

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

"Московский авиационный институт
(национальный исследовательский университет)"

УТВЕРЖДАЮ

Проректор по учебной работе

_____ Козорез Д.А.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (000205543)

Программирование

(указывается наименование дисциплины по учебному плану)

Направление подготовки	Информатика и вычислительная техника
Квалификация выпускника	Бакалавр
Профиль подготовки	Автоматизированные системы обработки информации и управления
Форма обучения	очная
	(очно, очно-заочное, заочное)
Выпускающая кафедра	МСиИТ
Обеспечивающая кафедра	МСиИТ
Кафедра-разработчик рабочей программы	МСиИТ

Семестр	З.Е.	Трудоемкость, час.	Лекций, час.	Практич. занятий, час.	Лаборат. работ, час.	СРС, час	Экзамен-нов, час.	Форма промежуточног о контроля
3	3	108	18	0	36	54	0	Зч
4	5	180	32	0	44	68	36	Э
Итого	8	288	50	0	80	122	36	

Москва
2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разделы рабочей программы

1. Цели освоения дисциплины. Перечень планируемых результатов обучения.
2. Место дисциплины в структуре образовательной программы.
3. Структура и содержание дисциплины.
4. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.
5. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.
6. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.
8. Методические указания для обучающихся по освоению дисциплины.
9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.
10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Приложения к рабочей программе дисциплины

Приложение 1. Аннотация рабочей программы

Приложение 2. Прикрепленные файлы

Программа составлена в соответствии с требованиями СУОС МАИ, разработанного на основе ФГОС ВО (3++) по направлению 09.03.01 Информатика и вычислительная техника

Авторы программы:

Челпанов А. В.

Заведующий обеспечивающей кафедрой МСиИТ

Программа одобрена:

Заведующий выпускающей кафедрой
МСиИТ

Директор выпускающего филиала СТ

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ КОМПЕТЕНЦИЙ И РЕЗУЛЬТАТОВ ОБУЧЕНИЯ.

Целью освоения дисциплины Программирование является достижение следующих результатов освоения(РО):

N	Шифр	Результат обучения
1	В-1(ОПК-11.1)	Владеть методиками использования программных средств для решения практических задач
2	В-1(ОПК-4.1)	Владеть методами разработки алгоритмического и специального программного обеспечения вычислительной техники
3	В-1(ОПК-4.3)	Владеть языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее, чем на одном из алгоритмических языков
4	В-2(ОПК-4.2)	Владеть навыками программирования в современных средах разработки программных приложений
5	З-1(ОПК-11.1)	Знать методы разработки алгоритмического и специального программного обеспечения вычислительной техники
6	У-1(ОПК-10.2)	Уметь ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы
7	У-1(ОПК-4.1)	Уметь работать с современными системами программирования, включая объектно-ориентированные

Перечисленные РО являются основой для формирования следующих компетенций:

N	Шифр	Компетенция
1	ОПК-4	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
2	ОПК-10	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
3	ОПК-11	Способен осваивать методики использования программных средств для решения практических задач

Индикаторы достижения компетенций, служащие для проверки сформированности части соответствующей компетенции:

N	Шифр	Индикатор компетенций
1	ОПК-10.2	Составляет алгоритмы, пишет и отлаживает коды на языке программирования, тестирует работоспособность программы, интегрирует программные модули
2	ОПК-11.1	Демонстрирует знание методик использования программных средств для решения практических задач
3	ОПК-4.1	Демонстрирует знание современных информационных технологий и методов их использования при решении задач профессиональной деятельности

4	ОПК-4.2	Выбирает современные информационные технологии, предпочтительно отечественного производства, для решения задач профессиональной деятельности
5	ОПК-4.3	Применяет необходимые информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.

Дисциплина Программирование является предшествующей и последующей для следующих дисциплин:

N	Предшествующие дисциплины	Последующие дисциплины
1	Информатика	Организация ЭВМ
2		Операционные системы
3		Итоговая гос. аттестация
4		Информационный менеджмент
5		Метрология, стандартизация и сертификация

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость практики составляет 8 зачетных(ые) единиц(ы), 288 часа(ов).

Модуль	Раздел	Лекции	Практич. занятия	Лаборат. работы	СРС	Всего часов	Всего с экзаменами и курсовыми
Программирован ие (3 семестр)	Введение	2	0	16	16	34	108
	Элементы общей теории языков программирования	2	0	8	10	20	
	Язык программирования с++	14	0	12	28	54	
Программирован ие (4 семестр)	Структуры данных	12	0	32	17	61	180
	Объектно-ориентир ованное и модульное программирование	8	0	12	12	32	
	Введение в языки программирования лисп, пролог, рубин, постскрипт и SQL	4	0	0	2	6	
	Общая теория языков программирования	4	0	0	2	6	

	Технологический процесс разработки программ	4	0	0	1	5	
Всего		50	0	80	88	218	288

3.1. Лекции

№ п/п	Раздел дисциплины	Объем часов	Тема лекции
1	1.1.Введение	2	Структура курса. Алгоритм. Основные способы записи алгоритмов. Блок-схема. Языки программирования низкого уровня, ЯП высоко уровня.
2	1.2.Элементы общей теории языков программирования	2	Общее определение языка программирования. Модель передачи сообщения: семантика, синтактика, прагматика. Классификация ошибок в программе.
3	1.3.Язык программирования c++	2	Синтаксические диаграммы. Алфавит, лексика, организация программы. Скалярные типы. Описание переменных, констант и типов. Преобразование типов.
4	1.3.Язык программирования c++	4	Регулярные типы. Комбинированные типы. Строки. Множества. Операторы.
5	1.3.Язык программирования c++	4	Функции. Способы передачи параметров. Динамические массивы. Рекурсия: прямая и косвенная. Внешние подпрограммы.
6	1.3.Язык программирования c++	4	Понятие об объектах. Ссылочные типы. Видимость имен. Динамические, временные и статические данные. Выражения. Файловые типы и переменные. Ввод-вывод.
7	2.1.Структуры данных	4	Стеки, очереди, деки - определения и основные операции.
8	2.1.Структуры данных	8	Бинарные деревья, списки — определения и основные операции.
9	2.2.Объектно-ориентированное и модульное программирование	4	Назначение и определение объектно-ориентированного подхода к программированию. Наследуемость и полиморфизм.
10	2.2.Объектно-ориентированное и модульное программирование	4	Виртуальные функции, конструкторы и деструкторы. Инкапсуляция. Примеры реализации концепций ООП. Визуальное и модульное программирование.
11	2.3.Введение в языки программирования лисп, пролог, рубин, постскрипт и SQL	4	Языки программирования: лисп, пролог, SQL, постскрипт, рубин.
12	2.4.Общая теория языков программирования	4	Систематическое изучение языков программирования с разных позиций. Основная цель программирования. Главная проблема программирования.

13	2.5.Технологический процесс разработки программ	4	Методы проектирования. Характеристики алгоритмов и программ. Использование процедурного, объектного, функционального и пр. программирования
Итого:		50	

3.2. Содержание лекций

- 1.1.1. Структура курса. Алгоритм. Основные способы записи алгоритмов. Блок-схема. Языки программирования низкого уровня, ЯП высоко уровня. (АЗ: 2, СРС: 4)**
Тип лекции: Информационная лекция
Форма организации: Лекция
- 1.2.1. Общее определение языка программирования. Модель передачи сообщения: семантика, синтактика, прагматика. Классификация ошибок в программе. (АЗ: 2, СРС: 2)**
Тип лекции: Информационная лекция
Форма организации: Лекция
- 1.3.1. Синтаксические диаграммы. Алфавит, лексика, организация программы. Скалярные типы. Описание переменных, констант и типов. Преобразование типов. (АЗ: 2, СРС: 2)**
Тип лекции: Информационная лекция
Форма организации: Лекция
- 1.3.2. Регулярные типы. Комбинированные типы. Строки. Множества. Операторы. (АЗ: 4, СРС: 4)**
Тип лекции: Информационная лекция
Форма организации: Лекция
- 1.3.3. Функции. Способы передачи параметров. Динамические массивы. Рекурсия: прямая и косвенная. Внешние подпрограммы. (АЗ: 4, СРС: 2)**
Тип лекции: Информационная лекция
Форма организации: Лекция
- 1.3.4. Понятие об объектах. Ссылочные типы. Видимость имен. Динамические, временные и статические данные. Выражения. Файловые типы и переменные. Ввод-вывод. (АЗ: 4, СРС: 4)**
Тип лекции: Информационная лекция
Форма организации: Лекция
- 2.1.1. Стеки, очереди, деки - определения и основные операции. (АЗ: 4, СРС: 4)**
Тип лекции: Информационная лекция
Форма организации: Лекция

2.1.2. Бинарные деревья, списки — определения и основные операции. (АЗ: 8, СРС: 4)

Тип лекции: Информационная лекция

Форма организации: Лекция

2.2.1. Назначение и определение объектно-ориентированного подхода к программированию. Наследуемость и полиморфизм. (АЗ: 4, СРС: 2)

Тип лекции: Информационная лекция

Форма организации: Лекция

2.2.2. Виртуальные функции, конструкторы и деструкторы. Инкапсуляция. Примеры реализации концепций ООП. Визуальное и модульное программирование. (АЗ: 4, СРС: 2)

Тип лекции: Информационная лекция

Форма организации: Лекция

2.3.1. Языки программирования: лисп, пролог, SQL, постскрипт, рубин. (АЗ: 4, СРС: 2)

Тип лекции: Информационная лекция

Форма организации: Лекция

2.4.1. Систематическое изучение языков программирования с разных позиций. Основная цель программирования. Главная проблема программирования. (АЗ: 4, СРС: 2)

Тип лекции: Информационная лекция

Форма организации: Лекция

2.5.1. Методы проектирования. Характеристики алгоритмов и программ. Использование процедурного, объектного, функционального и пр. программирования (АЗ: 4, СРС: 1)

Тип лекции: Информационная лекция

Форма организации: Лекция

3.3. Практические занятия

Не предусмотрено учебным планом.

3.4. Лабораторные работы

№ п/п	Раздел дисциплины	Объем часов	Наименование лабораторной работы
1	1.1.Введение	8	Среда программирования. Кодирование алгоритма на ЯП. Вычисление средних значений (гармонического, арифметического, ...)
2	1.1.Введение	8	Вычисление простых чисел, разработка функции-предиката.
3	1.2.Элементы общей теории языков программирования	8	Перевод программ с ЯП паскаль на с++

4	1.3.Язык программирования c++	4	Вычисление факториала, чисел Фибоначчи, заданной функции. Использование операций сдвигов.
5	1.3.Язык программирования c++	8	Исследование временных свойств алгоритмов сортировки данных: выбором, пузырьком, хэшем, деревом, методом Шелла, методом Хоара
6	2.1.Структуры данных	4	Реализация поиска с заменой. Разбор строки с адресом ресурса. Реализация базовой работы с множествами
7	2.1.Структуры данных	4	Реализация матричных операций в поле классов вычетов по заданному модулю, вычисление определителя матрицы
8	2.1.Структуры данных	4	Реализация циклического однонаправленного списка и поиска заданного элемента в нем, реализация ввод/вывода данных из файла
9	2.1.Структуры данных	4	Реализация стека
10	2.1.Структуры данных	4	Реализация дека
11	2.1.Структуры данных	4	Реализация двунаправленного списка
12	2.1.Структуры данных	8	Статистическая обработка последовательности случайных чисел
13	2.2.Объектно-ориентированное и модульное программирование	8	Реализация стека, дека, очереди, одно- и двунаправленного списков, бинарного дерева с динамическим распределением памяти методами ООП.
14	2.2.Объектно-ориентированное и модульное программирование	4	Построение графиков заданных функций и других графических объектов, используя средства наследования ООП и графический сервер
Итого:		80	

3.5.Содержание лабораторных работ

- 1.1.1. Среда программирования. Кодирование алгоритма на ЯП. Вычисление средних значений (гармонического, арифметического, ...) (АЗ: 8, СРС: 4)**
Форма организации: Лабораторная работа
- 1.1.2. Вычисление простых чисел, разработка функции-предиката. (АЗ: 8, СРС: 8)**
Форма организации: Лабораторная работа
- 1.2.1. Перевод программ с ЯП паскаль на c++ (АЗ: 8, СРС: 8)**
Форма организации: Лабораторная работа
- 1.3.1. Вычисление факториала, чисел Фибоначчи, заданной функции. Использование операций сдвигов. (АЗ: 4, СРС: 4)**
Форма организации: Лабораторная работа

1.3.2. Исследование временных свойств алгоритмов сортировки данных: выбором, пузырьком, хэшем, деревом, методом Шелла, методом Хоара (АЗ: 8, СРС: 12)
Форма организации: Лабораторная работа

2.1.1. Реализация поиска с заменой. Разбор строки с адресом ресурса. Реализация базовой работы с множествами (АЗ: 4, СРС: 1)
Форма организации: Лабораторная работа

2.1.2. Реализация матричных операций в поле классов вычетов по заданному модулю, вычисление определителя матрицы (АЗ: 4, СРС: 2)
Форма организации: Лабораторная работа

2.1.3. Реализация циклического однонаправленного списка и поиска заданного элемента в нем, реализация ввод/вывода данных из файла (АЗ: 4, СРС: 1)
Форма организации: Лабораторная работа

2.1.4. Реализация стека (АЗ: 4, СРС: 1)
Форма организации: Лабораторная работа

2.1.5. Реализация дека (АЗ: 4, СРС: 1)
Форма организации: Лабораторная работа

2.1.6. Реализация двунаправленного списка (АЗ: 4, СРС: 2)
Форма организации: Лабораторная работа

2.1.7. Статистическая обработка последовательности случайных чисел (АЗ: 8, СРС: 1)
Форма организации: Лабораторная работа

2.2.1. Реализация стека, дека, очереди, одно- и двунаправленного списков, бинарного дерева с динамическим распределением памяти методами ООП. (АЗ: 8, СРС: 4)
Форма организации: Лабораторная работа

2.2.2. Построение графиков заданных функций и других графических объектов, используя средства наследования ООП и графический сервер (АЗ: 4, СРС: 4)
Форма организации: Лабораторная работа

3.6. Курсовые работы и проекты по дисциплине

2.1. Исследование алгоритмов сортировки в среде ОС Windows/Linux

Тематика:

Трудоемкость(СРС): 34

Прикрепленные файлы: варианты кр.pdf

3.7. Промежуточная аттестация

1. Зачет (3 семестр)

Прикрепленные файлы: Зачет (3 семестр).pdf, вопросы к зачету.pdf

2. Экзамен (4 семестр)

Прикрепленные файлы: Экзамен (4 семестр).pdf, вопросы к экзамену.pdf

4. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Основная и дополнительная литература по дисциплине
2. Ресурсы информационно-телекоммуникационной сети «Интернет».
3. Ресурсы научно-технической библиотеки МАИ.
4. Информационные стенды кафедры.

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Описание показателей, критерии оценивания компетенций и описание шкал оценивания осуществляются в соответствии с Положением о балльно-рейтинговой системе оценки результатов обучения студентов по дисциплине (Приказ №42 от 04.04.2014 «Об утверждении положения «Рейтинг по дисциплине»).

Для оценивания интегрированных и практико-ориентированных заданий обучающихся используются следующие критерии по 100-балльной шкале:

1. Формулирование представленной информации в виде проблемы;
2. Предложение способа решения проблемы;
3. Обоснование способа решения проблемы;
4. Демонстрация способа решения проблемы.

Оценивание осуществляется по следующей шкале:

100-балльная шкала	Результат освоения
менее 40	Критерий не сформирован
41-70	Критерий четко не выражен
71-100	Критерий выражен четко

Для оценивания ситуационных заданий используется следующая шкала:

100-балльная шкала	Результат освоения
менее 30	обучающийся не может сформулировать проблему, представленную в задании
31-50	обучающийся формулирует поставленную задачу, у него сформированы изолированные знания и умения, однако отсутствуют интегрированные понятия и навыки, в результате чего допущены ошибки в решении и задание не выполнено
51-80	задание выполнено, обучающийся применяет знания для решения поставленной проблемы, однако не сформированы компетенции, вследствие чего обучающийся испытывает затруднения в демонстрации способов решения задачи
81-100	задание выполнено как в теоретическом, так и в практическом плане, обучающийся легко демонстрирует свою компетентность по данному вопросу

Фонды оценочных средств, позволяющие оценить результаты обучения, включают в себя:

- вопросы к промежуточной аттестации.

Перечень компетенций и этапы их формирования приведены в следующей таблице:

N	Шифр	Компетенция	Этапы формирования компетенции
1	ОПК-4	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Владеть методами разработки алгоритмического и специального программного обеспечения вычислительной техники Владеть языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее, чем на одном из алгоритмических языков Владеть навыками программирования в современных средах разработки программных приложений Уметь работать с современными системами программирования, включая объектно-ориентированные Семестры - 3, 4
2	ОПК-10	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Уметь ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы Семестры - 3, 4

3	ОПК-11	Способен осваивать методики использования программных средств для решения практических задач	Владеть методиками использования программных средств для решения практических задач Знать методы разработки алгоритмического и специального программного обеспечения вычислительной техники Семестры - 3, 4
---	--------	----------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Вопросы к промежуточной аттестации

"Программирование"

1. Зачет (3 семестр)

Прикрепленные файлы: Зачет (3 семестр).pdf, вопросы к зачету.pdf

2. Экзамен (4 семестр)

Прикрепленные файлы: Экзамен (4 семестр).pdf, вопросы к экзамену.pdf

6. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

а) Основная литература:

- 1. International standard ISO/IEC 14882. Programming languages — C++. //Third edition 2011-09-01. — 1356 p.
- 2. International standard ISO/IEC 14882. Programming languages — C++. //Second edition 2003-10-15. — 757 p.
- 3. International standard ISO/IEC 9899. Programming languages — C. //Second edition 2000-5-22. — 554 p.
- 4. М. И. Болски Язык программирования Си. Справочник /Пер. с англ. — М.: «Радио и связь», 1988 г. — 96 с.
- 5. Р. Лафорте Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. — СПб.: Питер, 2011. — 928 с.
- 6. Бьерн Страуструп Язык программирования C++, 3-е изд. /Пер. с англ. — СПб.; М.: «Невский диалект» — «Издательство БИНОМ», 1999 г. — 991 с.
- 7. Дерк Луис С и C++ — М.: Бином, 1997.
- 8. Кауфман В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс] /В. Ш. Кауфман — М.: ДМК Пресс, 2010. — 464 с. Режим доступа: <http://www.znaniyum.com/bookread.php?book=409077>
- 9. Новиков П.В. Объектно-ориентированное программирование : учеб. пособие к лабораторным работам / П.В. Новиков; под ред. О.М.Брехова; МАИ (Нац. исслед. ун-т). - Москва : МАИ, 2019. - 123 с. : ил. - (Учебное пособие). - Библиогр.: с.122 (17 назв.). - ISBN 978-5-4316-0625-0.

б) Дополнительная литература:

- 1. Ахо В. Структуры данных и алгоритмы /В. Ахо, Дж. Хопкрофт, Д. Ульман — М., Спб., Киев: Издательский дом “Вильямс”, 2010.
2. Братко И. Программирование на языке ПРОЛОГ для искусственного интеллекта /И. Братко — М.: Мир, 1990.
3. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста /Р. Мартин — СПб.: Питер, 2011. — 464 с.
4. Немнюгин С. А. Turbo Pascal. Практикум /С. А. Немнюгин — СПб.: Питер, 2007. — 268 с.
5. Себеста Р. У. Основные концепции языков программирования /Р. У. Себеста — М.: Издательский дом “Вильямс”, 2001. — 672 стр.
6. Страуструп Б. Язык программирования С++ /Б. Страуструп — М.: Изда-тельство Бином, 2011. — 1136 с.
7. Сузи Р. А. Язык программирования Python: учебное пособие. 2-е изд., ис- пр. /Р. А. Сузи — М: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2007. — 326 с.
8. Брюс А. Тейт Ruby on Rails. Быстрая веб-разработка /Брюс А. Тейт, Курт Ниббс — СПб.: БХВ-Петербург, 2008. — 224 с.
9. Фиайли К. SQL /К. Фиайли — М.: ДМК Пресс, 2008. — 451 с.
10. Фитцджеральд М. Изучаем Ruby /М. Фитцджеральд — СПб: БХВ- Петербург, 2008. — 366 с.
11. Чад Фоулер Rails. Сборник рецептов /Чад Фоулер — СПб.: 2008. — 256 с.
12. Хювнен Э. Мир Лиспа /Э. Хювнен, Й. Сеппянен — М.: Мир, 1990. — 788 с.
13. Шохирев М. В. Язык программирования Perl 5 /М. В. Шохирев — М: Ин-тернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2007. — 279 с.
14. Чиртик А. Программирование в Delphi /А. Чиртик — СПб.: Питер, 2010 — 400 с.
15. Postscript language reference manual /Adobe System Incorporated. 3rd ed. 1999. 909 p.

**7. ПЕРЕЧЕНЬ РЕСУРСОВ
ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ
«ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ
ДИСЦИПЛИНЫ**

Для обеспечения образовательного процесса по дисциплине обучающимся предоставляется возможность круглосуточного дистанционного индивидуального доступа к электронным библиотечным системам из любой точки, в которой имеется доступ к сети «Интернет».

Наименование ресурса	Интернет-ссылка на ресурс
"ZNANIUM.COM"	
Договор № 4855 эбс/027-1-3200-20 от 08.12.2020 с ООО "ЗНАНИУМ" С «18»12.2020 г. по «17»12.2021 г	http://znanium.com
Договор № эбс/027-1-3026-21 от 22.12.2021 с ООО "ЗНАНИУМ" С «15»12.2021 г. по «31»12.2022 г	https://znanium.com/
Договор № эбс/027-1-2586-22 от 07.12.2022 с ООО "ЗНАНИУМ" С «20»12.2022 г. по «31»12.2023 г	

ООО "Издательство Лань"	
Договор № 027-1-0234-21 от 18.02.2021 года с ООО "Издательство Лань" С «22»_02. 2021г. по « 21» 02.2022 г	e.lanbook.com
Договор № 027-1-0234-21 от 18.02.2021 года с ООО "ЭБС Лань" С «22»_02. 2021г. по « 21» 02.2022	
Договор № СЭБ 027-0-0400-21 от 15.09.2021 года с ООО "ЭБС Лань" С «15»_09. 2021г. по « 14» 09.2024	
Договор № 027-1-0169-22 от 07.02.2022 года с ООО "Издательство Лань" С «22»_02. 2022г. по « 21» 02.2023 г	
Договор № 027-1-0168-22 от 07.02.2022 года с ООО "ЭБС Лань" С «22»_02. 2022г. по « 21» 02.2023	
ООО "Электронное издательство ЮРАЙТ"	
Электронная библиотечная система ЮРАЙТ. ЭБС "Легендарные книги"	http://biblio-online.ru , https://biblio-online.ru/catalog/legendary
Договор № 027-1-3191-20 от 04.12.2020г ООО "Электронное издательство ЮРАЙТ" для СПО С «04»12.2020 г. по «03»12.2021	https://urait.ru/
Договор № 027-1-3194-20 от 04.12.2020г. с ООО "Электронное издательства ЮРАЙТ" С «04»12.2020 г. по «03»12.2021 г	https://urait.ru/
Договор № 027-1-3034-21 от 03.12.2021г ООО "Электронное издательство ЮРАЙТ" С «04»12.2021 г. по «03»12.2022 г	https://urait.ru/
Договор № 150-1-3269-21 от 10.12.21 ООО "Электронное издательство ЮРАЙТ" для СПО	https://urait.ru/
Договор № 027-1-2554-22 от 01.12.2022г ООО "Электронное издательство ЮРАЙТ" С «04»12.2022 г. по «03»12.2023 г	
Договор № 5537 от 25.11.2022 ООО "Электронное издательство ЮРАЙТ" для СПО	
Электронная библиотека МАИ	
Электронная библиотека МАИ (собственность МАИ). Лицензионный договор № 0267-НИЧ-13 от 11.12.2013 г. с ООО "Дата Экспресс "на право использования программы для ЭВМ Автоматизированная интегрированная библиотечная система (АИБС) «МегаПро» (для размещения Электронной библиотеки МАИ)	https://elibrary.mai.ru/MegaPro/Web
Электронная библиотека Консорциума аэрокосмических вузов России	
Электронная библиотека Консорциума аэрокосмических вузов России. Соглашение о создании Консорциума вузов России "Национальный объединенный аэрокосмический университет" от 03.09.2012 г. Договор о сетевом взаимодействии от 15.12.2014 г. Соглашение от «03»09.2012 г. бессрочно	

Библиотека РФФИ	
Библиотека РФФИ	http://www.rfbr.ru/rffi/ru/library
Единое окно доступа к образовательным ресурсам	
Единое окно доступа к образовательным ресурсам	http://window.edu.ru/
Polpred.com	
Polpred.com. Обзор СМИ	http://polpred.com
ООО "РУНЭБ"	
Договор № 027-1-3051-20 от 07.12.2020 с ООО "РУНЭБ" С «07»12.2020 г. по «06»12.2028	http://elibrary.ru
Договор № 027-1-2895-21 от 03.12.2021 с ООО "РУНЭБ" С «03»12.2021 г. по «02»12.2039	
Договор № 027-133215-22 от 20.12.2022 с ООО "НЭБ" С «20»12.2022 г. по «19»12.2030	
ООО "Национальный цифровой ресурс "Рукоонт"	
Договор № РКТ-054/20/027-1-1129-20 от 30.05.2020 с ООО "Национальный цифровой ресурс "Рукоонт" С «01»06.2020 г. по «31»05.2021 г	http://text.rucont.ru/
Договор № 027-1-1235-21 от 01.06.2021 с ООО "Национальный цифровой ресурс "Рукоонт" С «01»06.2021 г. по «31»05.2022 г	https://text.rucont.ru/
Договор № 027-1-1467-22 от 09.06.2022 с ООО "Национальный цифровой ресурс "Рукоонт" С «01»06.2022 г. по «31»05.2023 г	https://text.rucont.ru/
ФГБУ "РГБ"	
Договор о предоставлении доступа к Национальной электронной библиотеке (НЭБ) №101/НЭБ/2139 от 13.11.2018г. с ФГБУ" РГБ" С «13»11. 2018 г. по «12» 11. 2023	http://нэб.рф
НП НЭИКОН	
Соглашение № 715 ДС-2011 от 16.05.2011 о сотрудничестве в Консорциуме НЭИКОН С «16» 05.2011 г с автоматическим продлением Национальная подписка на-2021 г с РФФИ Государственного задания № 075-00011-20-00 Web Of Science- https://apps.webofknowledge.com Scopus- http://scopus.com Elsevier- http://www.sciencedirect.com , http://www.elsevierscience.ru/products/science-direct , https://www.elsevier.com/solutions/sciencedirect/content/journal-collections , https://www.elsevier.com/solutions/sciencedirect/content/backfile-collections	http://archive.neicon.ru https://apps.webofknowledge.com http://scopus.com http://www.sciencedirect.com , http://www.elsevierscience.ru/products/science-direct , https://www.elsevier.com/solutions/sciencedirect/content/journal-collections , https://www.elsevier.com/solutions/sciencedirect/content/backfile-collections
	http://rd.springer.com , http://www.springerprotocols.com

<p>Математическая база данных zbMATH: http://zbMATH.org</p> <p>American Chemical Society (ACS)- https://www.acs.org/content/acs/en.html</p> <p>American Institute of Physics (AIP)- https://www.scitation.org/</p> <p>American Physical Society- https://journals.aps.org/about</p> <p>EBSCO Publishing (База CASC)- http://search.ebscohost.com</p> <p>Cambridge University Press (CUP)- https://www.cambridge.org/core</p> <p>IEL издательства IEEE (Institute of Electrical and Electronics Engineers , Inc.)- https://ieeexplore.ieee.org</p> <p>INSPEC компании EBSCO- INSPEC</p> <p>Institute of Physics (IOP) издательства IOP Publishing- https://iopscience.iop.org/</p> <p>MathSciNet American Mathematical Society- https://www.ams.org/home/page</p> <p>Optical Society of America (OSA)- https://www.osapublishing.org/about.cfm</p> <p>Oxford University Press- https://academic.oup.com/journals/</p> <p>ProQuest Dissertations & Theses Global- https://search.proquest.com/index</p> <p>ORBIT Intelligence - база данных QUESTEL- https://www.orbit.com/</p> <p>SAGE Publication- https://journals.sagepub.com/</p> <p>Annual Reviews Science Collection (AR)- https://www.annualreviews.org</p> <p>JSTOR- www.jstor.org</p> <p>Wiley. John Wiley & Sons.- https://onlinelibrary.wiley.com/</p> <p>Национальная подписка на 2022 г с РФФИ Государственного задания</p>	<p>http://zbMATH.org</p> <p>https://www.acs.org/content/acs/en.html</p> <p>https://www.scitation.org/</p> <p>https://journals.aps.org/about</p> <p>http://search.ebscohost.com</p> <p>https://www.cambridge.org/core</p> <p>https://ieeexplore.ieee.org</p> <p>https://iopscience.iop.org/</p> <p>https://www.ams.org/home/page</p> <p>https://www.osapublishing.org/about.cfm</p> <p>https://academic.oup.com/journals/</p> <p>https://search.proquest.com/index</p> <p>https://www.orbit.com/</p> <p>https://journals.sagepub.com/</p> <p>https://www.annualreviews.org</p> <p>www.jstor.org</p> <p>https://onlinelibrary.wiley.com</p>
<p>Springer Nature:</p> <p>1. eBoock Collection: журналы, книги - https://link.springer.com</p> <p>2. Коллекция журналов и базы данных Springer Nature: https://link.springer.com</p> <p>Begell House Inc. https://www.dl.begellhouse.com/collections/6764f0021c05bd10.html</p> <p>China Academic Journals (CD Edition) Electronic Publishing House Co., Ltd: https://ar.cnki.net/ACADREF</p> <p>Institute of Electrical and Electronics Engineers:</p>	<p>https://link.springer.com</p> <p>https://www.dl.begellhouse.com/collections/6764f0021c05bd10.html</p> <p>https://ar.cnki.net/ACADREF</p> <p>https://ieeexplore.ieee.org/Xplore/</p>
<p>https://ieeexplore.ieee.org/Xplore/home.jsp; https://ieeexplore.ieee.org</p>	<p>home.jsp; https://ieeexplore.ieee.org</p>

EBSCO.	https://www.search.ebscohost.com/	https://www.search.ebscohost.com/
INSPEC:		
1. База данных Academic Search Premier		
2. База данных eBook Academic Collection		
3. eBook EngineeringCore Collection		
ORBIT Intelligence	- база данных QUESTEL:	https://www.orbit.com/
https://www.orbit.com/		
SAGE	https://journals.sagepub.com/	https://journals.sagepub.com/
Publication:		
Wiley:	https://onlinelibrary.wiley.com/	https://onlinelibrary.wiley.com/

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Эффективным способом развития творческих способностей студентов при изучении дисциплины является самостоятельная работа, которая нацелена на проработку студентами материала прошедших контактных занятий и подготовку к предстоящим занятиям.

Самостоятельная работа студентов проводится ими в соответствии с собственными возможностями. Можно, однако, рекомендовать групповое изучение материалов, обеспечивающее совместную работу нескольких студентов, что положительно влияет на качество проработки программы курса.

В то же время высокая степень усвоения изучаемой дисциплины достигается при постоянной работе студентов над текущим материалом. В этой связи желательна проработка лекционного материала в день его прочтения, что позволяет, во-первых, оперативно (на следующей лекции) снимать возникающие вопросы и, во-вторых, создавать багаж знаний по дисциплине задолго до промежуточной аттестации.

При подготовке к практическим занятиям также необходима проработка лекционного материала. Это позволит осознанно работать с предлагаемым материалом преподавателем на практическом занятии, а, следовательно, закладывать базу методик и приемов при решении практических задач.

При изучении материала необходимо делать акцент не на зазубривании материала, а на понимании его физической сути, что развивает мышление и позволяет понять методологию изучаемой дисциплины.

9. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Дисциплина ориентирована на применение компьютерной техники, ресурсов информационно-телекоммуникационной сети "Интернет", электронной библиотеки МАИ для поиска, сбора, хранения, обработки и представления информации.

Программное обеспечение, Интернет-ресурсы, электронные библиотечные системы:

1. Операционная система Linux.
2. Транслятор с языка программирования с++ (g++).
3. Транслятор с языка программирования лисп (GNU Clisp).
4. Транслятор с языка программирования пролог (GNU Prolog).
5. Транслятор языка программирования постскрипт (Ghostscript).
6. Транслятор с языка программирования рубин (Ruby).
7. Отладчик (GNU gdb или Microsoft debug).
8. www.fepo.ru, ru.wikipedia.org, сайт кафедры МСиИТ.

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

1. Лекционные занятия

- 1.1. Комплект электронных презентаций/слайдов.
- 1.2. Аудитория, оснащенная презентационной техникой (проектор, экран, компьютер)

2. Лабораторные работы

- 2.1. Аудитория, оснащенная презентационной техникой (проектор, экран, компьютер).
- 2.2. Персональные компьютеры – 27 рабочих мест.
- 2.3. Локальная вычислительная сеть доступом в Интернет.

Аннотация рабочей программы

Дисциплина "Программирование" является частью "Блока 1 Дисциплины" дисциплин подготовки студентов по направлению подготовки 09.03.01 "Информатика и вычислительная техника". Дисциплина реализуется на "Московского авиационного института (национального исследовательского университета)" кафедрой (кафедрами) .

Дисциплина нацелена на формирование следующих компетенций: ОПК-4, ОПК-10, ОПК-11.

Содержание дисциплины охватывает круг вопросов, связанных с: базовых теоретических положений использования языков программирования и практических методов использования этих положений, достаточно широкой подготовкой в области использования различных языков программирования, позволяющей в дальнейшем осуществить специализацию по выбранному профилю, формированием у студентов научного мышления, правильного понимания оптимальности применимости различных языков программирования для решения различных задач, усвоением основных понятий языков программирования, а также овладение основными методами математического моделирования, широко применяемыми в современной технике, выработкой у студентов владения приемами и навыками решения конкретных задач из разных областей программирования, помогающих в дальнейшем в решении инженерных задач по выбранной специальности, способностью представить адекватную современному уровню знаний научную картину мира на основе знания основных положений, законов и методов естественных наук, математики и программирования, умением использовать математический аппарат и численные методы для моделирования физико-химических процессов и явлений, лежащих в основе нанотехнологий, на вычислительной технике

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: Лекция, Лабораторная работа.

Программой дисциплины предусмотрены следующие виды контроля: промежуточная аттестация в форме Зачет (3 семестр), Экзамен (4 семестр).

Общая трудоемкость освоения дисциплины составляет 8 зачетных единиц, 288 часов. Программой дисциплины предусмотрены лекционные (50 часов), лабораторные (80 часов) занятия и (122 часов) самостоятельной работы студента.

Прикрепленные файлы

варианты кр.pdf

вопросы к экзамену.pdf

Экзамен (4 семестр).pdf

вопросы к зачету.pdf

Зачет (3 семестр).pdf

Курсовая работа по теме “Изучение характеристик алгоритмов сортировки данных” имеет 15 вариантов заданий, отличающихся друг от друга исследуемыми алгоритмами сортировки:
Варианты

1. Метод пузырька / Сортировка выбором
2. Метод пузырька / Метод Шелла
3. Метод пузырька / Сортировка бин. деревом
4. Метод пузырька / Сортировка массивом
5. Метод пузырька / Быстрая сортировка
6. Сортировка выбором / Метод Шелла
7. Сортировка выбором / Сортировка бин. деревом
8. Сортировка выбором / Сортировка массивом
9. Сортировка выбором / Быстрая сортировка
10. Метод Шелла / Сортировка бин. деревом
11. Метод Шелла / Сортировка массивом
12. Метод Шелла / Быстрая сортировка
13. Сортировка бин. деревом / Сортировка массивом
14. Сортировка бин. деревом / Быстрая сортировка
15. Сортировка массивом / Быстрая сортировка

Раздел 1. Введение в теорию алгоритмических языков

1. Алгоритм.
2. Основные способы записи алгоритмов.
3. Блок-схема.
4. Программа на АЯ или ЯП.
5. Языки программирования низкого уровня: машинный код, мнемокод, ассемблер. Исторический обзор языков программирования высокого уровня:
6. Общее определение языка программирования.
7. Модель передачи сообщения: семантика, синтактика, прагматика.
8. Классификация ошибок в программе.
9. Текст программы, алфавит языка, лексемы.
10. Типы лексем.
11. Комментарии.
12. Программа = Алгоритм + Структуры данных.
13. Определение понятия тип данных.
14. Переменные.

Раздел 2. Обзор языков программирования

1. Фортран.
2. Алгол-60/68.
3. ПЛ/1
4. Лисп.
5. Бэйсик.
6. Си/си++.
7. Паскаль
8. Оккам-2.
9. Форт.
10. Пролог.
11. Ада.
12. SQL.
13. Ява.
14. Яваскрипт.
15. Перл.
16. Питон.
17. Рубин.
18. Tcl/Tk.
19. Хаскель.
20. Луа.
21. Сравнительный анализ языков программирования бэйсика, паскаля и си++.

Раздел 3. Язык программирования паскаль

1. Иерархия типов данных и базовые операции в паскале.
2. Синтаксические диаграммы.
3. Алфавит, лексика, организация программы.
4. Система типов языка.
5. Скалярные типы.

6. Описание переменных, констант и типов.
7. Преобразование и совместимость типов.
8. Регулярные типы.
9. Комбинированные типы.
10. Строки.
11. Множества.
12. Операторы.
13. Процедуры и функции.
14. Способы передачи параметров.
15. Динамические массивы.
16. Рекурсия: прямая и косвенная.
17. Внешние подпрограммы.
18. Понятие об объектах.
19. Типы указателей.
20. Видимость имен.
21. Динамические, временные и статические данные.
22. Выражения.
23. Файловые типы и переменные.
24. Текстовые файлы.
25. Ввод-вывод.

Раздел 4. Программирование на паскале.

1. Описать алгоритм заполнения массива.
2. Описать алгоритм расчета среднего арифметического.
3. Описать алгоритм расчета среднего геометрического.
4. Описать алгоритм расчета среднего гармонического.
5. Описать алгоритм расчета среднего квадратического.
6. Каким способом лучше передавать параметр-массив в функции для расчета средних?
7. Каковы получаемые при прогонах программы отношения между средними значениями?
8. В каком случае лучше задавать диапазон индекса массива литералами, а в каком константами?
9. Как превратить цикл for в цикл while или repeat? Когда вместо цикла for предпочтительнее использовать цикл while или repeat?
10. Как в программе задать множество чисел для расчета средних?
11. Почему среднее геометрическое нельзя считать непосредственно по формуле-определению?
12. Описать алгоритм проверки числа на простоту.
13. Каким способом лучше передавать параметр-число в функцию для проверки на простоту?
14. Почему простой рекурсивный алгоритм для вычисления чисел Фибоначчи так неэффективен?
15. Каковы отношения между полученными средними значениями?
16. Можно ли утверждать, что нерекурсивный алгоритм для вычисления факториала будет намного быстрее рекурсивного?
17. Можно ли заменять оператор варианта на несколько условных операторов?
18. Как должно выглядеть определение функции, эквивалентной odd?
19. Сколько раз нужно использовать процедуру randomize в программе?
20. Как может выглядеть определение функции для быстрого расчета чисел Фибоначчи?
21. Каково максимальное число Фибоначчи и максимальный факториал, которые можно вычислить на компьютере при использовании типа longint?
22. Можно ли использовать операцию сдвига вместо деления на 2?
23. Сравнить способы инициализации данных на паскале и бейсике.

Варианты заданий. Каждый вариант содержит три задачи.

Задача 1.

Выписать идентификаторы языка программирования паскаль.

Задача 2.

Разобрать ход исполнения программы.

Задача 3.

Найти число элементов в заданном массиве.

<p style="text-align: center;"><i>Вариант 1</i></p> <pre> 1) _1234A, ELSE, A_B, 10_10 2) var i,j:integer; begin i:=17; i:=(5+i*7 div 5 +1) shl 1; if (i>5)or(i=7) then j:=\$e else j:=i+2; i:=1+j; writeln(i and 7) end. 3) array[false..true]of array[-1..11]of byte </pre>	<p style="text-align: center;"><i>Вариант 2</i></p> <pre> 1) if, ____, 56AB, myid, MS-DOS; 2) var i,j:integer; begin i:=14; i:=(5+i*7 div 6 +1) shl 1; if (i>5)or(i=7) then j:=\$d else j:=i+2; i:=2+j; writeln(i or 7) end. 3) array[0..4,'D'..'F']of array[-1..8]of byte </pre>
<p style="text-align: center;"><i>Вариант 3</i></p> <pre> 1) TestVal, then, IfElse, _44; 2) var i,j:integer; begin i:=7; i:=((5+i)*i div 4 +1) mod 5; if (i>5)or(i=7) then j:=\$c else j:=i+3; i:=74 mod j; writeln(i or 8) end. 3) array[-2..3,'F'..'I']of array[-1..9]of byte </pre>	<p style="text-align: center;"><i>Вариант 4</i></p> <pre> 1) case, Zero, A555 2) var i,j:integer; begin i:=6; i:=((5+i)*i div 4 +1) mod 6; if (i>5)or(i=7) then j:=\$c else j:=i+3; i:=14 mod (j+1); writeln(i or 5) end. 3) array[0..4,5..5]of array['X'..'Z']of byte </pre>
<p style="text-align: center;"><i>Вариант 5</i></p> <pre> 1) _744B, C,D, THEN, __1; 2) var i,j:integer; begin i:=7; i:=(i+11*i div 5 +1) shl 1; if (i>5)or(i=7) then j:=\$c else j:=i+2; i:=j div 3; writeln(i or 9) end. 3) array['Y'..'Z']of array[false..true]of byte </pre>	<p style="text-align: center;"><i>Вариант 6</i></p> <pre> 1) record, FOR, _6_, AB\$; 2) var i,j:integer; begin i:=7; i:=(i+11*i div 5 +1) shl 1; if (i>5)or(i=8) then j:=\$f else j:=i+2; i:=j div 2; writeln(i or 11) end. 3) array[0..4,'V'..'Y']of array[-4..4]of byte </pre>
<p style="text-align: center;"><i>Вариант 7</i></p> <pre> 1) \$55, 45, SelectCase, 'ABC'; 2) var i,j:integer; begin i:=47; i:=((5+i)*11 div 7 +1) shr 2; if (i>5)or(i=7) then j:=\$c else j:=i*\$e; i:=j mod 11; writeln(i xor 12) end. 3) array[2..5]of array['Q'..'S',-2..9]of byte </pre>	<p style="text-align: center;"><i>Вариант 8</i></p> <pre> 1) Main&Var, test(), long_identifier, ABC; 2) var i,j:integer; begin i:=17; i:=((25+i)*4 div 5 +1) shr 2; if (i>5)or(i=7) then j:=12 else j:=i*\$e; i:=j mod 11; writeln(i xor 8) end. 3) array[-2..4]of array['S'..'W',2..9]of byte </pre>
<p style="text-align: center;"><i>Вариант 9</i></p> <pre> 1) do, _AZ.09_, 55.88, "Var"; 2) var i,j:integer; begin i:=14; i:=((i-2)*7 div 2 +1) mod 11; if (i>5)or not (i=7) then j:=\$e else j:=i+2; i:=(j+1)*i; writeln(i and 14) end. 3) array[2..7]of array[-5..5,2..9]of byte </pre>	<p style="text-align: center;"><i>Вариант 10</i></p> <pre> 1) 'This_is_a_variable', truth, AB.i77, const; 2) var i,j:integer; begin i:=12; i:=((i-1)*7 div 2 +1) mod 11; if (i>5)or not (i=7) then j:=\$10 else j:=i+2; i:=j*(i+1); writeln(i and 13) end. 3) array[2..5]of array[11..18,-2..9]of byte </pre>

<p style="text-align: center;"><i>Вариант 11</i></p> <pre> 1) A\$, ID, BEGIN, _0000; 2) var i,j:integer; begin i:=15; i:=(5+i div 7 -4) mod 2; if (i>5) xor (i<7) then j:=12 else j:=i*\$12; i:=j shl 1; writeln(i xor 11) end. 3) array[false..true] of array[7..12] of byte </pre>	<p style="text-align: center;"><i>Вариант 12</i></p> <pre> 1) TestVar, repeat, case, mod; 2) var i,j:integer; begin i:=16; i:=(5+i div 7 -5) mod 3; if (i>5) xor (i<9) then j:=12 else j:=i*\$12; i:=j shl 1; writeln(i xor 5) end. 3) array[2..5] of array[\$10..\$20,-2..9] of byte </pre>
<p style="text-align: center;"><i>Вариант 13</i></p> <pre> 1) \$AA, \$55, 0x66, ERROR; 2) var i,j:integer; begin i:=12; i:=((i+i)*i shl 4 +1) mod 2; if (i<=5) xor (i=7) then j:=\$c else j:=i*2; i:=j+4; writeln(i or 14); end. 3) array[2..5] of array[\$12..\$1d,-2..4] of byte </pre>	<p style="text-align: center;"><i>Вариант 14</i></p> <pre> 1) GloriousVar, Finish!, integer, int; 2) var i,j:integer; begin i:=10; i:=((i+i)*i shl 4 +1) mod 2; if (i<=5) xor (i=7) then j:=\$d else j:=i*2; i:=j+5; writeln(i xor 12); end. 3) array[2..6] of array[\$11..\$1f,-2..1] of byte </pre>
<p style="text-align: center;"><i>Вариант 15</i></p> <pre> 1) music, Basic, circle&box, DIV; 2) var i,j:integer; begin i:=7; i:=((5+i)*i div 5 +1) shl 1; if (i>7) or (i=7) then j:=\$c else j:=i*2; i:=j-4; writeln(i and 10) end. 3) array[2..15] of array[8..\$1a,-2..6] of byte </pre>	<p style="text-align: center;"><i>Вариант 16</i></p> <pre> 1) IF, beauty&beast, _ABC_, GoTo; 2) var i,j:integer; begin i:=9; i:=((3+i)*i div 5 +1) shl 1; if (i>5) or (i=7) then j:=\$c else j:=i*2; i:=j-2; writeln(i and 11) end. 3) array[2..5] of array[8..\$10,'U'..'V'] of byte </pre>
<p style="text-align: center;"><i>Вариант 17</i></p> <pre> 1) i80486dx, CPU, Money\$, A.B1; 2) var i,j:integer; begin i:=4; i:=((5+i)*i div 5 +2) shl 1; if (i<2) or (i=7) then j:=\$c else j:=i*4; i:=j-7; writeln(i xor 14) end. 3) array[2..\$a] of array['J'..'L',-2..8] of byte </pre>	<p style="text-align: center;"><i>Вариант 18</i></p> <pre> 1) aa456mi, x44..45, A.b.CD, until; 2) var i,j:integer; begin i:=5; i:=((5+i)*i div 5 +2) shl 1; if (i<2) or (i=7) then j:=\$c else j:=i*4; i:=j-7; writeln(i or 2) end. 3) array[7..\$b] of array['L'..'N',-2..0] of byte </pre>
<p style="text-align: center;"><i>Вариант 19</i></p> <pre> 1) FINISH, SUMX, 'QUESTION', WHILE; 2) var i,j:integer; begin i:=3; i:=((5-i)*i div 4 -1) shr 1; if (i>5) and (i<7) then j:=12 else j:=i+2; i:=j+5; writeln(i or 5) end. 3) array[boolean] of array[1..\$4,2..8] of byte </pre>	<p style="text-align: center;"><i>Вариант 20</i></p> <pre> 1) break, bang, BREAK, LOOK_AT_ME; 2) var i,j:integer; begin i:=5; i:=((7-i)*i div 4 -1) shr 1; if (i>5) and (i<7) then j:=12 else j:=i+2; i:=j+7; writeln(i xor 6) end. 3) array[char] of array[boolean] of byte </pre>

<p style="text-align: center;"><i>Вариант 21</i></p> <p>1) Pentium, DMA, Money, A+B1; 2) var i,j:integer; begin i:=6; i:=(5+i)*i div 7 +2) shl 1; if (i<2)or(i=7) then j:=\$c else j:=i+4; i:=j-7; writeln(i xor 12) end. 3) array[boolean,char]of array[3..4]of byte</p>	<p style="text-align: center;"><i>Вариант 22</i></p> <p>1) DownTo, MMU, OS/3, A_B1; 2) var i,j:integer; begin i:=3; i:=((7+i)*i div 5 +2) shl 1; if (i<2)or(i=8) then j:=\$c else j:=i+4; i:=j-6; writeln(i or 8) end. 3) array[byte]of array[Boolean,5..6]of byte</p>
<p style="text-align: center;"><i>Вариант 23</i></p> <p>1) i80186, Unix, Monkey, A-B1; 2) var i,j:integer; begin i:=4; i:=(5+i)*i div 5 +2) shl 1; if (i<2)or(i=7) then j:=\$c else j:=i+4; i:=j-7; writeln(i xor 14) end. 3) array[boolean,'U'..'V',char,3..4]of byte</p>	<p style="text-align: center;"><i>Вариант 24</i></p> <p>1) i8086, OS/2, With, A_B1; 2) var i,j:integer; begin i:=6; i:=((4+i)*i div 5 +2) shl 1; if (i<2)or(i=7) then j:=\$c else j:=i+4; i:=j-4; writeln(i and 14) end. 3) array[byte,boolean,'Q'..'R',3..4]of byte</p>
<p style="text-align: center;"><i>Вариант 25</i></p> <p>1) to, Macintosh, Money.Cents, AAB1A2; 2) var i,j:integer; begin i:=2; i:=(8+i)*i div 4 +2) shl 1; if (i<4)or(i=7) then j:=\$c else j:=i+4; i:=j-7; writeln(i and 15) end. 3) array[boolean,'V'..'Y',boolean,2..5]of byte</p>	<p style="text-align: center;"><i>Вариант 26</i></p> <p>1) of, ____1A, 56AB7, UCSD, Pascal; 2) var i,j:integer; begin i:=12; i:=(5+i*7 div 5 +1) shl 1; if (i>5)or(i=9) then j:=\$d else j:=i*2; i:=2+j; writeln(i xor 9) end. 3) array[byte,'R'..'W',3..4,-5..2]of byte</p>
<p style="text-align: center;"><i>Вариант 27</i></p> <p>1) caSe, 12____, 156A, Atari, C++; 2) var i,j:integer; begin i:=10; i:=(5+i*7 div 6 +1) shl 1; if (i>5)or(i=7) then j:=\$d else j:=i+2; i:=4+j; writeln(i xor 7) end. 3) array[byte,7..\$f,'S'..'V',4..7]of byte</p>	<p style="text-align: center;"><i>Вариант 28</i></p> <p>1) Mod, _ABC____, 5AB77, Sinclair, Forth; 2) var i,j:integer; begin i:=8; i:=(6+i*7 div 6 +1) shl 1; if (i>5)or(i=7) then j:=\$d else j:=i*2; i:=5+j; writeln(i xor 11) end. 3) array[0..4,'D'..'F',-1..8,\$12..\$25]of byte</p>
<p style="text-align: center;"><i>Вариант 29</i></p> <p>1) caSE, 2____A, 15A7, BBC, Java; 2) var i,j:integer; begin i:=15; i:=(5+i*4 div 6 +1) shl 1; if (i>5)or(i=6) then j:=\$d else j:=i*2; i:=4+j; writeln(i xor 7) end. 3) array[byte,9..\$f,'S'..'V',4..7]of byte</p>	<p style="text-align: center;"><i>Вариант 30</i></p> <p>1) MOD, _ABC.x., 5AB77, Spektrum, Logo; 2) var i,j:integer; begin i:=1; i:=(6+i*7 div 6 +1) shl 1; if (i>5)or(i=5) then j:=\$d else j:=i*2; i:=8+j; writeln(i xor 11) end. 3) array[0..0,'D'..'J',-1..8,\$12..\$25]of byte</p>

Раздел 5. Введение в ООП

1. Методы и свойства.
2. Публичные и приватные поля и методы.
3. Наследование.
4. Полиморфизм.

Раздел 6. Структуры данных

1. Стек.
 2. Очередь.
 3. Дек.
 4. Бинарные деревья.
 5. Обобщения бинарных деревьев.
 6. Хэш.
 7. Список однонаправленный.
 8. Список двунаправленный.
-
1. Назвать преимущества объектно-ориентированного подхода для моделирования структур данных.
 2. На основе какой структуры создаётся стек.
 3. На основе какой структуры создаётся дек.
 4. Можно ли реализовать стек (дек) на основе однонаправленного списка?
 5. Можно ли реализовать очередь на основе однонаправленного списка?
 6. Описать интерфейсный модуль для работы с однонаправленным списком.
 7. Описать интерфейсный модуль для работы с двунаправленным списком.
 8. Перечислить операции для типа бинарное дерево.

Раздел 7. Концепции ООП

1. Объектный тип.
2. Инкапсуляция данных.
3. Конструкторы.
4. Деструкторы.
5. Дружественные функции.
6. Наследование.
7. Виртуальные функции и полиморфизм.
8. Динамическое выделение и освобождение памяти для объекта.

Раздел 8. Использование графики

1. Почему указатель на следующий графический объект в объекте-круге должен быть типа указатель на базовый абстрактный объект?
2. Почему нельзя использовать указатель типа указатель на объект-круг?
3. Какой метод программы наиболее естественно сделать виртуальным?
4. Производным от какого объектного типа следовало бы сделать тип-кольцо? Как добавить в иерархию графических объектов такой тип?
5. Как добавить в список графических объектов еще один круг?

6. Почему отказ от использования наследования значительно усложнил бы программирование графических интерфейсов?
7. Необходим ли программе механизм полиморфизма концепции ООП?
8. Какие недостатки в реализации программы простого графического сервера sgserv вы обнаружили?
9. Почему базовый абстрактный тип не нуждается в конструкторе? Чем полезно введение виртуального деструктора в абстрактный тип?
10. Сколько раз будет вызван деструктор круга при работе программы?
11. Нужно ли при реализации объектной иерархии программы использовать механизм инкапсуляции ООП?
12. Алгоритмы ООП работы с графикой в клиент-серверной архитектуре.
13. Построение графиков заданных функций и других графических объектов, используя средства наследования ООП и графический сервер.
14. Пример сценария для построения графика функций.

Раздел 9. Модульное программирование

1. Назначение служебного слова `unit`.
2. Назначение служебного слова `uses`.
3. Назначение служебного слова `interface..`
4. Назначение служебного слова `implementation..`
5. Сборка модульных программ.

Раздел 10. Языки лисп, пролог, рубин и постскрипт

1. Функциональное программирование.
2. Атомы и списки.
3. Данные как программа.
4. Базовые функции лиспа.
5. Логическое программирование.
6. Отношения: факты и правила.
7. Запросы.
8. Языки сценариев.
9. Присваивание по ссылке.
10. Средства рубин для работы со структурами данных.
11. MVC. Ruby on rails.
12. Самодокументированность языка рубин.
13. Постфиксная запись.
14. Вычисление чисел Фибонначи на постскрипт.

Раздел 11. Теоретические вопросы программирования

1. Систематическое изучение языков программирования с разных позиций.
2. Технологическая позиция.
3. Семиотическая позиция.
4. Авторская позиция.
5. Математическая позиция.
6. Реализаторская позиция.
7. Основная цель программирования.
8. Главная проблема программирования.
9. Источники сложности в программировании.
10. Средства борьбы со сложностью.

11. Технологический процесс разработки программ.
12. Методы проектирования структуры программы.
13. Характеристики алгоритмов и программ.
14. Правильность. Надежность. Дружественность. Сопровождение.
15. Стил ь программирования.
16. Требования к оформлению программы.
17. Мобильность.
18. Использование процедурного, объектного, ситуационного, функционального, доказательного, логического программирования при разработке ПО/МО.
19. Категории ПО.

Темы курсовых работ:

1. Исследование алгоритмов сортировки данных: метод пузырька, сортировка выбором.
2. Исследование алгоритмов сортировки данных: метод пузырька, метод Шелла.
3. Исследование алгоритмов сортировки данных: метод пузырька, сортировка бинарным деревом.
4. Исследование алгоритмов сортировки данных: метод пузырька, сортировка массивом.
5. Исследование алгоритмов сортировки данных: метод пузырька, быстрая сортировка.
6. Исследование алгоритмов сортировки данных: сортировка выбором, метод Шелла.
7. Исследование алгоритмов сортировки данных: сортировка выбором, сортировка бинарным деревом.
8. Исследование алгоритмов сортировки данных: сортировка выбором, сортировка массивом.
9. Исследование алгоритмов сортировки данных: сортировка выбором, быстрая сортировка.
10. Исследование алгоритмов сортировки данных: метод Шелла сортировка, бинарным деревом.
11. Исследование алгоритмов сортировки данных: метод Шелла, сортировка массивом.
12. Исследование алгоритмов сортировки данных: метод Шелла, быстрая сортировка.
13. Исследование алгоритмов сортировки данных: сортировка бин. деревом, сортировка массивом.
14. Исследование алгоритмов сортировки данных: сортировка бин. деревом, быстрая сортировка.
15. Исследование алгоритмов сортировки данных: сортировка массивом, быстрая сортировка.

Раздел 1. Введение в теорию алгоритмических языков

1. Алгоритм.
2. Основные способы записи алгоритмов.
3. Блок-схема.
4. Программа на АЯ или ЯП.
5. Языки программирования низкого уровня: машинный код, мнемокод, ассемблер. Исторический обзор языков программирования высокого уровня:
6. Общее определение языка программирования.
7. Модель передачи сообщения: семантика, синтактика, прагматика.
8. Классификация ошибок в программе.
9. Текст программы, алфавит языка, лексемы.
10. Типы лексем.
11. Комментарии.
12. Программа = Алгоритм + Структуры данных.
13. Определение понятия тип данных.
14. Переменные.

Раздел 2. Обзор языков программирования

1. Фортран.
2. Алгол-60/68.
3. ПЛ/1
4. Лисп.
5. Бэйсик.
6. Си/си++.
7. Паскаль
8. Оккам-2.
9. Форт.
10. Пролог.
11. Ада.
12. SQL.
13. Ява.
14. Яваскрипт.
15. Перл.
16. Питон.
17. Рубин.
18. Tcl/Tk.
19. Хаскель.
20. Луа.
21. Сравнительный анализ языков программирования бэйсика, паскаля и си++.

Раздел 3. Язык программирования паскаль

1. Иерархия типов данных и базовые операции в паскале.
2. Синтаксические диаграммы.
3. Алфавит, лексика, организация программы.
4. Система типов языка.
5. Скалярные типы.

6. Описание переменных, констант и типов.
7. Преобразование и совместимость типов.
8. Регулярные типы.
9. Комбинированные типы.
10. Строки.
11. Множества.
12. Операторы.
13. Процедуры и функции.
14. Способы передачи параметров.
15. Динамические массивы.
16. Рекурсия: прямая и косвенная.
17. Внешние подпрограммы.
18. Понятие об объектах.
19. Типы указателей.
20. Видимость имен.
21. Динамические, временные и статические данные.
22. Выражения.
23. Файловые типы и переменные.
24. Текстовые файлы.
25. Ввод-вывод.

Раздел 4. Программирование на паскале.

1. Описать алгоритм заполнения массива.
2. Описать алгоритм расчета среднего арифметического.
3. Описать алгоритм расчета среднего геометрического.
4. Описать алгоритм расчета среднего гармонического.
5. Описать алгоритм расчета среднего квадратического.
6. Каким способом лучше передавать параметр-массив в функции для расчета средних?
7. Каковы получаемые при прогонах программы отношения между средними значениями?
8. В каком случае лучше задавать диапазон индекса массива литералами, а в каком константами?
9. Как превратить цикл for в цикл while или repeat? Когда вместо цикла for предпочтительнее использовать цикл while или repeat?
10. Как в программе задать множество чисел для расчета средних?
11. Почему среднее геометрическое нельзя считать непосредственно по формуле-определению?
12. Описать алгоритм проверки числа на простоту.
13. Каким способом лучше передавать параметр-число в функцию для проверки на простоту?
14. Почему простой рекурсивный алгоритм для вычисления чисел Фибоначчи так неэффективен?
15. Каковы отношения между полученными средними значениями?
16. Можно ли утверждать, что нерекурсивный алгоритм для вычисления факториала будет намного быстрее рекурсивного?
17. Можно ли заменять оператор варианта на несколько условных операторов?
18. Как должно выглядеть определение функции, эквивалентной odd?
19. Сколько раз нужно использовать процедуру randomize в программе?
20. Как может выглядеть определение функции для быстрого расчета чисел Фибоначчи?
21. Каково максимальное число Фибоначчи и максимальный факториал, которые можно вычислить на компьютере при использовании типа longint?
22. Можно ли использовать операцию сдвига вместо деления на 2?
23. Сравнить способы инициализации данных на паскале и бейсике.

Варианты заданий. Каждый вариант содержит три задачи.

Задача 1.

Выписать идентификаторы языка программирования паскаль.

Задача 2.

Разобрать ход исполнения программы.

Задача 3.

Найти число элементов в заданном массиве.

<p style="text-align: center;"><i>Вариант 1</i></p> <pre> 1) _1234A, ELSE, A_B, 10_10 2) var i,j:integer; begin i:=17; i:=(5+i*7 div 5 +1) shl 1; if (i>5)or(i=7) then j:=\$e else j:=i+2; i:=1+j; writeln(i and 7) end. 3) array[false..true]of array[-1..11]of byte </pre>	<p style="text-align: center;"><i>Вариант 2</i></p> <pre> 1) if, ____, 56AB, myid, MS-DOS; 2) var i,j:integer; begin i:=14; i:=(5+i*7 div 6 +1) shl 1; if (i>5)or(i=7) then j:=\$d else j:=i+2; i:=2+j; writeln(i or 7) end. 3) array[0..4,'D'..'F']of array[-1..8]of byte </pre>
<p style="text-align: center;"><i>Вариант 3</i></p> <pre> 1) TestVal, then, IfElse, _44; 2) var i,j:integer; begin i:=7; i:=((5+i)*i div 4 +1) mod 5; if (i>5)or(i=7) then j:=\$c else j:=i+3; i:=74 mod j; writeln(i or 8) end. 3) array[-2..3,'F'..'I']of array[-1..9]of byte </pre>	<p style="text-align: center;"><i>Вариант 4</i></p> <pre> 1) case, Zero, A555 2) var i,j:integer; begin i:=6; i:=((5+i)*i div 4 +1) mod 6; if (i>5)or(i=7) then j:=\$c else j:=i+3; i:=14 mod (j+1); writeln(i or 5) end. 3) array[0..4,5..5]of array['X'..'Z']of byte </pre>
<p style="text-align: center;"><i>Вариант 5</i></p> <pre> 1) _744B, C,D, THEN, __1; 2) var i,j:integer; begin i:=7; i:=(i+11*i div 5 +1) shl 1; if (i>5)or(i=7) then j:=\$c else j:=i+2; i:=j div 3; writeln(i or 9) end. 3) array['Y'..'Z']of array[false..true]of byte </pre>	<p style="text-align: center;"><i>Вариант 6</i></p> <pre> 1) record, FOR, _6_, AB\$; 2) var i,j:integer; begin i:=7; i:=(i+11*i div 5 +1) shl 1; if (i>5)or(i=8) then j:=\$f else j:=i+2; i:=j div 2; writeln(i or 11) end. 3) array[0..4,'V'..'Y']of array[-4..4]of byte </pre>
<p style="text-align: center;"><i>Вариант 7</i></p> <pre> 1) \$55, 45, SelectCase, 'ABC'; 2) var i,j:integer; begin i:=47; i:=((5+i)*11 div 7 +1) shr 2; if (i>5)or(i=7) then j:=\$c else j:=i*\$e; i:=j mod 11; writeln(i xor 12) end. 3) array[2..5]of array['Q'..'S',-2..9]of byte </pre>	<p style="text-align: center;"><i>Вариант 8</i></p> <pre> 1) Main&Var, test(), long_identifier, ABC; 2) var i,j:integer; begin i:=17; i:=((25+i)*4 div 5 +1) shr 2; if (i>5)or(i=7) then j:=12 else j:=i*\$e; i:=j mod 11; writeln(i xor 8) end. 3) array[-2..4]of array['S'..'W',2..9]of byte </pre>
<p style="text-align: center;"><i>Вариант 9</i></p> <pre> 1) do, _AZ.09_, 55.88, "Var"; 2) var i,j:integer; begin i:=14; i:=((i-2)*7 div 2 +1) mod 11; if (i>5)or not (i=7) then j:=\$e else j:=i+2; i:=(j+1)*i; writeln(i and 14) end. 3) array[2..7]of array[-5..5,2..9]of byte </pre>	<p style="text-align: center;"><i>Вариант 10</i></p> <pre> 1) 'This_is_a_variable', truth, AB.i77, const; 2) var i,j:integer; begin i:=12; i:=((i-1)*7 div 2 +1) mod 11; if (i>5)or not (i=7) then j:=\$10 else j:=i+2; i:=j*(i+1); writeln(i and 13) end. 3) array[2..5]of array[11..18,-2..9]of byte </pre>

<p style="text-align: center;"><i>Вариант 11</i></p> <pre> 1) A\$, ID, BEGIN, _0000; 2) var i,j:integer; begin i:=15; i:=(5+i div 7 -4) mod 2; if (i>5) xor (i<7) then j:=12 else j:=i*\$12; i:=j shl 1; writeln(i xor 11) end. 3) array[false..true] of array[7..12] of byte </pre>	<p style="text-align: center;"><i>Вариант 12</i></p> <pre> 1) TestVar, repeat, case, mod; 2) var i,j:integer; begin i:=16; i:=(5+i div 7 -5) mod 3; if (i>5) xor (i<9) then j:=12 else j:=i*\$12; i:=j shl 1; writeln(i xor 5) end. 3) array[2..5] of array[\$10..\$20,-2..9] of byte </pre>
<p style="text-align: center;"><i>Вариант 13</i></p> <pre> 1) \$AA, \$55, 0x66, ERROR; 2) var i,j:integer; begin i:=12; i:=((i+i)*i shl 4 +1) mod 2; if (i<=5) xor (i=7) then j:=\$c else j:=i*2; i:=j+4; writeln(i or 14); end. 3) array[2..5] of array[\$12..\$1d,-2..4] of byte </pre>	<p style="text-align: center;"><i>Вариант 14</i></p> <pre> 1) GloriousVar, Finish!, integer, int; 2) var i,j:integer; begin i:=10; i:=((i+i)*i shl 4 +1) mod 2; if (i<=5) xor (i=7) then j:=\$d else j:=i*2; i:=j+5; writeln(i xor 12); end. 3) array[2..6] of array[\$11..\$1f,-2..1] of byte </pre>
<p style="text-align: center;"><i>Вариант 15</i></p> <pre> 1) music, Basic, circle&box, DIV; 2) var i,j:integer; begin i:=7; i:=((5+i)*i div 5 +1) shl 1; if (i>7) or (i=7) then j:=\$c else j:=i*2; i:=j-4; writeln(i and 10) end. 3) array[2..15] of array[8..\$1a,-2..6] of byte </pre>	<p style="text-align: center;"><i>Вариант 16</i></p> <pre> 1) IF, beauty&beast, _ABC_, GoTo; 2) var i,j:integer; begin i:=9; i:=((3+i)*i div 5 +1) shl 1; if (i>5) or (i=7) then j:=\$c else j:=i*2; i:=j-2; writeln(i and 11) end. 3) array[2..5] of array[8..\$10,'U'..'V'] of byte </pre>
<p style="text-align: center;"><i>Вариант 17</i></p> <pre> 1) i80486dx, CPU, Money\$, A.B1; 2) var i,j:integer; begin i:=4; i:=((5+i)*i div 5 +2) shl 1; if (i<2) or (i=7) then j:=\$c else j:=i*4; i:=j-7; writeln(i xor 14) end. 3) array[2..\$a] of array['J'..'L',-2..8] of byte </pre>	<p style="text-align: center;"><i>Вариант 18</i></p> <pre> 1) aa456mi, x44..45, A.b.CD, until; 2) var i,j:integer; begin i:=5; i:=((5+i)*i div 5 +2) shl 1; if (i<2) or (i=7) then j:=\$c else j:=i*4; i:=j-7; writeln(i or 2) end. 3) array[7..\$b] of array['L'..'N',-2..0] of byte </pre>
<p style="text-align: center;"><i>Вариант 19</i></p> <pre> 1) FINISH, SUMX, 'QUESTION', WHILE; 2) var i,j:integer; begin i:=3; i:=((5-i)*i div 4 -1) shr 1; if (i>5) and (i<7) then j:=12 else j:=i+2; i:=j+5; writeln(i or 5) end. 3) array[boolean] of array[1..\$4,2..8] of byte </pre>	<p style="text-align: center;"><i>Вариант 20</i></p> <pre> 1) break, bang, BREAK, LOOK_AT_ME; 2) var i,j:integer; begin i:=5; i:=((7-i)*i div 4 -1) shr 1; if (i>5) and (i<7) then j:=12 else j:=i+2; i:=j+7; writeln(i xor 6) end. 3) array[char] of array[boolean] of byte </pre>

<p style="text-align: center;"><i>Вариант 21</i></p> <p>1) Pentium, DMA, Money, A+B1; 2) var i,j:integer; begin i:=6; i:=(5+i)*i div 7 +2) shl 1; if (i<2)or(i=7) then j:=\$c else j:=i+4; i:=j-7; writeln(i xor 12) end. 3) array[boolean,char]of array[3..4]of byte</p>	<p style="text-align: center;"><i>Вариант 22</i></p> <p>1) DownTo, MMU, OS/3, A_B1; 2) var i,j:integer; begin i:=3; i:=((7+i)*i div 5 +2) shl 1; if (i<2)or(i=8) then j:=\$c else j:=i+4; i:=j-6; writeln(i or 8) end. 3) array[byte]of array[Boolean,5..6]of byte</p>
<p style="text-align: center;"><i>Вариант 23</i></p> <p>1) i80186, Unix, Monkey, A-B1; 2) var i,j:integer; begin i:=4; i:=(5+i)*i div 5 +2) shl 1; if (i<2)or(i=7) then j:=\$c else j:=i+4; i:=j-7; writeln(i xor 14) end. 3) array[boolean,'U'..'V',char,3..4]of byte</p>	<p style="text-align: center;"><i>Вариант 24</i></p> <p>1) i8086, OS/2, With, A_B1; 2) var i,j:integer; begin i:=6; i:=((4+i)*i div 5 +2) shl 1; if (i<2)or(i=7) then j:=\$c else j:=i+4; i:=j-4; writeln(i and 14) end. 3) array[byte,boolean,'Q'..'R',3..4]of byte</p>
<p style="text-align: center;"><i>Вариант 25</i></p> <p>1) to, Macintosh, Money.Cents, AAB1A2; 2) var i,j:integer; begin i:=2; i:=(8+i)*i div 4 +2) shl 1; if (i<4)or(i=7) then j:=\$c else j:=i+4; i:=j-7; writeln(i and 15) end. 3) array[boolean,'V'..'Y',boolean,2..5]of byte</p>	<p style="text-align: center;"><i>Вариант 26</i></p> <p>1) of, ____1A, 56AB7, UCSD, Pascal; 2) var i,j:integer; begin i:=12; i:=(5+i*7 div 5 +1) shl 1; if (i>5)or(i=9) then j:=\$d else j:=i*2; i:=2+j; writeln(i xor 9) end. 3) array[byte,'R'..'W',3..4,-5..2]of byte</p>
<p style="text-align: center;"><i>Вариант 27</i></p> <p>1) caSe, 12____, 156A, Atari, C++; 2) var i,j:integer; begin i:=10; i:=(5+i*7 div 6 +1) shl 1; if (i>5)or(i=7) then j:=\$d else j:=i+2; i:=4+j; writeln(i xor 7) end. 3) array[byte,7..\$f,'S'..'V',4..7]of byte</p>	<p style="text-align: center;"><i>Вариант 28</i></p> <p>1) Mod, _ABC____, 5AB77, Sinclair, Forth; 2) var i,j:integer; begin i:=8; i:=(6+i*7 div 6 +1) shl 1; if (i>5)or(i=7) then j:=\$d else j:=i*2; i:=5+j; writeln(i xor 11) end. 3) array[0..4,'D'..'F',-1..8,\$12..\$25]of byte</p>
<p style="text-align: center;"><i>Вариант 29</i></p> <p>1) caSE, 2____A, 15A7, BBC, Java; 2) var i,j:integer; begin i:=15; i:=(5+i*4 div 6 +1) shl 1; if (i>5)or(i=6) then j:=\$d else j:=i*2; i:=4+j; writeln(i xor 7) end. 3) array[byte,9..\$f,'S'..'V',4..7]of byte</p>	<p style="text-align: center;"><i>Вариант 30</i></p> <p>1) MOD, _ABC.x., 5AB77, Spektrum, Logo; 2) var i,j:integer; begin i:=1; i:=(6+i*7 div 6 +1) shl 1; if (i>5)or(i=5) then j:=\$d else j:=i*2; i:=8+j; writeln(i xor 11) end. 3) array[0..0,'D'..'J',-1..8,\$12..\$25]of byte</p>

Раздел 5. Введение в ООП

1. Методы и свойства.
2. Публичные и приватные поля и методы.
3. Наследование.
4. Полиморфизм.

Раздел 6. Структуры данных

1. Стек.
 2. Очередь.
 3. Дек.
 4. Бинарные деревья.
 5. Обобщения бинарных деревьев.
 6. Хэш.
 7. Список однонаправленный.
 8. Список двунаправленный.
-
1. Назвать преимущества объектно-ориентированного подхода для моделирования структур данных.
 2. На основе какой структуры создаётся стек.
 3. На основе какой структуры создаётся дек.
 4. Можно ли реализовать стек (дек) на основе однонаправленного списка?
 5. Можно ли реализовать очередь на основе однонаправленного списка?
 6. Описать интерфейсный модуль для работы с однонаправленным списком.
 7. Описать интерфейсный модуль для работы с двунаправленным списком.
 8. Перечислить операции для типа бинарное дерево.

Раздел 7. Концепции ООП

1. Объектный тип.
2. Инкапсуляция данных.
3. Конструкторы.
4. Деструкторы.
5. Дружественные функции.
6. Наследование.
7. Виртуальные функции и полиморфизм.
8. Динамическое выделение и освобождение памяти для объекта.

Раздел 8. Использование графики

1. Почему указатель на следующий графический объект в объекте-круге должен быть типа указатель на базовый абстрактный объект?
2. Почему нельзя использовать указатель типа указатель на объект-круг?
3. Какой метод программы наиболее естественно сделать виртуальным?
4. Производным от какого объектного типа следовало бы сделать тип-кольцо? Как добавить в иерархию графических объектов такой тип?
5. Как добавить в список графических объектов еще один круг?

6. Почему отказ от использования наследования значительно усложнил бы программирование графических интерфейсов?
7. Необходим ли программе механизм полиморфизма концепции ООП?
8. Какие недостатки в реализации программы простого графического сервера sgserv вы обнаружили?
9. Почему базовый абстрактный тип не нуждается в конструкторе? Чем полезно введение виртуального деструктора в абстрактный тип?
10. Сколько раз будет вызван деструктор круга при работе программы?
11. Нужно ли при реализации объектной иерархии программы использовать механизм инкапсуляции ООП?
12. Алгоритмы ООП работы с графикой в клиент-серверной архитектуре.
13. Построение графиков заданных функций и других графических объектов, используя средства наследования ООП и графический сервер.
14. Пример сценария для построения графика функций.

Раздел 9. Модульное программирование

1. Назначение служебного слова `unit`.
2. Назначение служебного слова `uses`.
3. Назначение служебного слова `interface..`
4. Назначение служебного слова `implementation..`
5. Сборка модульных программ.

Раздел 10. Языки лисп, пролог, рубин и постскрипт

1. Функциональное программирование.
2. Атомы и списки.
3. Данные как программа.
4. Базовые функции лиспа.
5. Логическое программирование.
6. Отношения: факты и правила.
7. Запросы.
8. Языки сценариев.
9. Присваивание по ссылке.
10. Средства рубин для работы со структурами данных.
11. MVC. Ruby on rails.
12. Самодокументированность языка рубин.
13. Постфиксная запись.
14. Вычисление чисел Фибонначи на постскрипт.

Раздел 11. Теоретические вопросы программирования

1. Систематическое изучение языков программирования с разных позиций.
2. Технологическая позиция.
3. Семиотическая позиция.
4. Авторская позиция.
5. Математическая позиция.
6. Реализаторская позиция.
7. Основная цель программирования.
8. Главная проблема программирования.
9. Источники сложности в программировании.
10. Средства борьбы со сложностью.

11. Технологический процесс разработки программ.
12. Методы проектирования структуры программы.
13. Характеристики алгоритмов и программ.
14. Правильность. Надежность. Дружественность. Сопровождение.
15. Стил ь программирования.
16. Требования к оформлению программы.
17. Мобильность.
18. Использование процедурного, объектного, ситуационного, функционального, доказательного, логического программирования при разработке ПО/МО.
19. Категории ПО.

Темы курсовых работ:

1. Исследование алгоритмов сортировки данных: метод пузырька, сортировка выбором.
2. Исследование алгоритмов сортировки данных: метод пузырька, метод Шелла.
3. Исследование алгоритмов сортировки данных: метод пузырька, сортировка бинарным деревом.
4. Исследование алгоритмов сортировки данных: метод пузырька, сортировка массивом.
5. Исследование алгоритмов сортировки данных: метод пузырька, быстрая сортировка.
6. Исследование алгоритмов сортировки данных: сортировка выбором, метод Шелла.
7. Исследование алгоритмов сортировки данных: сортировка выбором, сортировка бинарным деревом.
8. Исследование алгоритмов сортировки данных: сортировка выбором, сортировка массивом.
9. Исследование алгоритмов сортировки данных: сортировка выбором, быстрая сортировка.
10. Исследование алгоритмов сортировки данных: метод Шелла сортировка, бинарным деревом.
11. Исследование алгоритмов сортировки данных: метод Шелла, сортировка массивом.
12. Исследование алгоритмов сортировки данных: метод Шелла, быстрая сортировка.
13. Исследование алгоритмов сортировки данных: сортировка бин. деревом, сортировка массивом.
14. Исследование алгоритмов сортировки данных: сортировка бин. деревом, быстрая сортировка.
15. Исследование алгоритмов сортировки данных: сортировка массивом, быстрая сортировка.

1. Алгоритм. Способы записи алгоритма. Блок-схема. Алгоритмические языки. Языки программирования.
2. Основные языки программирования. Уровни языков программирования. Сравнительный обзор языков программирования бэйсик, си++ и паскаль.
3. Общее определение языка программирования. Модель передачи сообщения. Классификация ошибок в программе.
4. Основные понятия языков программирования: текст программы, алфавит, лексемы. Типы лексем. Комментарии в языке паскаль.
5. Типы данных и переменные. Иерархия типов данных. Наиболее общие операции.
6. Скалярные и дискретные типы данных.
7. Целые типы.
8. Перечислимый, логический и символьный типы данных. Ограниченные типы.
9. Вещественный тип. Составные типы. Регулярный тип данных.
10. Строки.
11. Множества.
12. Записи. Оператор присоединения.
13. Записи с вариантами.
14. Тип указателей.
15. Файлы. Работа с типизированными файлами.
16. Текстовые файлы.
17. Синтаксические диаграммы. Алфавит и лексика языка Паскаль.
18. Общая структура программы на языке Паскаль. Блоки деклараций и операторов. Операторы: пустой, перехода. Структурное программирование.
19. Операторы: присваивания, условный и вызова процедуры.
20. Операторы циклов. Стандартные процедуры break, continue, exit и halt.
21. Операторы: составной и варианта.
22. Подпрограммы.
23. Рекурсия. Директивы forward и external подпрограмм.
24. Способы передачи параметров в подпрограммы.
25. Открытые массивы.
26. Видимость имен. Побочный эффект подпрограмм.
27. Динамические, статические и временные данные.
28. Совместимость типов. Преобразование типов.
29. Константы.
30. Понятие об объектном типе данных.
31. Структуры данных: стеки, деки. Прерывания. Организация подпрограмм и прерываний.
32. Структуры данных: очереди. Использование очередей для буферизации между асинхронными процессами.
33. Структуры данных: списки однонаправленные.
34. Структуры данных: списки двунаправленные.

1. Алгоритм. Способы записи алгоритма. Блок-схема. Алгоритмические языки. Языки программирования.
2. Основные языки программирования. Уровни языков программирования. Сравнительный обзор языков программирования бэйсик, си++ и паскаль.
3. Общее определение языка программирования. Модель передачи сообщения. Классификация ошибок в программе.
4. Основные понятия языков программирования: текст программы, алфавит, лексемы. Типы лексем. Комментарии в языке паскаль.
5. Типы данных и переменные. Иерархия типов данных. Наиболее общие операции.
6. Скалярные и дискретные типы данных.
7. Целые типы.
8. Перечислимый, логический и символьный типы данных. Ограниченные типы.
9. Вещественный тип. Составные типы. Регулярный тип данных.
10. Строки.
11. Множества.
12. Записи. Оператор присоединения.
13. Записи с вариантами.
14. Тип указателей.
15. Файлы. Работа с типизированными файлами.
16. Текстовые файлы.
17. Синтаксические диаграммы. Алфавит и лексика языка Паскаль.
18. Общая структура программы на языке Паскаль. Блоки деклараций и операторов. Операторы: пустой, перехода. Структурное программирование.
19. Операторы: присваивания, условный и вызова процедуры.
20. Операторы циклов. Стандартные процедуры break, continue, exit и halt.
21. Операторы: составной и варианта.
22. Подпрограммы.
23. Рекурсия. Директивы forward и external подпрограмм.
24. Способы передачи параметров в подпрограммы.
25. Открытые массивы.
26. Видимость имен. Побочный эффект подпрограмм.
27. Динамические, статические и временные данные.
28. Совместимость типов. Преобразование типов.
29. Константы.
30. Понятие об объектном типе данных.
31. Структуры данных: стеки, деки. Прерывания. Организация подпрограмм и прерываний.
32. Структуры данных: очереди. Использование очередей для буферизации между асинхронными процессами.
33. Структуры данных: списки однонаправленные.
34. Структуры данных: списки двунаправленные.