рекурсивные функции. Алгоритм Евклида (НОД). Примеры.
29. Шаблонные функции. Примеры.
30. Работа с файлами(fstream). Примеры.
31. Функции работы со строками. Библиотека string.h Примеры.
32. Функции работы со строками. Библиотека string Примеры.
33. Структуры (struct). Примеры.
34. Стек. Реализация на базе массива. Примеры.
35. Динамический стек. Реализация на базе линейного списка. Примеры.
36. Стек (STL) Примеры.
37. Очередь. Реализация на базе массива. Примеры.
38. Динамическая очередь. Реализация на базе линейного списка. Примеры.
39. Очередь (STL) Примеры.
40. Очередь с приоритетами. Основные операции. Примеры.
41. Дек. Реализация на базе массива. Примеры.
42. Динамический дек. Примеры.
43. Линейный список. Примеры.

44. Двунаправленный линейный список. Примеры.
45. Деревья. Способы представления.
46. Деревья общего вида. Реализация. Примеры.
47. Бинарное дерево поиска. Реализация. Обход. Примеры.
48. АВЛ-дерево. Балансировка. Примеры.
49. АВЛ-дерево. Алгоритм добавление вершины. Примеры.
50. АВЛ-дерево. Алгоритм удаления вершины. Примеры.
51. Способы организации дерева произвольного вида. Примеры.
52. Динамическое программирование на примере вычисления чисел Фибоначчи. Примеры.
53. Динамическое программирование сверху вниз. Примеры.
54. Динамическое программирование снизу вверх. Примеры.
55. Динамическое программирование. Нахождение наибольшей возрастающей подпоследовательности. Примеры.
56. Двумерная динамика. Расстояние редактирования. Примеры.
57. Динамика по профилю. Задачи о замощении доминошками и симпатичных узорах.
58. Задача о рюкзаке. С повторениями. Примеры.
59. Задача о рюкзаке. Без повторений. Примеры.
60. Жадные алгоритмы. Покрытие точек единичными отрезками. Примеры.
61. Жадные алгоритмы. Задача о выборе заявок. Примеры.
62. Жадные алгоритмы. Задача о непрерывном рюкзаке. Примеры.
63. Жадные алгоритмы. Кодирование Хаффмана. Примеры.
64. Перебор размещений. Рекурсивный и итеративный алгоритмы.
65. Три алгоритма перебора всевозможных последовательностей заданной длины из нулей и единиц. Перебор подмножеств.
66. Перебор размещений с повторениями. Рекурсивный и итеративный алгоритмы.
67. Рекурсивный и итеративный алгоритмы перебора сочетаний без повторений.
68. Рекурсивный и итеративный алгоритмы перебора сочетаний с повторениями.
69. Алгоритмы перебора всевозможных разбиений числа на натуральные слагаемые. Фиксированное количество неотрицательных слагаемых.
70. Перебор всевозможных способов разложения числа на множители (>1).
71. Перебор перестановок. Алгоритм Нарайаны.
72. Алгоритмы нахождения обратной перестановки, разложения перестановки на независимые циклы, порядка перестановки.
73. Алгоритмы построения таблицы инверсий, восстановления перестановки по таблице инверсий.
74. Взаимное расположение точек, прямых, лучей, отрезков.
75. Задача об изотетичных прямоугольников. Метод сканирующей прямой.
76. Взаимное расположение окружности и точки, прямой, луча, отрезка, двух окружностей.
77. Взаимное расположение точки и треугольника. Площадь треугольника. Вписанные и описанные окружности.
78. Алгоритм Джарвиса построения выпуклой оболочки.
79. Алгоритм Грэхема построения выпуклой оболочки.
80. Алгоритм Эндрю построения выпуклой оболочки.
81. Алгоритм Мелькмана построения выпуклой оболочки.

82. Алгоритм QuickHull построения выпуклой оболочки.
83. Принадлежность точки выпуклому многоугольнику.
84. Принадлежность точки невыпуклому многоугольнику.
85. Понятие ориентированной площади. Алгоритмы нахождения площади многоугольника.

Примерный билет к экзамену после второго семестра по дисциплине

Теоретический вопрос

Динамическая очередь. Реализация на базе линейного списка. Примеры.

Задание 1

Дана строка, заканчивающаяся точкой. Среди символов строки особую роль играет символ #, появление которого в строке означает удаление предыдущего символа. Соответственно, k символов # подряд отменяют k предыдущих символов строки, если таковые имеются.

Требуется написать программу, преобразующую строку с учетом указанного значения символа #.

Замечания:

1) Если в какой-то момент перед некоторым символом # на этой строке не осталось символов, то его следует игнорировать.

2) В выходную строку символы # выводить не следует ни в каком случае.

3) Если в результате преобразований все символы в строке были удалены, то следует вывести пустую строку.

Формат входных данных

Признаком окончания ввода служит точка (символ "."). Строка содержит не более 200 символов.

Входная информация	Выходная информация
Hello ww#orld! # a##abc# the##he end.	Hello world ab the end.

Задание 2

Для заданной числовой последовательности A[0..N] найти длину максимальной подпоследовательности (могут идти не подряд) элементы которой образуют арифметическую прогрессию

Формат входных данных:

Число N (1<=N<=1000).

Элементы последовательности, A[i] (-100<=A[i]<=100).

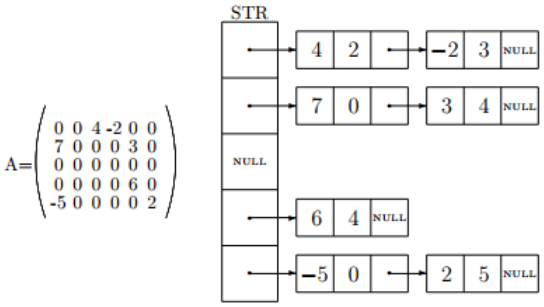
Формат выходных данных:

Длина искомой подпоследовательности. Сама подпоследовательность.

Входные данные	Выходные данные
5 5 -3 6 15 -10	3 -3 6 15

Задание 3.

Разреженная квадратная матрица A (матрица с относительно малым количеством ненулевых элементов) может храниться в памяти в виде набора списков STR, как показано на рисунке. Каждый ненулевой элемент матрицы хранится в структуре, имеющей три поля: int Val; — значение элемента, int Row; — номер столбца, Next — указатель на следующий ненулевой элемент строки. Компонента STR[i] массива STR — это указатель на первый ненулевой элемент i-й строки. Напишите программу, которая строит разреженную матрицу, затем напишите функцию, которая выполняет преобразование: к элементам главной диагонали прибавьте 2, а затем поменяйте местами строки матрицы (первую с последней, вторую с предпоследней и т.д.).



Приложение № 2 к рабочей программе дисциплины
«Основы программирования»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Основы программирования» являются академические лекции с элементами лекции-беседы. Это связано с тем, что для формирования алгоритмического мышления необходим опыт формализации задач и их поэтапного решения. По большинству тем предусмотрены лабораторные занятия в рассматриваемом и смежном курсе «Практикум по основам программирования», на которых происходит закрепление лекционного материала.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, требующих разработки алгоритма и написания программы, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются на лекциях и практических занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – сформировать алгоритмическое мышление. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и практических занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач. Задачи сдаются через систему Яндекс-контест, обязательны к решению три задачи из каждого тура.

В конце 2 и 3 семестров студенты сдают экзамен. Экзамен принимается по экзаменационным билетам, каждый из которых включает в себя один теоретический вопрос и три задачи. На самостоятельную подготовку к экзамену выделяется 3 дня, во время подготовки к экзамену предусмотрена групповая консультация.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, с подробно разобранными решениями задач. К таким можно отнести следующие издания:

1. Основы программирования: учебное пособие / О. В. Власова, Н. П. Федотова, О. П. Якимова; Яросл. гос. ун-т им. П. Г. Демидова. - Ярославль: ЯрГУ, 2019. – 83
2. Основы программирования: учебное пособие / О. В. Власова, Н. П. Федотова, О. П. Якимова; Яросл. гос. ун-т им. П. Г. Демидова. - Ярославль: ЯрГУ, 2021. – 72
3. Ярославские олимпиады по информатике : сборник задач с решениями / С. Г. Волченков, П. А. Корнилов, Ю. А. Белов, Н. Л. Дашниц, В. А. Никулин, Н. И. Заводчикова, М., БИНОМ. Лаборатория знаний, 2010, 405с

Проверку решений задач по различной тематике можно выполнять на специализированных платформах:

1. https://informatics.msk.ru/
2. https://codeforces.com/

Рекомендуется дополнительно прослушать бесплатные обучающие курсы по языку C++ на платформе Stepik
https://stepik.org/catalog/search?free=true

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов, указанных в разделе 7