

Негосударственное частное образовательное учреждение
высшего образования
"Алтайский экономико-юридический институт"
Кафедра государственно-правовых дисциплин



Рабочая программа по дисциплине

Вычислительная техника и основы программирования

для направления 38.03.01 Экономика
квалификация (степень) "бакалавр"
Профиль подготовки
"Финансы и кредит"

Барнаул 2016

Рабочая программа разработана _____

Рабочая программа рассмотрена и утверждена " ____ " _____ 2016 г.

Программа рассмотрена и одобрена методической комиссией.

ОГЛАВЛЕНИЕ

1.Цели и задачи дисциплины.	4
2. Место дисциплины в структуре ООП, требования к знаниям, умениям и навыкам студента.	6
3. Учебно-тематический план дисциплины (с указанием общей трудоемкости и количеством часов, отводимых на различные разделы и виды учебной деятельности).	7
4. Содержание дисциплины.	9
5. Планы практических занятий.	12
6. Самостоятельная работа студентов.	28
7. Образовательные технологии, используемые в преподавании дисциплины.....	29
8. Критерии оценки результатов обучения.	30
9. Материалы для текущей, промежуточной и итоговой аттестации	34
10. Материально-техническое обеспечение дисциплины.	77
11. Информационное обеспечение дисциплины.	78
12. Методические материалы, определяющие процедуры оценивания результатов освоения программы дисциплины	79
13. Методические указания для обучающихся по освоению дисциплины	82

1.Цели и задачи дисциплины.

Целью учебной дисциплины «Вычислительная техника и основы программирования» является формирование у студентов практических навыков по информатике и программированию решения различных задач, по основам алгоритмизации вычислительных процессов, развитие умения работы с персональным компьютером на высоком пользовательском уровне; создание необходимой основы для использования современных средств вычислительной техники и пакетов прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин в течение всего периода обучения.

Задачи дисциплины «Вычислительная техника и основы программирования»:

- 1- изучение основных принципов хранения и обработки информации с применением программных и аппаратных средств вычислительной техники;
- 2- изучение методов разработки алгоритмов решения задач, связанных с обработкой информации, и приобретения навыков алгоритмирования;
- 3- изучение алгоритмических языков программирования и приобретение навыков написания программ для решения типовых задач по вводу и обработке информации;
- 4- изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования.

2. Место дисциплины в структуре ООП, требования к знаниям, умениям и навыкам студента.

Согласно ФГОС ВО дисциплина «Вычислительная техника и основы программирования» относится к вариативной части Блока 1 программы бакалавриата обязательных дисциплин.

Дисциплина изучается в третьем семестре, то есть на момент, когда студенты уже освоили базовый курс «Информационные технологии в экономике». К началу обучения дисциплина студенты должны иметь сведения по информатике в объёме курса «Информационные технологии в экономике», а так же школьного курса информатики.

Дисциплина «Вычислительная техника и основы программирования» направлена на формирование у студентов следующих компетенций:

1. Студент должен обладать общепрофессиональными компетенциями (ОПК):

способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	(ОПК-1)
---	---------

2. Студент должен обладать профессиональными компетенциями (ПК):

способностью использовать для решения аналитических и исследовательских задач современные технические средства и информационные технологии	(ПК-8)
способностью использовать для решения коммуникативных задач современные технические средства и информационные технологии	(ПК-10)

В результате освоения дисциплины обучающийся должен **уметь**:

1. использовать персональный компьютер в профессиональной и повседневной деятельности;
2. использовать интегрированные среды систем программирования;

3. использовать методы алгоритмизации и программирования на языках высокого уровня;
4. составлять алгоритмы и реализовывать их на алгоритмических языках программирования.

В результате освоения дисциплины обучающийся должен **знать**:

- 1- устройство, назначение и правила эксплуатации ЭВМ и её периферийных устройств;
- 2- современные операционные системы;
- 3- правила использования программных средств автоматизации математических расчетов;
- 4- интегрированные среды систем программирования;
- 5- методы алгоритмизации и программирования на языках высокого уровня;
- 6- алгоритмы решения математических задач.

3. Учебно-тематический план дисциплины (с указанием общей трудоемкости и количеством часов, отводимых на различные разделы и виды учебной деятельности).

Общая трудоемкость дисциплины «Вычислительная техника и основы программирования» составляет 2 зачетные единицы (72 часа).

№ п/п	Раздел (тема) учебной дисциплины	Виды учебной деятельности, и трудоемкость (в часах)					с работы преподавателем
		Лекции	Практические занятия (семинары)	СРС	Контроль	Всего часов	
1	Понятие алгоритма и его свойства.	2	2	4		8	4
2	Основы программирования на языке Си.	2	2	4		8	4
3	Линейный вычислительный процесс.	2	2	4		8	4
4	Операторы управления. Алгоритмы ветвления.	2	2	4		8	4
5	Операторы управления. Циклические алгоритмы.	2	2	4		8	4
6	Производные типы. Одномерные массивы.	2	2	4		8	4
7	Производные типы. Двумерные массивы.	2	2	4		8	4
8	Указатели.	2	2	4		8	4
9	Функции.	2	2	4		8	4
	ИТОГО: 2 ЗЕТ	18	18	36		72	36
	Форма контроля – зачет						

4. Содержание дисциплины.

Лекция 1 (2 часа)

Тема: Понятие алгоритма и его свойства.

Вопросы:

1. Понятие алгоритма.
2. Блок-схема алгоритма.
3. Основные алгоритмические конструкции.
4. Базовые алгоритмы.

Лекция 2 (2 часа)

Тема: Основы программирования на языке Си.

Вопросы:

1. Начальные сведения о языке программирования Си.
2. Типы данных.
3. Переменные и константы.
4. Преобразование типов.
5. Ввод и вывод данных.

Лекция 3 (2 часа)

Тема: Линейный вычислительный процесс.

Вопросы:

1. Операторы.
2. Операторы. Точка с запятой.
3. Комментарии.
4. Основные математические операторы.

Лекция 4 (2 часа)

Тема: Операторы управления. Алгоритмы ветвления.

Вопросы:

1. Выражение и операторы.
2. Построение условий.
3. Алгоритмы разветвления.
4. Оператор ветвления if.
5. Оператор ветвления switch.

Лекция 5 (2 часа)

Тема:Операторы управления. Циклические алгоритмы.

Вопросы:

1. Оператор цикла for.
2. Оператор цикла while.
3. Оператор цикла do.
4. Оператор break.
5. Оператор continue.

Лекция 6 (2 часа)

Тема:Производные типы. Одномерные массивы.

Вопросы:

1. Понятие массива.
2. Определение одномерного массива.
3. Обращение к элементам одномерного массива.
4. Типовые задачи по обработке одномерных массивов.

Лекция 7 (2 часа)

Тема:Производные типы. Двумерные массивы.

Вопросы:

- 1.Понятие двумерного массива.
- 2.Определение двумерного массива.
- 3.Обращение к элементам двумерного массива.

4. Типовые задачи по обработке двумерных массивов.

Лекция 8 (2 часа)

Тема: Указатели.

Вопросы:

1. Понятие указателя.
2. Адресные операции.
3. Адресная арифметика.
4. Динамическое распределение памяти.
5. Связь между указателями и массивами.

Лекция 9 (2 часа)

Тема: Функции.

Вопросы:

1. Определение функции.
2. Описание функции.
3. Управление видимости функций.
4. Вызов функций.
5. Передача параметров.
6. Передача массивов в качестве параметров.

5. Планы практических занятий.

Цель проведения семинарских (практических) занятий заключается в закреплении теоретических знаний, полученных на лекциях, и в процессе самостоятельного изучения студентами специальной литературы.

Практическое занятие №1 (2 часа)

Тема: Понятие алгоритма и его свойства.

Вопросы:

1. Что такое алгоритм?
2. Что такое блок-схема?
3. Основные элементы блок-схем.
4. Правила построения блок-схем.
5. Базовые алгоритмы.

Задания для самоконтроля:

1. Дать определение алгоритма.
2. Дать определение математического алгоритма.
3. Дать определение блок-схем.
4. Изучить нормативные документы по составлению блок-схем.
5. Какие алгоритмы можно иллюстрировать с помощью блок-схем?
6. Что такое граф-схема алгоритма?
7. Какие формы представления алгоритмов существуют?
8. Для чего необходимо письменное представление алгоритмов?

Задачи:

Задача 1.

1. Решить задачу: Крестьянину нужно перевезти через реку волка, козу и капусту. Но лодка такова, что в ней может поместиться только

крестьянин, а с ним или один волк, или одна коза, или одна капуста. Но если оставить волка с козой, то волк съест козу, а если оставить козу с капустой, то коза съест капусту. Как перевез свой груз крестьянин? (Задача, придуманная ученым монахом и математиком из Ирландии Алкуином (735–804))

2. Записать словесный алгоритм решения задачи.
3. Записать решение задачи в виде блок-схемы.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №2 (2 часа)

Тема: Основы программирования на языке Си.

Вопросы:

1. Что такое алфавит языка программирования?
2. Какие типы данных существуют и чем они отличаются друг от друга?
3. Какова структура программы на языке Си?
4. Что такое константы в языке Си и как они описываются?
5. Что такое переменные в языке Си и как они описываются?
6. Чем переменные отличаются от констант в языке Си?
7. Что такое препроцессор?
8. Стандартные функции ввода.
9. Стандартные функции вывода.

Задания для самоконтроля:

1. Ознакомится с алфавитом языка Си.
2. Ознакомиться со способами описания переменных и констант.
3. Ознакомиться со структурой программы на языке Си.
4. Ознакомиться с функциями ввода/вывода.

Задачи:

Задача 1. Разработать программу на языке Си. Необходимо вывести текст в соответствии с вариантом задания.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры с ответами.
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №3 (2 часа)

Тема: Линейный вычислительный процесс.

Вопросы:

1. Операторы.
2. Операторы. Точка с запятой.
3. Комментарии.
4. Основные математические операторы.

Вопросы для самоконтроля:

1. Что такое оператор в языке программирования?
2. Классификация операторов в языке программирования Си.
3. Что такое простой оператор в языке Си?
4. Что такое составной оператор в языке Си?
5. Определение пустого оператора в языке Си.
6. Что такое метка оператора в языке Си?
7. Точка с запятой.
8. Унарные операторы в языке Си.
9. Бинарные операторы в языке Си.
10. Комментарии в языке Си.
11. Основные математические операторы.

Задачи:

Задача 1. Разработать программу на языке Си реализующую линейный алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №4 (2 часа)

Тема: Операторы управления. Алгоритмы ветвления.

Вопросы:

1. Выражение и операторы.
2. Построение условий.
3. Алгоритмы разветвления.
4. Оператор ветвления if.
5. Оператор ветвления switch.

Вопросы для самоконтроля:

1. Что такое выражение в языке Си?
2. Чем отличаются выражения и операторы в языке Си?
3. Построение условий логических выражений.
4. С помощью каких операторов реализуются алгоритмы ветвления в языке Си?
5. Использование меток в алгоритмах ветвления.
6. Оператор goto.

Задачи:

Задача 1. Разработать программу на языке Си реализующую алгоритм ветвления. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор if.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Задача 2. Разработать программу на языке Си реализующую алгоритм ветвления. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор switch.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №5 (2 часа)

Тема: Операторы управления. Циклические алгоритмы.

Вопросы:

1. Оператор цикла for.
2. Оператор цикла while.
3. Оператор цикла do.
4. Оператор break.
5. Оператор continue.

Вопросы для самоконтроля

1. Для реализации каких задач предназначен оператор for?
2. Для реализации каких задач предназначен оператор while?
3. Для реализации каких задач предназначен оператор do?
4. В чём различие операторов цикла for, while, do?
5. Приведите примеры стандартных задач, для которых используются операторы циклов for, while, do?
6. Оператор break.
7. Оператор continue.

Задачи:

Задача 1. Разработать программу на языке Си реализующую циклический алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор while.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.

3. В программе предусмотреть возможность повторного запуска решения математических функций с помощью оператора `do`.
4. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 2. Разработать программу на языке Си реализующую циклический алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор `for`.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска решения математических функций с помощью оператора `do`.
4. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 3. Разработать программу на языке Си реализующую циклический алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор `do`.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска решения математических функций с помощью оператора `do`.

4. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №6 (2 часа)

Тема: Производные типы. Одномерные массивы.

Вопросы:

1. Понятие массива.
2. Определение одномерного массива.
3. Обращение к элементам одномерного массива.
4. Типовые задачи по обработке одномерных массивов.

Вопросы для самоконтроля

- 1- Что такое массив?
- 2- С какого номера начинаются индексы элементов массива в языке Си?
- 3- Что такое индекс элемента массива?
- 4- Что такое количество элементов в массиве?
- 5- Являются ли равнозначным понятия элемент массива и индекс элемента?

- 6- Являются ли равнозначным понятия максимальный элемент массива и максимальный индекс?
- 7- Должно ли совпадать число, обозначающее фактическое количество элементов в массиве, и число, обозначающее количество возможных элементов в массиве?
- 8- Что такое типа элементов массива, и каким он может быть?
- 9- Как происходит обращение к элементу массива?

Задачи:

Задача 1. Разработать программу на языке Си, реализующую работу с одномерным массивом. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), а так же выполнить дополнительное задание согласно элементу.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №7 (2 часа)

Тема: Производные типы. Двумерные массивы.

Вопросы:

1. Понятие двумерного массива.
2. Определение двумерного массива.
3. Обращение к элементам двумерного массива.
4. Типовые задачи по обработке двумерных массивов.

Вопросы для самоконтроля:

1. Определение двумерного массива.
2. Являются ли понятия двумерного массива и матрицы равнозначными?
3. Каким образом возможно обратиться к элементу двумерного массива?
4. При обращении к двумерному массиву первый индекс обозначает столбец или строку?
5. Является ли одномерный массив частным случаем двумерного массива?
6. Возможно ли обратиться к произвольному элементу массиву или обращение возможно только к элементам по порядку?
7. Назовите стандартные задачи при работе с массивами.
8. Возможно ли создать трёхмерный или четырехмерный массив?

Задачи:

Задача 1. Разработать программу на языке Си, реализующую работу с двумерным массивом, состоящим из чисел. Необходимо найти минимальный

и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), найти сумму всех элементов на главной диагонали, а так же выполнить дополнительное задание согласно элементу.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №8 (2 часа)

Тема: Указатели.

Вопросы:

1. Понятие указателя.
2. Адресные операции.
3. Адресная арифметика.
4. Динамическое распределение памяти.

5. Связь между указателями и массивами.

Вопросы для самоконтроля:

1. Что такое указатель в языке Си?
2. Какие существуют адресные операции?
3. Как узнать адрес переменной с помощью указателя?
4. Как узнать значение переменной с помощью указателя?
5. Какая спецификация стандартно используется для вывода адреса переменной?
6. Что такое адресная арифметика?
7. Что такое динамическое распределение памяти?
8. Какие переменные называются динамическими?
9. Как происходит обращение к элементу одномерного массива с помощью указателя?
10. Как происходит обращение к элементу двумерного массива с помощью указателя?

Задачи:

Задача 1. Разработать программу на языке Си, реализующую работу с двумерным массивом, состоящим из чисел. Все основные переменные будут динамическими. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), найти сумму всех элементов на главной диагонали, а так же выполнить дополнительное задание согласно элементу.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.

4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

Практическое занятие №9 (2 часа)

Тема: Функции.

Вопросы:

1. Определение функции.
2. Описание функции.
3. Управление видимости функций.
4. Вызов функций.
5. Передача параметров.
6. Передача массивов в качестве параметров.

Вопросы для самоконтроля:

1. Что такое функция в языке Си?
2. Что такое подпрограмма?
3. Что такое определение функции в языке Си?

4. Что такое описание функции в языке Си?
5. Что такое «тело функции»?
6. Что такое «формальные параметры» функции?
7. Каждая ли функция должна возвращать какие-либо параметры?
8. Что означает ключевое слово `static`?
9. Чем отличается описание функции возвращающей результат и не возвращающей результат?
10. Может ли одна функция вызывать другую?

Задачи:

Задача 1. Разработать программу на языке Си, реализующую работу с двумерным массивом, состоящим из чисел. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), найти сумму всех элементов на главной диагонали, а так же выполнить дополнительное задание согласно элементу. Все математические алгоритмы необходимо оформить функциями.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора `do`.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Основная литература:

1. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.

Дополнительная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.

6. Самостоятельная работа студентов.

Самостоятельная работа студентов делится на два вида: аудиторная самостоятельная работа студентов под контролем преподавателя и внеаудиторная работа студентов.

К самостоятельной работе студентов под контролем преподавателя относится:

- 1) решение задач, тестов, написание программ;
- 2) выполнение различного вида контрольных работ;
- 3) участие в конференциях, коллоквиумах и т.д.

К самостоятельной внеаудиторной работе относится:

- 1) подготовка к аудиторным занятиям (лекциям, практическим и др.) и выполнение соответствующих заданий;
- 2) самостоятельная работа над отдельными темами учебных дисциплин в соответствии с учебно-тематическими планам;
- 3) подготовка ко всем видам контрольных испытаний;
- 4) решение задач, написание программ, тестов, указанных в планах практических занятий;
- 5) подготовка докладов, рефератов;
- 6) изучение учебно-методических пособий;
- 7) подготовка презентаций и т.д.;

7. Образовательные технологии, используемые в преподавании дисциплины.

При изучении учебной дисциплины «Вычислительная техника и основы программирования» для получения знаний и формирования компетенций могут быть использованы следующие образовательные технологии и формы преподавания:

- 1) Проблемная лекция.
- 2) Лекция-беседа.
- 3) Лекция-дискуссия.
- 4) Решение задач.
- 5) Решение тестов.
- 6) Подготовка презентаций.
- 7) Составление таблиц и схем.

Чтение лекций производится с использованием мультимедийной аппаратуры, все практические семинары проводятся в аудитории, оснащённой персональными компьютерами с установленной средой разработки MinGW.

8. Критерии оценки результатов обучения.

Критерии оценки результатов обучения представляет собой дифференциацию оценки знаний студента по балльно-рейтинговой системе.

Характеристика работы студента	Диапазон баллов рейтинга	Традиционная шкала оценки
«Отлично» – работа высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	95-100	5+
	90-94	5
	85-89	5-
«Хорошо» – уровень выполнения работы отвечает всем основным	80-84	4+

требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки		
	75-79	4
	70-74	4-
«Удовлетворительно» – уровень выполнения работы отвечает большинству основных требований, теоретическое содержание курса освоено частично, но	65-69	3+
	60-64	3

<p>пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном</p>		
<p>сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками</p>	55-59	3-
<p>«Неудовлетворительно» (без возможности пересдачи) – теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, всевыполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа</p>	0-54	2

над материалом курса не приведет к какому- либо значимому повышению качества выполнения учебных заданий		
--	--	--

9. Материалы для текущей, промежуточной и итоговой аттестации.

Тестовые задания:

1. Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	А	В	С	D	E	F
А			3			
В			9		4	
С	3	9		3	8	
D			3		2	
E		4	8	2		7
F					7	

Определите длину кратчайшего пути между пунктами А и F (при условии, что передвигаться можно только по построенным дорогам).

1-11

2-13

3-15

4-17

2. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А–10, Б–001, В–0001, Г–110, Д–111.

Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно? Коды остальных букв меняться не должны. Выберите правильный вариант ответа.

1 это невозможно

2 для буквы В — 000

3 для буквы Б — 0

4 для буквы Г — 11

3. В программе описан одномерный целочисленный массив с индексами от 0 до n . Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, обрабатывающей данный массив:

```
s = 0;
z = A[n];
for (i = 0; i <= n; i++){
    if (A[i] > z)
        s++;
}
```

Чему будет равно значение переменной s после выполнения данной программы, при любых значениях элементов массива?

1 Максимальному элементу в массиве A

2 Количеству элементов массива A , больших последнего элемента массива

3 Индексу последнего элемента массива A , который меньше $A[0]$

4 Количеству элементов массива A , меньших последнего элемента массива

4. В программе описан одномерный целочисленный массив с индексами от 0 до n (т.е. первый элемент имеет индекс 0, последний - индекс n). Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, обрабатывающей данный массив:

Си
<pre>s = n; z = A[0]; for (i = 1; i <= n; i++){ if (A[i] == z) s--; }</pre>

Чему будет равно значение переменной s после выполнения данной программы, при любых значениях элементов массива?

- 1 Количеству элементов массива A , больших первого элемента массива
- 2 Количеству элементов массива A , не превосходящих первого элемента массива

3 Количеству элементов массива A , не равных первому элементу массива

- 4 Количеству элементов массива A , равных первому элементу массива

5. В программе описан одномерный целочисленный массив с индексами от 0 до n . Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Си
<pre>s = 0; z = A[n]; for (i = 0; i < n; i++) if (A[i] < z) s=s + A[i];</pre>

Чему будет равно значение переменной s после выполнения данной программы? Ответ должен быть верным при любых значениях элементов массива.

- 1 Минимальному элементу в массиве A
- 2 Количеству элементов массива A , меньших последнего элемента массива

3 Сумме всех элементов массива A, меньших последнего элемента массива

4 Индексу первого элемента массива A, который меньше A[n]

6. В программе описан одномерный целочисленный массив с индексами от 0 до n. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Си
<pre>s = 0; z = A[n]; for (i = 0; i <= n-1; i++) if (A[i] > z) s=s+ A[i];</pre>

Чему будет равно значение переменной s после выполнения данной программы? Ответ должен быть верным при любых значениях элементов массива.

1 Максимальному элементу в массиве A

2 Количеству элементов массива A, больших последнего элемента массива

3 Сумме всех элементов массива A, больших последнего элемента массива

4 Индексу последнего элемента массива A, который больше A[n]

7. В программе описан одномерный целочисленный массив с индексами от 0 до n (т.е. первый элемент имеет индекс 0, последний - индекс n). Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Си
<pre>s = 0; for (i = 1; i <= n; i++) if (A[i] == A[0]) s=i;</pre>

Чему будет равно значение переменной s после выполнения данной программы? Ответ должен быть верным при любых значениях элементов массива.

1 Минимальному элементу в массиве A

2 Количеству элементов массива A , равных первому элемента массива

3 Сумме всех элементов массива A , равных последнему элементу массива

4 Наибольшему индексу k , для которого элемент массива с индексом k равен первому элементу массива

8. В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Бейсик	Паскаль
<pre>s = 0 n = 10 FOR i = 1 TO n IF A(i)-A(i-1) > 10 THEN s = s+A(i) END IF NEXT i</pre>	<pre>s := 0; n := 10; for i := 1 to n do begin if A[i]-A[i-1] > 10 then s := s+A[i]; end;</pre>
Си	Алгоритмический язык
<pre>s = 0; n = 10; for (i = 1; i <= n; i++) if(A[i]-A[i-1] > 10) s = s+A[i];</pre>	<pre>с := 0 n := 10 нц для i от 1 до n если A[i]-A[i-1] > 10 то с := с+А[i] все кц</pre>

В начале выполнения этого фрагмента в массиве находились числа 0,10,20,30,40,50,60,70,80,90,100, т.е. $A[0]=0$, $A[1]=10$ и т.д. Чему будет равно значение переменной s после выполнения данной программы?

- 1 0
- 2 10
- 3 30
- 4 550

9. В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Бейсик	Паскаль
<pre>s = 0 n = 10 FOR i = 1 TO n IF A(i)-A(i-1) < i THEN s = s+i END IF NEXT i</pre>	<pre>s := 0; n := 10; for i := 1 to n do begin if A[i]-A[i-1] < i then s := s+i; end;</pre>
Си	Алгоритмический язык
<pre>s = 0; n = 10; for (i = 1; i <= n; i++) if(A[i]-A[i-1] < i) s = s+i;</pre>	<pre>s := 0 n := 10 нц для i от 1 до n если A[i]-A[i-1] < i то s := s+i все кц</pre>

В начале выполнения этого фрагмента в массиве находились числа Фибоначчи: 1,1,2,3,5,8,13,21,34,55,89, т.е. $A[0]=1$, $A[1]=1$, $A[2]=2$ и т.д. Чему будет равно значение переменной s после выполнения данной программы?

Условие $A[i]-A[i-1]<i$ выполнено $i=1,\dots,6$. Поэтому итоговое значение s равно $1+2+3+4+5+6=(1+6)\cdot 6/2=7\cdot 3=21$.

- 1 0
- 2 21
- 3 34

4 55

10. В программе описан одномерный целочисленный массив с индексами от 0 до 100. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Чему будет равно значение переменной s после выполнения данной программы?

Бейсик	Паскаль
<pre>N = 100 FOR i = 0 TO n A(i) = i NEXT i FOR i = 1 TO n A(i) = A(i) - A(i-1) NEXT i s = A(90)</pre>	<pre>n := 100; for i := 0 to n do begin A[i] := i; end; for i := 1 to n do begin A[i] := A[i] - A[i-1]; end; s := A[90]</pre>
Си	Алгоритмический язык
<pre>n = 100; for (i = 0; i <= n; i++) A[i] = i for (i = 1; i <= n; i++) A[i] = A[i] - A[i-1]; s = A[90]</pre>	<pre>n := 100 <u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> n A[i] := i <u>кц</u> <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> n A[i] := A[i] - A[i-1] <u>кц</u> s := A[90]</pre>

1 0

2 30

3 45

4 550

11. В программе описан одномерный целочисленный массив с индексами от 0 до n . Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, обрабатывающей данный массив:

Бейсик	Паскаль
<pre>s = 0 z = A(n) FOR i = 0 TO n IF A(i) > z THEN s = s + 1 NEXT i</pre>	<pre>s:=0; z:=A[n]; for i:=0 to n do begin if A[i]>z then s:=s+1; end</pre>
Си	Алгоритмический язык
<pre>s = 0; z = A[n]; for (i = 0; i <= n; i++){ if (A[i] > z) s++; }</pre>	<pre>s:=0 z:=A[n] нцдляіот0до n если A[i]> z то s:=s+1 все кц</pre>

Чему будет равно значение переменной s после выполнения данной программы, при любых значениях элементов массива?

1 Максимальному элементу в массиве A

2 **Количеству элементов массива A , больших последнего элемента массива**

3 Индексу последнего элемента массива A , который меньше $A[0]$

4 **Количеству элементов массива A , меньших последнего элемента массива**

12. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх \uparrow , вниз \downarrow , влево \leftarrow , вправо \rightarrow .

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл

ПОКА < условие >

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся.

Схема лабиринта:

1						
2						
3						
4						
5						
6						
	A	B	C	D	E	F

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА<справа свободно ИЛИ снизу свободно>

ПОКА <снизу свободно>

вниз

КОНЕЦ ПОКА

ПОКА<справа свободно>

вправо

КОНЕЦ ПОКА

КОНЕЦ ПОКА

КОНЕЦ

1 18

2 22

3 26

4 30

13. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
--------------	-------------	--------------	---------------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
------------------------	-----------------------	-----------------------	------------------------

Цикл

ПОКА < условие >

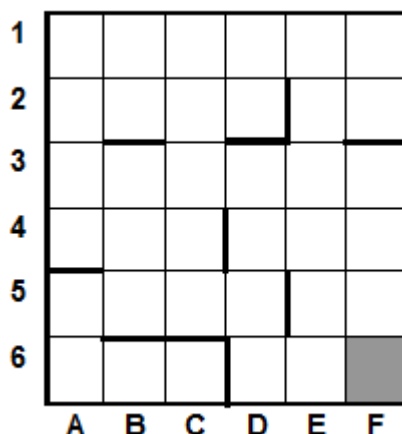
последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним

стены, то он разрушится и программа прервётся. Схема лабиринта:



Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА<справа свободно ИЛИ снизу свободно>

ПОКА <справа свободно>

вправо

КОНЕЦ ПОКА

ПОКА<снизу свободно>

вниз

КОНЕЦ ПОКА

КОНЕЦ ПОКА

КОНЕЦ

1 18

2 21

3 24

4 27

14. У исполнителя Калькулятор две команды, которым присвоены номера:

1. **отними 2**
2. **раздели на 3**

Выполняя первую из них, Калькулятор отнимает от числа на экране 2, а выполняя вторую, делит его на 3 (если деление нацело невозможно, Калькулятор отключается).

Запишите порядок команд в программе получения из числа 37 число 3, содержащей не более 5 команд, указывая лишь номера команд.

(Например, программа **2121** – это программа

раздели на 3

отними 2

раздели на 3

отними 2

Эта программа, например, преобразует число 60 в число 4.)

Ответ: 11212

15. Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre> a = 30 b = 6 a = a / 5 * b IF a > b THEN c = a - 4 * b ELSE c = a + 4 * b ENDIF </pre>	<pre> a := 30; b := 6; a := a / 5 * b; if a > b then c := a - 4 * b else c := a + 4 * b; </pre>
Си	Алгоритмический
<pre> a = 30; b = 6; a = a / 5 * b; if (a > b) c = a - 4 * b; else c = a + 4 * b; </pre>	<pre> a := 30 b := 6 a := a / 5 * b <u>если</u> a > b <u>то</u> c := a - 4 * b <u>иначе</u> c := a + 4 * b <u>все</u> </pre>

Ответ: 12

16. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre> a = 30 b = 6 a = a / 3 * b IF a > b THEN c = a - 5 * b ELSE c = a + 5 * b ENDIF </pre>	<pre> a := 30; b := 6; a := a / 3 * b; if a > b then c := a - 5 * b else c := a + 5 * b; </pre>
Си	Алгоритмический
<pre> a = 30; b = 6; a = a / 3 * b; if (a > b) c = a - 5 * b; else c = a + 5 * b; </pre>	<pre> a := 30 b := 6 a := a / 3 * b <u>если</u> a > b <u>то</u> c := a - 5 * b <u>иначе</u> c := a + 5 * b <u>все</u> </pre>

Ответ: 30

17. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках

программирования):

Бейсик	Паскаль
<pre> a = 60 b = 20 a = a / b * 2 IF a > b THEN c = a - 4 * b ELSE c = a + 4 * b ENDIF </pre>	<pre> a := 60; b := 20; a := a / b * 2; if a > b then c := a - 4 * b else c := a + 4 * b; </pre>
Си	Алгоритмический
<pre> a = 60; b = 20; a = a / b * 2; if (a > b) c = a - 4 * b; else c = a + 4 * b; </pre>	<pre> a := 60 b := 20 a := a / b * 2 <u>если</u> a > b <u>то</u>c := a - 4 * b <u>иначе</u>c := a + 4 * b <u>все</u> </pre>

Ответ: 86

18. Определите значение переменной c после выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> a = 30 b = 10 a = a - b * 2 IF a > b THEN c = a * 4 * b ELSE c = a * 4 / b ENDIF </pre>	<pre> a := 30; b := 10; a := a - b * 2; if a > b then c := a * 4 * b else c := a * 4 / b; </pre>
Си	Алгоритмический
<pre> a = 30; b = 10; a = a - b * 2; if (a > b) c = a * 4 * b; else c = a * 4 / b; </pre>	<pre> a := 30 b := 10 a := a - b * 2 <u>если</u> a > b <u>то</u>c := a*4 * b <u>иначе</u> := a*4 /b <u>все</u> </pre>

Ответ: 4

19. Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre> a = 50 b = 10 a = a / b * 2 IF a > b THEN c = a - 4 * b ELSE c = a + 4 * b ENDIF </pre>	<pre> a := 50; b := 10; a := a / b * 2; if a > b then c := a - 4 * b else c := a + 4 * b; </pre>
Си	Алгоритмический
<pre> a = 50; b = 10; a = a / b * 2; if (a > b) c = a - 4 * b; else c = a + 4 * b; </pre>	<pre> a := 50 b := 10 a := a / b * 2 <u>если</u> a > b <u>то</u>с := a - 4 * b <u>иначе</u>с := a + 4 * b <u>все</u> </pre>

Правильно: **50**

20. **В6.6.** Определите значение переменной **c** после выполнения следующего фрагмента программы (*записанного ниже на разных языках программирования*):

Бейсик	Паскаль
<pre> a = 50 b = 10 a = a / b * 2 IF a >= b THEN c = (a - b) * (a + b) ELSE </pre>	<pre> a := 50; b := 10; a := a / b * 2; if a >= b then c := (a - b) * (a + b); else </pre>

$c = a + 4 * b$ <i>ENDIF</i>	$c := a + 4 * b;$
Си	Алгоритмический
$a = 50;$ $b = 10;$ $a = a / b * 2;$ <i>if</i> ($a >= b$) $c = (a - b) * (a + b);$ <i>else</i> $c = a + 4 * b;$	$a := 50$ $b := 10$ $a := a / b * 2$ <i>если</i> $a >= b$ <i>то</i> $c := (a - b) * (a + b)$ <i>иначе</i> $c := a + 4 * b$ <i>все</i>

Решение: После присваивания $a = a / b * 2$, значение a станет равным $50 / 10 * 2 = (50/10) * 2 = 5 * 2 = 10$.

Значение выражения $a >= b$ – истина, поэтому в условном операторе будет выполнен вариант THEN (то) т.е. присваивание $c = (a - b) * (a + b)$. При этом значение c станет равным $(10 - 10) * (10 + 10) = 0 * 20 = 0$.

Ответ: 0

21. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках

программирования). Ответ запишите с точностью до целого.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a/5 - b IF a > b THEN c = a - 5*b ELSE c = a + 5*b ENDIF</pre>	<pre>a := 30; b := 6; a := a/5 - b; if a > b then c := a - 5*b else c := a + 5*b;</pre>
Си	Алгоритмический
<pre>a = 30; b = 6; a = a/5 - b; if (a > b) c = a - 5*b; else c = a + 5*b;</pre>	<pre>a := 30 b := 6 a := a/5 - b если a > b то c := a - 5*b иначе c := a + 5*b все</pre>

Ответ: 30

22. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках

программирования). Ответ запишите с точностью до целого.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a/5 - b/3 IF 2*a > b THEN c = a - 5*b + 27 ELSE c = a + 5*b + 27 ENDIF</pre>	<pre>a := 30; b := 6; a := a/5 - b/3; if 2*a > b then c := a - 5*b + 27 else c := a + 5*b + 27;</pre>
Си	Алгоритмический
<pre>a = 30; b = 6; a = a/5 - b/3; if (2*a > b) c = a - 5*b + 27; else c = a + 5*b + 27;</pre>	<pre>a := 30 b := 6 a := a/5 - b/3 если 2*a > b то c := a - 5*b + 27 иначе c := a + 5*b + 27 все</pre>

Ответ: 1

23. Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите с точностью до целого.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a/5 - (b - 3) IF 2*a > b THEN c = a - 5*b + 2 ELSE c = a + 5*b + 2 ENDIF</pre>	<pre>a := 30; b := 6; a := a/5 - (b - 3); if 2*a > b then c := a - 5*b + 2 else c := a + 5*b + 2;</pre>
Си	Алгоритмический
<pre>a = 30; b = 6; a = a/5 - (b - 3); if (2*a > b) c = a - 5*b + 2; else c = a + 5*b + 2;</pre>	<pre>a := 30 b := 6 a := a/5 - (b - 3) если 2*a > b то c := a - 5*b + 2 иначе c := a + 5*b + 2 все</pre>

Ответ: 35

24. Определите, что будет напечатано в результате выполнения программы, записанной ниже на разных языках программирования:

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 101 S = S + 7 N = N + 1 WEND PRINT S </pre>	<pre> var n, s: integer; begin n := 1; s := 0; while n <= 101 do begin s := s + 7; n := n + 1 end; write(s) end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 1; s = 0; while (n <= 101) { s = s + 7; n = n + 1; } printf("%d", s); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 1 s := 0 <u>нцпока</u> n <= 101 s := s + 7 n := n + 1 <u>кц</u> <u>вывод</u> s <u>кон</u> </pre>

Ответ: 707

25. (ege.yandex.ru-2)

Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 4 : S = 0 WHILE N <= 8 S = S + 15 N = N + 1 WEND PRINT S </pre>	<pre> var n, s: integer; begin n := 4; s := 0; while n <= 8 do begin s := s + 15; n := n + 1; end; write(s); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main(){ int n, s; n = 4; s = 0; while (n <= 8){ s = s + 15; n = n + 1; } printf(«%d», s); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>целн</u>, s n := 4; s := 0 <u>нцпока</u>n <= 8 s := s + 15 n := n + 1 <u>кц</u> <u>вывод</u> s <u>кон</u> </pre>

Ответ: 75

26. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 0 : S = 512 WHILE S >= 0 S = S — 20 N = N + 1 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 0; s := 512; while s >= 0 do begin s := s — 20; n := n + 1; end; write(n); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main(){ int n, s; n = 0; s = 512; while (s >= 0){ s = s — 20; n = n + 1; } printf(«%d», n); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 0; s := 512 <u>нцпока</u> s >= 0 s := s — 20 n := n + 1 <u>кц</u> <u>вывод</u> n <u>кон</u> </pre>

Ответ: 26

27. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER </pre>	<pre> var n, s: integer; </pre>

<pre> N = 14 : S = 0 WHILE N <= 18 S = S + 25 N = N + 1 WEND PRINT S </pre>	<pre> begin n := 14; s := 0; while n <= 18 do begin s := s + 25; n := n + 1; end; write(s); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main(){ int n, s; n = 14; s = 0; while (n <= 18){ s = s + 25; n = n + 1; } printf(«%d», s); } </pre>	<pre> алг нач целн, s n := 14; s := 0 <u>нцпока</u> n <= 18 s := s + 25 n := n + 1 <u>кц</u> <u>вывод</u> s кон </pre>

Ответ: 125

28. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 24 : S = 0 WHILE N <= 28 S = S + 20 N = N + 2 </pre>	<pre> var n, s: integer; begin n := 24; s := 0; while n <= 28 do begin s := s + 20; </pre>

<p><i>WEND</i></p> <p><i>PRINT S</i></p>	<p>$n := n + 2;$</p> <p><i>end;</i></p> <p><i>write(s);</i></p> <p><i>end.</i></p>
<p>Си</p>	<p>Алгоритмический</p>
<pre><i>#include<stdio.h></i> <i>void main(){</i> <i>int n, s;</i> <i>n = 24; s = 0;</i> <i>while (n <= 28){</i> <i>s = s + 20;</i> <i>n = n + 2;</i> <i>}</i> <i>printf(«%d», s);</i> <i>}</i></pre>	<pre><u>алг</u> <u>нач</u> <u>целн, s</u> $n := 24; s := 0$ <u>нцпокап</u> $<= 28$ $s := s + 20$ $n := n + 2$ <u>кц</u> <u>вывод</u> s <u>кон</u></pre>

Ответ: 60

29. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre>DIM N, S AS INTEGER N = 0 S = 0 WHILE S <= 365 S = S + 37 N = N + 5 WEND PRINT N</pre>	<pre>var n, s: integer; begin n := 0; s := 0; while s <= 365 do begin s := s + 37; n := n + 5 end; write(n) end.</pre>
Си	Алгоритмический
<pre>#include<stdio.h> void main() { int n, s; n = 0; s = 0; while (s <= 365) { s = s + 37; n = n + 5; } printf("%d", n); }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> n, s n := 0 s := 0 <u>нц пока</u> s <= 365 s := s + 37 n := n + 5 <u>кц</u> <u>вывод</u> n <u>кон</u></pre>

Ответ: 50

30. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 1 S = 5 WHILE S <= 365 S = S + 37 N = N * 2 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 1; s := 5; while s <= 365 do begin s := s + 37; n := n * 2; end; write(n) end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 1; s = 5; while (s <= 365) { s = s + 37; n = n * 2; } printf("%d", n); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 1 s := 5 <u>нц пока</u> s <= 365 s := s + 37 n := n * 2 <u>кц</u> <u>вывод</u> n <u>кон</u> </pre>

Ответ: 1024

31. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE S <= 370 S = S + 37 N = N + 10 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 1; s := 0; while s <= 370 do begin s := s + 37; n := n + 10; end; write(n) end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 1; s = 0; while (s <= 370) { s = s + 37; n = n + 10; } printf("%d", n); } </pre>	<pre> алг нач цел n, s n := 1 s := 0 нц пока s <= 370 s := s + 37 n := n + 10 кц вывод n кон </pre>

Ответ: 111

32. Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующим соотношением:

$$F(n) = F(n-1) + 2 \cdot F(n-2) \text{ при } n > 2$$

$$F(1) = 0$$

$$F(2) = 1$$

Чему равно значение функции $F(6)$? В ответ запишите только натуральное число.

Ответ: 11

33. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
--------	---------

<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 9*(x+19)*(x-19)+1; END FUNCTION </pre>	<pre> vara,b,t,M,R :integer; Function F(x: integer):integer; begin F := 9*(x+19)*(x-19) + 1; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M:=t; R:=F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 9*(x+19)*(x-19)+1; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { </pre>	<pre> <u>алг</u> <u>нач</u> <u>цела</u>, b, t, R, M a:= -20; b:= 20 M:= a; R:= F(a) <u>нцдля</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>M </pre>

<pre> M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>кОН</u> алг цел F(<u>цел</u> x) <u>нач</u> <u>знач</u> := 9*(x+19)*(x-19)+1 <u>кОН</u> </pre>
--	--

Ответ: 0

34. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19*(x-19)*(x-17)+17; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 19*(x-19)*(x-17)+17; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> </pre>	<pre> <u>алг</u> </pre>

<pre> int F(int x) { return 19*(x-19)*(x-17)+17; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>нач</u> <u>цел</u> a, b, t, R, M a := -20; b := 20 M := a; R := F(a) <u>нц для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u> <u>алг цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> :=19*(x-19)*(x-17)+17 <u>кон</u> </pre>
---	---

Ответ: 18

35. Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19*(x-19)* (x+17)+17; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 19*(x-19)* (x+17)+17; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19*(x-19)* (x+17)+17; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u>а, b, t, R, M а := -20; b := 20 М := а; R := F(а) <u>нц</u>л<u>я</u> t <u>от</u> а <u>до</u> b <u>если</u> F(t)< R <u>то</u> М := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>М <u>кон</u> <u>алгцел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> :=19*(x-19)* (x+17)+17 <u>кон</u> </pre>

Ответ: 1

36. Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) r = (x-3)*(x-3) F = r*r; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin integer r; r:= (x-3)*(x-3); F := r*r; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) {int r; r=(x-3)*(x-3); return r*r; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> целa, b, t, R, M a := -20; b := 20 M := a; R := F(a) <u>нц</u> <u>для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>M <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>цел</u> r r:= (x-3)*(x-3) <u>знач</u> :=r*r <u>кон</u> </pre>

Ответ: 3

37. Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19*(x-1)*(x-1); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 19*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u>а, b, t, R, M а := -10; b := 20 М := а; R := F(а) <u>нц</u>л<u>я</u> t <u>от</u> а <u>до</u> b <u>если</u> F(t) > R <u>то</u> М := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>М <u>кон</u> <u>алгцел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> :=19*(x-1)*(x-1) <u>кон</u> </pre>

Ответ: 20

38. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19 - 19*(x-1)*(x-1); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 19 - 19*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19 - 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, M, R a := -10; b := 20 M := a; R := F(a) нц для t от a до b если F(t) > R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 19 - 19*(x-1)*(x-1) кон </pre>

Ответ: 1

39. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 21 - 19*(x-1)*(x-1); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 21 - 19*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
Сп	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 21 - 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", R); } </pre>	<pre> алг нач цел a, b, t, M, R a := -10; b := 20 M := a; R := F(a) нц для t от a до b если F(t) > R то M := t; R := F(t) все кц вывод R кон алг цел F(цел x) нач знач := 21 - 19*(x-1)*(x-1) кон </pre>

Правильно: **21**

40. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 19*(x-21)*(x-21); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 19*(x-21)*(x-21); end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19*(x-21)*(x-21); } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t) < R) { M = t; R = F(t); } } printf("%d", R); } </pre>	<pre> алг нач цел a, b, t, M, R a := -20; b := 20 M := a; R := F(a) нц для t от a до b если F(t) < R то M := t; R := F(t) все кц вывод R кон алг цел F(цел x) нач знач := 19*(x-21)*(x-21) кон </pre>

Ответ: 19

41. **А 12 № 3354.** В программе описан одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент этой программы, в котором значения элементов массива сначала задаются, а затем меняются.

$A[0] := 1;$

for $i := 1$ to 10 do

$A[i] := 2 + A[i-1];$

for $i := 0$ to 5 do

$A[10-i] := A[i]-1;$

Чему будут равны элементы этого массива?

- 1) **1 3 5 7 9 10 8 6 4 2 0**
- 2) 1 3 5 7 9 11 13 15 17 19 21
- 3) 1 3 5 7 9 11 9 7 5 3 1
- 4) 1 3 5 7 9 11 8 6 4 3 0

42. В программе описан одномерный целочисленный массив с индексами от 0 до 10 и целочисленные переменные k , i . В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i := 0 to 10 do A[i] := i;
for i := 0 to 4 do begin
  k := A[i];
  A[i] := A[10-i];
  A[10-i] := k;
end;
```

Чему будут равны элементы этого массива?

- 1) **10 9 8 7 6 5 4 3 2 1 0**
- 2) 0 1 2 3 4 5 6 7 8 9 10
- 3) 0 1 2 3 4 5 4 3 2 1 0
- 4) 10 9 8 7 6 5 6 7 8 9 10

43. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i := 0 to 10 do
  A[i] := i;
for i := 1 to 11 do
  A[i-1] := A[11-i];
```

Чему будут равны элементы этого массива?

- 1) 10 9 8 7 6 5 4 3 2 1 0
- 2) 11 10 9 8 7 6 5 4 3 2 1
- 3) 10 9 8 7 6 5 6 7 8 9 10**
- 4) 11 10 9 8 7 6 7 8 9 10 11

44. Значения элементов двух массивов А и В размером 1 x 100 задаются с помощью следующего фрагмента программы:

```
for i:=1 to 100 do  
  A[i] := 50 - i;  
for i:=1 to 100 do  
  B[i] := A[i] + 49;
```

Сколько элементов массива В будут иметь отрицательные значения?

- 1) 1**
- 2) 10
- 3) 50
- 4) 100

45. Значения двух массивов А[1..100] и В[1..100] задаются с помощью следующего фрагмента программы:

```
for n:=1 to 100 do  
  A[n] := n - 10;  
for n:=1 to 100 do  
  B[n] := A[n]*n;
```

Сколько элементов массива В будут иметь положительные значения?

- 1) 10
- 2) 50
- 3) 90**
- 4) 100

Примерная тематика рефератов, докладов и презентаций:

1. Информационно вычислительная система (ивс).
2. Языки программирования.
3. История развития языка программирования Си.
4. Языки программирования (содержание реферата: машинный код процессора, алгоритм и программа, компиляторы и интерпретаторы, уровни языков программирования, поколения языков программирования).
5. Языки программирования (содержание реферата: языки программирования высокого уровня (Fortran, Cobol, Algol, Pascal, Basic, C, C++, Java).
6. Языки программирования (содержание реферата: языки программирования баз данных, языки программирования для Internet).
7. Алгоритмическое (модульное) программирование (содержание реферата: алгоритм, требования к модулю, переменные, константы, операции, указатели, операторы).
8. Структурное программирование (содержание реферата: подпрограммы, программирование “сверху вниз”, процедуры и функции, событийно-ориентированное программирование).

9. Проектирование программ (содержание реферата: программирование как вид деятельности, экономические аспекты программирования, этапы разработки программ, стандарты качества программного).
10. Коллективная разработка программных продуктов.

Содержание и структура реферата.

Структура реферата:

1. Введение — излагается цель и задачи работы, обоснование выбора темы и её актуальность. Объём: 1—2 страницы.
2. Основная часть — точка зрения автора на основе анализа литературы по проблеме. Объём: 12—15 страниц.
3. Заключение — формируются выводы и предложения. Заключение должно быть кратким, четким, выводы должны вытекать из содержания основной части. Объём: 1—3 страницы.
4. Список используемой литературы.

Текст и его оформление.

Размер шрифта 12—14 пунктов, гарнитура TimesNewRoman, обычный; интервал между строк: 1,5—2; размер полей: левого — 30 мм, правого — 10 мм, верхнего — 20 мм, нижнего — 20 мм.

Точку в конце заголовка не ставят. Заглавия всегда выделены жирным шрифтом. Обычно: 1 заголовок — шрифт размером 16 пунктов, 2 заголовок - шрифт размером 14 пунктов, 3 заголовок - шрифт размером 14 пунктов, курсив.

Расстояние между заголовками главы или параграфа и последующим текстом должно быть равно трем интервалам. Чтобы после оформления работы получить автоматическое оглавление, необходимо проставить названия глав как «Заголовок 1», «Заголовок 2», «Заголовок 3»:

Текст печатается на одной стороне страницы; сноски и примечания обозначаются в самом тексте, так [3, с. 55-56].

Все страницы нумеруются, начиная с титульного листа; цифру номера страницы ставят вверху по центру страницы; на титульном листе номер страницы не ставится. Каждый новый раздел начинается с новой страницы.

Перечень вопросов для подготовки к экзамену:

Тема: Понятие алгоритма и его свойства.

1. Понятие алгоритма.
2. Понятие блок-схемы.
3. Основные элементы блок-схем.
4. Правила построения блок-схем.

Тема: Основы программирования на языке Си.

1. Начальные сведения о языке программирования Си.
2. Алфавит языка Си.
3. Типы данных.
4. Переменные и константы.
5. Преобразование типов.
6. Ввод и вывод данных.

Тема: Линейный вычислительный процесс.

1. Понятие оператора.
2. Классификация операторов в языке программирования Си.
3. Простой оператор в языке Си.
4. Составной оператор в языке Си.
5. Метка оператора в языке Си.
6. Точка с запятой пустого оператора в языке Си.
7. Унарные операторы в языке Си.
8. Бинарные операторы в языке Си.
9. Комментарии в языке Си.
10. Основные математические операторы.

Тема: Операторы управления. Алгоритмы ветвления.

1. Выражение и операторы.

2. Построение условий.
3. Алгоритмы разветвления.
4. Оператор ветвления if.
5. Оператор ветвления switch.

Тема: Операторы управления. Циклические алгоритмы.

1. Оператор цикла for.
2. Оператор цикла while.
3. Оператор цикла do.
4. Оператор break.
5. Оператор continue.

Тема: Производные типы. Одномерные массивы.

1. Понятие массива.
2. Определение одномерного массива.
3. Обращение к элементам одномерного массива.
4. Типовые задачи по обработке одномерных массивов.

Тема: Производные типы. Двумерные массивы.

1. Понятие двумерного массива.
2. Определение двумерного массива.
3. Обращение к элементам двумерного массива.
4. Типовые задачи по обработке двумерных массивов.

Тема: Указатели.

1. Понятие указателя.
2. Адресные операции.
3. Адресная арифметика.
4. Динамическое распределение памяти.
5. Связь между указателями и массивами.

Тема: Функции.

1. Определение функции.
2. Описание функции.
3. Управление видимости функций.
4. Вызов функций.
5. Передача параметров.
6. Передача массивов в качестве параметров.

10. Материально-техническое обеспечение дисциплины.

Материально-техническое обеспечение дисциплины

«Административное право» включает в себя следующие средства:

- мультимедийный проектор;
- средства аудиовоспроизведения;
- компьютерный класс с установленной средой разработки MinGW;
- книжный фонд библиотеки.

11. Информационное обеспечение дисциплины.

Основная литература:

1. Шилдт Г. С: полное руководство, классическое издание. — М.: «Вильямс», 2011. — 704 с.
2. http://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%A1%D0%B8_%D0%B2_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85 - Язык Си в примерах.
3. Шелест В. Программирование – СПб.:БХВ-Санкт-Петербург, 2002, - 592 с.
4. Информационные технологии в экономике и управлении: Под ре. Трофимова В.В. - М.: Юрайт, 2011, - 478 с.
5. Егорова Е.В. Программирование на языке Си – Барнаул: Изд-во АлтГТУ, 2012. – 184 с.

Дополнительная литература:

1. Гукин Д. Язык программирования Си для "чайников" – М.:Вильямс, 2006, - 352 с.
2. Егорова Е.В. Программирование на языке высокого уровня: учебное пособие часть 1. – Барнаул: Изд-во АлтГТУ, 2010. – 209 с.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня: для вузов. – СПб.: Питер, 2010 – 460 с.
4. Подвельский В.В. Язык С++: учебное пособие для вузов. – М.: Финансы и статистика, 2004. – 560 с.

Интернет ресурсы:

1. <http://www.mingw.org/> - Home of the MinGW and MSYS Projects.
2. <http://www.intuit.ru/> - Национальный Открытый Университет «ИНТУИТ».
3. <http://dict.kolbi.ru/cgi-bin/dict2/view2.pl> - Англо-русский компьютерный словарь on-line.

4. <http://pcabc.ru/> - Самоучитель персонального компьютера.
5. <http://www.osp.ru> - Открытые системы: издания по информационным технологиям.
6. <http://www.junior.ru/wwwexam/> - Тесты по информатике и информационным технологиям.
7. <http://www.dummies.com/Section/Computers-Software.id-323006.html> - сайт издательства серии книг ForDummies (для «чайников») на английском языке, раздел Computers&Software (Компьютеры и программное обеспечение).
8. <http://habrahabr.ru/> - многофункциональный сайт, созданный для публикации новостей, аналитических статей, связанных с информационными технологиями, бизнесом и Интернетом.

12. Методические материалы, определяющие процедуры оценивания результатов освоения программы дисциплины

Текущий контроль успеваемости студентов.

Текущий контроль успеваемости – это установление уровня знаний, умений, владений студентов по отношению к объему и содержанию разделов (модулей, частей) учебных дисциплин, представленных и утвержденных в учебных планах и учебных программах.

Текущий контроль успеваемости осуществляется через комплекс испытаний студентов в виде устных и письменных опросов, коллоквиумов, контрольных работ, проверки домашних заданий, защиты отчетов, компьютерного и бланочного тестирования. Возможны и другие виды контроля по усмотрению кафедры, обеспечивающей учебный процесс по данной дисциплине, в том числе, контроль посещаемости занятий.

В систему текущего контроля рекомендуется вводить необязательные мероприятия, позволяющие повысить семестровый рейтинг, например, участие в олимпиадах, научное исследование, участие в научных конференциях с докладом по теме изучаемого предмета и т.д. с назначением определенных баллов, прибавляемых к семестровому рейтингу по дисциплине. При этом рейтинг не должен превышать 100 баллов.

Для текущего контроля успеваемости на кафедрах, осуществляющих учебный процесс, создаются и периодически актуализируются банки тестов, заданий, программы компьютерных проверок и т.п. материалы.

Виды и сроки проведения мероприятий текущего контроля устанавливаются рабочей программой учебной дисциплины.

Промежуточная аттестация.

Промежуточная аттестация студентов – это установление уровня знаний, умений, владений обучаемых, как показателя уровня освоения

требуемых компетенций, по отношению к объему и содержанию учебной дисциплины.

Оценка промежуточной аттестации студента по дисциплине формируется на основании семестрового рейтинга текущего контроля и рейтинга экзаменационного испытания. Экзаменационное испытание проводится в сроки, устанавливаемые в соответствии с утвержденными учебными планами, календарными учебными графиками и приказами.

Преподаватель имеет право принять у студента экзамен только при наличии первичных документов по учету результатов промежуточной аттестации. Первичными документами являются экзаменационные ведомости, индивидуальные разрешения на сдачу экзамена. Все первичные документы должны передаваться в деканат преподавателем лично не позднее следующего дня после проведения испытания промежуточной аттестации.

По результатам промежуточной аттестации студенту, кроме итогового рейтинга по 100-балльной шкале, выставляется итоговая отметка: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

При аттестации на «отлично», «хорошо» и «удовлетворительно» студент считается получившим положительную оценку и прошедшим промежуточную аттестацию. Положительные оценки и соответствующие рейтинги заносятся в первичные документы и зачетные книжки студентов. Записи в зачетных книжках студентов должны осуществляться только после оформления первичных документов.

Оценки «неудовлетворительно» проставляются только в первичные документы.

Неудовлетворительные результаты промежуточной аттестации по дисциплине или непрохождение промежуточной аттестации в установленные сроки признаются академической задолженностью. Студенты обязаны ликвидировать академическую задолженность.

Виды и сроки проведения мероприятий промежуточной аттестации устанавливаются рабочей программой учебной дисциплины.

13. Методические указания для обучающихся по освоению дисциплины

Подготовка к лекциям

Главное в период подготовки к лекционным занятиям - научиться методам самостоятельного умственного труда, сознательно развивать свои творческие способности и овладевать навыками творческой работы. Для этого необходимо строго соблюдать дисциплину учебы и поведения. Четкое планирование своего рабочего времени и отдыха является необходимым условием для успешной самостоятельной работы.

В основу его нужно положить рабочие программы изучаемых в семестре дисциплин. Ежедневной учебной работе студенту следует уделять 9-10 часов своего времени, т.е. при шести часах аудиторных занятий самостоятельной работе необходимо отводить 3-4 часа.

Самостоятельная работа на лекции

Слушание и запись лекций - сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом.

Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Конспект лекции лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, определения, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п.

Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор.

Подготовка к практическим занятиям

Подготовку к каждому практическому занятию каждый студент должен начать с ознакомления с планом практического занятия, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованной к данной теме. На основе индивидуальных предпочтений студенту необходимо самостоятельно выбрать тему доклада по проблеме практического занятия и по возможности подготовить по нему презентацию. Если программой дисциплины предусмотрено выполнение практического

задания, то его необходимо выполнить с учетом предложенной инструкции (устно или письменно). Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы практического занятия, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

Структура занятия В зависимости от содержания и количества отведенного времени на изучение каждой темы практическое занятие может состоять из четырех-пяти частей:

1. Обсуждение теоретических вопросов, определенных программой дисциплины.
2. Доклад и/ или выступление с презентациями по проблеме практического занятия.
3. Обсуждение выступлений по теме - дискуссия.
4. Выполнение практического задания с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома, если это предусмотрено программой.
5. Подведение итогов занятия.

Первая часть - обсуждение теоретических вопросов - проводится в виде фронтальной беседы со всей группой и включает выборочную проверку преподавателем теоретических знаний студентов. Примерная продолжительность — до 15 минут.

Вторая часть — выступление студентов с докладами, которые должны сопровождаться презентациями с целью усиления наглядности восприятия, по одному из вопросов практического занятия. Примерная продолжительность — 20-25 минут.

После докладов следует их обсуждение - дискуссия. В ходе этого этапа практического занятия могут быть заданы уточняющие вопросы к докладчикам. Примерная продолжительность - до 15-20 минут.

Если программой предусмотрено выполнение практического задания в рамках конкретной темы, то преподавателями определяется его содержание и дается время на его выполнение, а затем идет обсуждение результатов. Если практическое задание должно было быть выполнено дома, то на практическом занятии преподаватель проверяет его выполнение (устно или письменно). Примерная продолжительность - 15-20 минут.

Подведением итогов заканчивается практическое занятие. Студентам должны быть объявлены оценки за работу и даны их четкие обоснования. Примерная продолжительность — 5 минут.

Работа с литературными источниками В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками,

учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме практического или практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

Подготовка презентации и доклада

Для подготовки презентации рекомендуется использовать: PowerPoint, MS Word, Acrobat Reader. Самая простая программа для создания презентаций - Microsoft PowerPoint.

Для подготовки презентации необходимо собрать и обработать начальную информацию. Последовательность подготовки презентации:

1. Четко сформулировать цель презентации: вы хотите свою аудиторию мотивировать, убедить, заразить какой-то идеей или просто формально отчитаться.
2. Определить каков будет формат презентации: живое выступление (тогда, сколько будет его продолжительность) или электронная рассылка (каков будет контекст презентации).
3. Отобрать всю содержательную часть для презентации и выстроить логическую цепочку представления.
4. Определить ключевые моменты в содержании текста и выделить их.
5. Определить виды визуализации (картинки) для отображения их на слайдах в соответствии с логикой, целью и спецификой материала.
6. Подобрать дизайн и форматировать слайды (количество картинок и текста, их расположение, цвет и размер).
7. Проверить визуальное восприятие презентации.

К видам визуализации относятся иллюстрации, образы, диаграммы, таблицы. **Иллюстрация** - представление реально существующего зрительного ряда. **Образы** - в отличие от иллюстраций - метафора. Их назначение - вызвать эмоцию и создать отношение к ней, воздействовать на аудиторию. С помощью хорошо продуманных и представляемых образов, информация может надолго остаться в памяти человека. **Диаграмма** - визуализация количественных и качественных связей. Их используют для убедительной демонстрации данных, для пространственного мышления в дополнение к логическому. **Таблица** - конкретный, наглядный и точный показ данных. Ее основное назначение - структурировать информацию, что порой облегчает восприятие данных аудиторией.

Практические советы по подготовке презентации

- готовьте отдельно: печатный текст + слайды + раздаточный материал;
- слайды - визуальная подача информации, которая должна содержать минимум текста, максимум изображений, несущих смысловую нагрузку, выглядеть наглядно и просто;
- текстовое содержание презентации - устная речь или чтение, которая должна включать аргументы, факты, доказательства и эмоции;
- обязательная информация для презентации: тема, фамилия и инициалы выступающего; план сообщения; краткие выводы из всего сказанного; список использованных источников;
- раздаточный материал - должен обеспечивать ту же глубину и охват, что и живое выступление: люди больше доверяют тому, что они могут унести с собой, чем исчезающим изображениям, слова и слайды забываются, а раздаточный материал остается постоянным осязаемым напоминанием; раздаточный материал важно раздавать в конце презентации; раздаточный материалы должны отличаться от слайдов, должны быть более информативными.

Тема доклада должна быть согласованна с преподавателем и соответствовать теме учебного занятия. Материалы при его подготовке, должны соответствовать научно-методическим требованиям вуза и быть указаны в докладе. Необходимо соблюдать регламент, оговоренный при получении задания. Иллюстрации должны быть достаточными, но не чрезмерными.

Работа студента над докладом-презентацией включает отработку умения самостоятельно обобщать материал и делать выводы в заключении, умения ориентироваться в материале и отвечать на дополнительные вопросы слушателей, отработку навыков ораторства, умения проводить диспут.

Докладчики должны знать и уметь: сообщать новую информацию; использовать технические средства; хорошо ориентироваться в теме всего практического занятия; дискутировать и быстро отвечать на заданные вопросы; четко выполнять установленный регламент (не более 10 минут); иметь представление о композиционной структуре доклада и др.

Структура выступления

Вступление помогает обеспечить успех выступления по любой тематике. Вступление должно содержать: название, сообщение основной идеи, современную оценку предмета изложения, краткое перечисление рассматриваемых вопросов, живую интересную форму изложения, акцентирование внимания на важных моментах, оригинальность подхода.

Основная часть, в которой выступающий должен глубоко раскрыть суть затронутой темы, обычно строится по принципу отчета. Задача основной части - представить достаточно данных для того, чтобы слушатели заинтересовались темой и захотели ознакомиться с материалами. При этом логическая структура теоретического блока не должны даваться без наглядных пособий, аудио-визуальных и визуальных материалов.

Заключение - ясное, четкое обобщение и краткие выводы, которых всегда ждут слушатели.

Подготовка реферата

Реферат - письменный доклад по определенной теме, в котором собрана информация из одного или нескольких источников. Рефераты пишутся обычно стандартным языком, с использованием типологизированных речевых оборотов вроде: «важное значение имеет», «уделяется особое внимание», «поднимается вопрос», «делаем следующие выводы», «исследуемая проблема», «освещаемый вопрос» и т.п.

К языковым и стилистическим особенностям рефератов относятся слова и обороты речи, носящие обобщающий характер, словесные клише. У рефератов особая логичность подачи материала и изъяснения мысли, определенная объективность изложения материала.

Реферат не копирует дословно содержание первоисточника, а представляет собой новый вторичный текст, создаваемый в результате систематизации и обобщения материала первоисточника, его аналитико-синтетической переработки.

Будучи вторичным текстом, реферат составляется в соответствии со всеми требованиями, предъявляемыми к связанному высказыванию: так ему присущи следующие категории: оптимальное соотношение и завершенность (смысловая и жанрово-композиционная). Для реферата отбирается информация, объективно-ценная для всех читающих, а не только для одного автора. Автор реферата не может пользоваться только ему понятными значками, пометами, сокращениями.

Работа, проводимая автором для подготовки реферата должна обязательно включать самостоятельное мини-исследование, осуществляемое студентом на материале или художественных текстов по литературе, или архивных первоисточников по истории и т.п.

Организация и описание исследования представляет собой очень сложный вид интеллектуальной деятельности, требующий культуры научного мышления, знания методики проведения исследования, навыков оформления научного труда и т.д. Мини-исследование раскрывается в реферате после глубокого, полного обзора научной литературы по проблеме исследования.

В зависимости от количества реферируемых источников выделяют следующие виды рефератов:

- **монографические** - рефераты, написанные на основе одного источника;
- **обзорные** - рефераты, созданные на основе нескольких исходных текстов, объединенных общей темой и сходными проблемами исследования.

Структура реферата

1. Титульный лист
2. Оглавление

3. Введение
4. Основная часть
5. Заключение
6. Список использованной литературы
7. Приложения

Подготовка эссе

Эссе - вид самостоятельной исследовательской работы студентов, с целью углубления и закрепления теоретических знаний и освоения практических навыков. Цель эссе состоит в развитии самостоятельного творческого мышления и письменного изложения собственных мыслей.

В зависимости от темы формы эссе могут быть различными. Это может быть анализ имеющихся статистических данных по изучаемой проблеме, анализ материалов из средств массовой информации и подробный разбор проблемной ситуации с развернутыми мнениями, подбором и детальным анализом примеров, иллюстрирующих проблему и т.п.

В процессе выполнения эссе студенту предстоит выполнить следующие виды работ: составить план эссе; отобрать источники, собрать и проанализировать информацию по проблеме; систематизировать и проанализировать собранную информацию по проблеме; представить проведенный анализ с собственными выводами и предложениями.

Эссе выполняется студентом под руководством преподавателя кафедры самостоятельно. Тему эссе студент выбирает из предлагаемого примерного перечня и для каждого студента она должна быть индивидуальной (темы в одной группе совпадать не могут).

Структура эссе

1. Титульный лист.
2. План.
3. Введение с обоснованием выбора темы.
4. Текстовое изложение материала (основная часть).
5. Заключение с выводами по всей работе.
6. Список использованной литературы.

Титульный лист является первой страницей и заполняется по строго определенным правилам.

Введение (вводная часть) - суть и обоснование выбора данной темы, состоит из ряда компонентов, связанных логически и стилистически. На этом этапе очень важно правильно сформулировать вопрос, на который Вы собираетесь найти ответ в ходе своего исследования. При работе над введением могут помочь ответы на следующие вопросы:

1. Надо ли давать определения терминам, прозвучавшим в теме эссе?
2. Почему тема, которую я раскрываю, является важной в настоящий момент?
3. Какие понятия будут вовлечены в мои рассуждения по теме?

4. Могу ли я разделить тему на несколько составных частей?

Таким образом, в водной части автор определяет проблему и показывает умение выявлять причинно-следственные связи, отражая их в методологии решения поставленной проблемы через систему целей, задач и т.д.

Текстовое изложение материала (основная часть) - теоретические основы выбранной проблемы и изложение основного вопроса. Данная часть предполагает развитие аргументации и анализа, а также обоснование их, исходя из имеющихся данных, других аргументов и позиций по этому вопросу. В этом заключается основное содержание эссе и это представляет главную трудность при его написании. Поэтому большое значение имеют подзаголовки, на основе которых осуществляется выстраивание аргументации; именно здесь необходимо обосновать (логически, используя данные и строгие рассуждения) предлагаемую аргументацию/анализ. В качестве аналитического инструмента можно использовать графики, диаграммы и таблицы там, где это необходимо. Традиционно в научном познании анализ может проводиться с использованием следующих категорий: причина - следствие, общее - особенное, форма - содержание, часть - целое, постоянство - изменчивость.

В процессе построения эссе надо помнить, что один параграф должен содержать только одно утверждение и соответствующее доказательство, подкрепленное графическим или иллюстративным материалом. Следовательно, наполняя разделы содержанием аргументации (а это должно найти отражение в подзаголовках), в пределах параграфа необходимо ограничить себя рассмотрением одной главной мысли.

Хорошо проверенный способ построения любого эссе - использование подзаголовков для обозначения ключевых моментов аргументированного изложения: это помогает посмотреть на то, что предполагается сделать и ответить на вопрос, хорош ли замысел. При этом последовательность подзаголовков свидетельствует также о наличии или отсутствии логики в освещении темы эссе.

Таким образом, основная часть - рассуждение и аргументация, В этой части необходимо представить релевантные теме концепции, суждения и точки зрения, привести основные аргументы "за" и "против" них, сформулировать свою позицию и аргументировать ее.

Заключение (заключительная часть) - обобщения и аргументированные выводы по теме эссе с указанием области ее применения и т.д. Оно подытоживает эссе или еще раз вносит пояснения, подкрепляет смысл и значение изложенного в основной части. Методы, рекомендуемые для составления заключения: повторение, иллюстрация, цитата, утверждение. Заключение может содержать такой очень важный, дополняющий эссе элемент, как указание на применение исследования, не исключая взаимосвязи с другими проблемами.

Таким образом, в заключительной части эссе должны быть сформулированы выводы и определено их приложение к практической области деятельности.

Список использованной литературы составляет одну из частей работы, отражающей самостоятельную творческую работу автора и позволяющей судить о степени фундаментальности данной работы. При составлении списка литературы в перечень включаются только те источники, которые действительно были использованы при подготовке эссе. Список использованной литературы составляется строго в алфавитном порядке в следующей последовательности: законы РФ и другие официальные материалы (указы, постановления, решения министерств и ведомств); печатные работы (книги, монографии, сборники); периодика; Интернет-сайты. По возможности список должен содержать современную литературу по теме. Общее оформление списка использованной литературы для эссе аналогично оформлению списка использованной литературы для реферата.

Приложения могут включать иллюстративный материал (схемы, диаграммы, рисунки, таблицы и др.). При этом приложения являются продолжением самой работы, т.е. на них продолжается сквозная нумерация, но в общем объеме эссе они не учитываются.