МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет Энергетический

Кафедра Прикладной информатики и математики

**УЧЕБНЫЕ МАТЕРИАЛЫ**

**для студентов заочной формы обучения**

*(с полным сроком обучения)*

по дисциплине **ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

для направления подготовки 09.03.03 Прикладная информатика

Профиль:прикладная информатика в экономике

Общая трудоемкость дисциплины – 9 зачетных единиц

1 семестр – 4 ЗЕ

2 семестр – 5 ЗЕ

Форма текущего контроля в 1 семестре – контрольная работа

во2 семестре – контрольная работа

Курсовая работа (курсовой проект) (КР, КП) –нет.

Форма промежуточного контроля в 1 семестре –экзамен

во2 семестре– экзамен

**Краткое содержание курса**

Перечень изучаемых тем и разделов дисциплины

Раздел 1. Основы передачи информации и аппаратные средства

Тема 1.1Теоретические основы передачи сообщений

Тема 1.2 Аппаратное и программное обеспечение с точки зрения пользователя

Тема 1.3 Безопасность установки и эксплуатации компьютерного оборудования.

Раздел 2. Программирование вычислительных процессов на алгоритмическом языке Паскаль

Тема 2.1 Основные понятия языка

Тема 2.2 Управляющие операторы языка

Тема 2.3 Типы данных, определяемые программистом

Раздел 3. Технологии структурного программирования

Тема 3.1. Файлы

Тема 3.2. Модульное программирование

Тема 3.3. Технология структурного программирования

Раздел 4. Работа с динамической памятью

Тема 4.2. Динамические структуры

Тема 4.1. Динамические переменные

данных.

Раздел 5. Объектно – ориентированное программирование

Тема 5.1. Основы объектно - ориентированного программирования.

Тема 5.2. Программирование под Windows

**Семестр 1**

**Форма текущего контроля: контрольная работа**

**КОНТРОЛЬНАЯ РАБОТА №1**

**Указания к выполнению работы**

Контрольная работа включает в себя **три** задания. Два теоретических вопроса и набор из четырех задач.

Вариант выбирается по **последней цифре** в номере зачетной книжки.

Работа (1 и 2-е задания )оформляется в печатном виде в соответствии с «Требованиями к оформлению». Скачать требования можно по ссылке:

<http://zabgu.ru/files/html_document/pdf_files/fixed/Normativny'e_dokumenty'/Obshhie_trebovaniya_k_postroeniyu_i_oformleniyu_uchebnoj_tekstovoj_dokumentacii.pdf>

Пример титульного листа в *Приложении 1*

Ответ на очередной вопрос должен начинаться с плана, который представляет собой перечень рассматриваемых в данном пункте вопросов.

Очередное задание должно начинаться с новой страницы.

Объём ответа на задания №1 и №2 не менее 10-ти и не более 20-ти печатных страниц.

В качестве ответа на вопрос №3 нужно привести листинги задач.

Для проверки программ можно использовать транслятор ABCPascal(скачать по адресу <http://pascalabc.net>)

В конце контрольной работы приводится список использованной литературы и/или Интернет – источников.

Контрольную работу необходимо прислать на проверку по адресу [Diplom\_PI@mail.ru](mailto:Diplom_PI@mail.ru)

После проверки контрольной работы преподавателем, и устранения всех указанных недочетов во время сессии по печатной версии работы проводится собеседование и выставляется оценка.

Для успешного усвоения необходимых тем можно использовать прилагаемые к указанию дополнительные материалы (см. в отдельных файлах).

**ВНИМАНИЕ!**

1. Для допуска к сдаче экзамена необходимо защитить контрольную работу.
2. Последний день приема контрольных работ – второй учебный день занятий во время сессии.
3. Контактная информация: тел. кафедры ПИМ 41-73-12

E-mail: **kafedra\_pim@mail.ru**(письмо с темой «для Яковлевой Л.Л.») или [Diplom\_PI@mail.ru](mailto:Diplom_PI@mail.ru)

**ЗАДАНИЯ ПО ВАРИАНТАМ**

**Вариант № 0**

1. Структура программного обеспечения современного ПК.
2. ЯП ВУ[[1]](#footnote-1) Pascal: реализация подпрограмм. Процедуры и функции. Программирование рекурсивных алгоритмов.
3. Задачи: 10, 11,21,31.

**Вариант №1**

1. Алгоритмизация и программирование. Основные парадигмы современного программирования.
2. ЯП ВУ Pascal: структура программы, переменные, выражения, операторы ввода – вывода. Оператор присваивания. Стандартные типы данных. Линейные программы.
3. Задачи: 9, 12, 22, 32.

**Вариант №2**

1. Основные этапы компьютерного решения задач.
2. ЯП ВУ Pascal: Модульное программирование: подпрограммы: процедуры и функции. Модули: описание, использование. Стандартные модули.
3. Задачи: 8, 20, 30, 40.

**Вариант №3**

1. Алгоритм: понятие, свойства, способы записи. Основные алгоритмические конструкции.
2. ЯП ВУ Pascal: реализация циклических вычислений (цикл For, циклы Whileи Repeat). Особенности, примеры использования при решении задач.
3. Задачи: 7, 13, 23, 33.

**Вариант №4**

1. Типы данных в ЯП ВУ Pascal. Классификация типов. Преобразование типов. Совместимость типов. Особенности различных типов.
2. ЯП ВУ Pascal: реализация ветвлений: условный оператор (if…then…else), оператор варианта Case. Особенности, примеры использования при решении задач.
3. Задачи: 6, 18, 28, 38.

**Вариант №5**

1. Информация: понятие, свойства, подходы к измерению количества (статистический, алфавитный, семантический).
2. ЯП ВУ Pascal: массивы. Виды: одномерные и двумерные. Способы описания. Обработка элементов. Примеры использования при решении задач.
3. Задачи: 5, 14, 24, 34.

**Вариант №6**

1. ЯП ВУ Pascal: типы данных определяемые программистом. Интервальный, перечисляемый. Примеры использования при решении задач.
2. Критерии качества программы. Этапы создания структурной программы.
3. Задачи: 4, 15, 25, 35.

**Вариант №7**

1. ЯП ВУ Pascal: сложный тип данных - строки. Описание. Операции со строками (склеивание, сравнение, удаление, вставка, копирование, определение длины, выделение подстроки). Примеры использования при решении задач.
2. Отладка и тестирование программ. Этапы тестирования. Виды тестирования. Требования к наборам тестовых данных.
3. Задачи: 3, 19, 29, 39.

**Вариант №8**

1. ЯП ВУ Pascal: комбинированный тип данных (записи). Описание. Доступ к полям. Записи с вариантной частью. Примеры использования при решении задач.
2. Структурное программирование.
3. Задачи: 2, 16, 26, 36.

**Вариант №9**

1. ЯП ВУ Pascal: файловый тип данных. Описание. Операции для работы с файлами. Подпрограммы для работы со всеми типами файлов. Текстовые файлы. Бестиповые, компонентные. Прямой доступ. Примеры использования при решении задач.
2. Основы защиты информации.
3. Задачи: 1, 17, 27, 37.

**Задачи к контрольной работе № 1**

**Ветвления**

1. Составить программу вычисления функции :
2. Даны целочисленные координаты точки на плоскости. Если точка не лежит на координатных осях, то вывести 0. Если точка совпадает с началом координат, то вывести 1. Если точка не совпадает с началом координат, но лежит на оси OX или OY, то вывести соответственно 2 или 3.
3. Даны вещественные координаты точки, не лежащей на координатных осях OX и OY. Вывести номер координатной четверти, в которой находится данная точка.
4. Дан номер некоторого года (положительное целое число). Вывести соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.
5. Для данного x вычислить значение следующей функции f, принимающей значения целого типа:

|  |  |  |
| --- | --- | --- |
|  | 0, | если x < 0, |
| f(x)= | 1, | если x принадлежит [0,1), [2,3), ... , |
|  | –1, | если x принадлежит [1,2), [3,4), ... . |

1. Дано целое число, лежащее в диапазоне от –999 до 999. Вывести строку — словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.
2. Дано целое число, лежащее в диапазоне от 1 до 9999. Вывести строку — словесное описание данного числа вида "четное двузначное число", "нечетное четырехзначное число" и т.д.
3. Написать программу решения квадратного алгебраического уравнения с произвольными параметрами: A\*X\*X+B\*X+C=0.
4. Написать программу, которая требует ввода времени дня и, в зависимости от введенного значения, желает доброго утра, доброго дня, доброго вечера или спокойной ночи.
5. Известно расстояние до каждого из трех сел от районного центра. Определить, какое из них находится ближе всего к районному центру, а какое дальше всего от него.

**Циклы**

1. Начав тренировки, спортсмен в первый день пробежал 10км. Каждый следующий день он увеличивал дневную норму на 10% отнормы предыдущего дня. Какой суммарный путь пробежит спортсмен за7 дней.
2. Одноклеточная амеба каждые 3 часа делится на 2 клетки.Определить, сколько клеток будет через 3, 6, 9, 12,...24 часа.Попробовать изобразить это графически.
3. Написать программу угадывания (целых) чисел. Один изпартнеров вводит в программу число, а второй должен отгадать эточисло. Причем на каждое предлагаемый вариант программа отвечаетлибо "больше", либо "меньше" до тех пор, пока число не будетотгадано.
4. В течение двух недель человек заключает пари на скачках.В первый день он делает ставку в 1 фунт и проигрывает. Во второйдень, чтобы возместить свой проигрыш, он делает ставку в 2 фунта,но снова проигрывает. На третий день он ставит 3 фунта, однакоопять проигрывает. Он упрямо продолжает увеличивать на 1 фунтсвою ставку, но продолжает проигрывать. Сколько денег проиграетэтот человек через 2 недели?
5. Припишите к 1022 слева и справа по одной цифре так,чтобы полученное число делилось на 7, 8 и 9.
6. В десятичной записи числа 42\*4\* две цифры пропущены.Восстановите их, если известно, что число кратно 72.
7. В 1202 году итальянский математик Леонард Пизанский (Фибоначчи) предложил такую задачу: пара кроликов каждый месяц дает приплод – двух кроликов (самца и самку), от которых через два месяца уже получается новый приплод. Сколько кроликов будет через год, если в начале года имелась одна пара? Согласно условию задачи числа, соответствующие количеству кроликов, которые появляются через месяц, составляют последовательность 1, 1, 2, 3, 5, 8, 13, 21, 34, … Составить программу, позволяющую найти все числа Фибоначчи, меньшие заданного N.
8. В старояпонском календаре был принят 60-летний цикл, состоявший из 12-летних подциклов. Подциклы обозначались названиями цвета: зеленый, красный, желтый, белый, черный. Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. 1984 год – год зеленой крысы – был годом начала очередного цикла. Напишите программу, которая вводит номер некоторого года и печатает его название по старояпонскому календарю.
9. Натуральное число из nцифр называется числом Армстронга, если сумма его цифр, возведенных в n -ю степень, равна самому числу. Например, 153=13+53+33. Найти все числаАрмстронга, состоящие из двух, трех и четырех цифр.
10. Автоморфным называется число, содержащее в последних разрядах свой квадрат: 5\*5=25; 25\*25=625.Дано число n. Выяснить является ли оно автоморфным.

**Массивы**

Внимание!

Массив должен хранится в текстовом файле. Размерности массивов следует задать именованными константами. Все необходимые данные должны передаваться подпрограммам в качестве параметров; все величины, используемые только внутри подпрограмм, должны быть описаны как локальные. Использование глобальных переменных в подпрограммах не допускается. Вывод результатов работы подпрограмм должен выполняться в главной программе.

1. Элемент матрицы называется локальным минимумом, если он строго меньшевсех имеющихся у него соседей (определение соседних элементов см. в варианте 9). Подсчитать количество локальных минимумов заданной матрицыразмером 10x10 (оформить в виде процедуры).
2. Найти сумму модулей элементов, расположенных выше главной диагонали(оформить в виде функции).
3. Коэффициенты системы линейных уравнений заданы в виде прямоугольнойматрицы. С помощью допустимых преобразований привести систему к треугольному виду (оформить в виде процедуры).
4. Найти количество строк, среднее арифметическое элементов которых меньшезаданной величины (оформить в виде функции).
5. Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненныенулями (оформить в виде процедуры).
6. Найти номер первой из строк, содержащих хотя бы один положительный элемент (оформить в виде функции).
7. Осуществить циклический сдвиг элементов прямоугольной матрицы на *п* элементов вправо или вниз (в зависимости от введенного режима), *п* может быть больше количества элементов в строке или столбце (оформить в виде процедуры).
8. Осуществить циклический сдвиг элементов квадратной матрицы размером М х N вправо на *k*элементов таким образом: элементы первой строки сдвигаются в последний столбец сверху вниз, из него — в последнюю строку справа налево, из нее — в первый столбец снизу вверх, из него — в первую строку; для остальных элементов — аналогично (оформить в виде процедуры).
9. Дана целочисленная прямоугольная матрица. Определить номер первогоиз столбцов, содержащих хотя бы один нулевой элемент (оформить в видефункции).
10. Характеристикой строки целочисленной матрицы назовем сумму ее отрицательных четных элементов. Переставляя строки заданной матрицы, расположитьих в соответствии с убыванием характеристик (оформить в виде процедуры).
11. Упорядочить строки целочисленной прямоугольной матрицы по возрастанию количества одинаковых элементов в каждой строке (оформить в видепроцедуры).
12. Найти номер первого из столбцов, не содержащих ни одного отрицательногоэлемента (оформить в виде функции).
13. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине — в позиции (2, 2), следующий по величине — в позиции (3, 3) и т. д., заполнив таким образом всю главную диагональ (оформить в виде процедуры).
14. Найти номер первой из строк, не содержащих ни одного положительного элемента (оформить в виде функции).
15. Дана целочисленная прямоугольная матрица. Определитьколичество строк, содержащих хотя бы один нулевой элемент (оформить в видефункции);
16. Дана целочисленная квадратная матрица. Определитьсумму элементов в тех строках, которые не содержат отрицательных элементов (оформить в виде функции).
17. Дана целочисленная квадратная матрица. Определитьминимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы (оформить в виде процедуры).
18. Матрица А имеет седловую точку Aij, если Aij является минимальным элементом в i-й строке и максимальным в j-омстолбце.
19. Дана целочисленная прямоугольная матрица. Определитьколичество отрицательных элементов в тех строках, которые содержат хотябы один нулевой элемент (оформить в виде функции).
20. Дана целочисленная прямоугольная матрица. Определитьномера строк и столбцов всех седловых точек матрицы (оформить в виде процедуры).

**Форма промежуточного контроля**

**1 семестр - экзамен**

**ВОПРОСЫ К ЭКЗАМЕНУ**

**(1 семестр)**

1. Понятие – информация. Информатизация и компьютеризация общества. Свойства информации. Единицы измерения информации.
2. Информатика как предметная область. Основные направления информатики.
3. История развития информатики и вычислительной техники.
4. Структура программного обеспечения (системное, инструментальное, прикладное)
5. Особенности распространения и установки ПО.
6. Понятие и свойства алгоритмов. Способы записи алгоритмов.
7. Основные алгоритмические конструкции.
8. Языки программирования: понятие, уровни языков, виды трансляторов.
9. Классификация языков программирования.
10. Современные парадигмы программирования.
11. Язык программирования Паскаль: история возникновения и развития, структура программы, операторы ввода-вывода, арифметические операции, стандартные функции.
12. Типы данных в Паскале и операции над данными различных типов.
13. Реализация разветвляющихся вычислительных процессов в ЯП Паскаль (полная и сокращенная форма оператора IF, вложенные условия, оператор CASE). Примеры.
14. Организация циклических вычислений. Циклы: For,While,Repeat. Примеры
15. Этапы решения задачи на ЭВМ (технология подготовки и решения задач с применением ЭВМ).
16. Отладка и тестирование алгоритмов и программ. Этапы тестирования. Требования к тестовым данным. Примеры.
17. Типы данных определяемые пользователем.Перечисляемый и ограниченный типы в Pascale.
18. Одномерные и двумерные массивы: способы описания, ввод-вывод элементов, типовые операции по обработке элементов.
19. Файловый тип данных: технология работы с файлами. Примеры.
20. Реализация процедур и функций, локальные и глобальные переменные, виды параметров подпрограмм.
21. Модульное программирование: описание и использование модулей, стандартные модули в ЯП Паскаль.
22. Понятие рекурсии, виды и примеры рекурсивных алгоритмов.
23. Строки: назначение, описание, операции над строками, процедуры и функции для работы со строками.
24. Множества:назначение, описание, операции над множествами.
25. Записи: назначение, описание, работа с записями.
26. Технология структурного программирования: критерии качества программы.
27. Технология структурного программирования: этапы создания структурной программы

**Семестр 2**

**Форма текущего контроля: контрольная работа**

**КОНТРОЛЬНАЯ РАБОТА № 2**

**Указания к выполнению работы**

Контрольная работа включает в себя **два** задания. Один теоретический вопрос и набор из 2 х задач.

Вариант выбирается по **последней цифре** в номере зачетной книжки.

Работа оформляется в печатном виде в соответствии с «Требованиями к оформлению». Скачать требования можно по ссылке:

<http://zabgu.ru/files/html_document/pdf_files/fixed/Normativny'e_dokumenty'/Obshhie_trebovaniya_k_postroeniyu_i_oformleniyu_uchebnoj_tekstovoj_dokumentacii.pdf>

Пример титульного листа в *Приложении 2*

Ответ на вопрос должен начинаться с плана, который представляет собой перечень рассматриваемых в данном пункте вопросов.

В качестве ответа на вопрос №2 нужно привести листинги задач.

Для проверки программ можно использовать транслятор ABCPascal (скачать по адресу <http://pascalabc.net>) и IDE Delphi (бесплатную версию (на 30 дней) можно скачать по адресу <https://www.embarcadero.com/ru/products/delphi/downloads> )

В конце контрольной работы приводится список использованной литературы и/или Интернет – источников.

Контрольную работу необходимо прислать на проверку по адресу [Diplom\_PI@mail.ru](mailto:Diplom_PI@mail.ru)

После проверки контрольной работы преподавателем, и устранения всех указанных недочетов во время сессии по печатной версии работы проводится собеседование и выставляется оценка.

**ВНИМАНИЕ!**

1. Для допуска к сдаче экзамена необходимо защитить контрольную работу.
2. Последний день приема контрольных работ – второй учебный день занятий во время сессии.
3. Контактная информация: тел. кафедры ПИМ 41-73-12 E-mail: [Diplom\_PI@mail.ru](mailto:Diplom_PI@mail.ru)

**ЗАДАНИЯ ПО ВАРИАНТАМ**

**Вариант № 0**

1. Динамические структуры данных. Основные сведения о ссылочном типе данных (указателях). Динамические массивы.

2. Задачи: 10, 20

**Вариант №1**

1. Динамические структуры данных. Массивы. Линейные списки.

2. Задачи: 1,11

**Вариант №2**

1. Динамические структуры данных. Массивы. Стеки.

Задачи: 2,12

**Вариант №3**

1. Динамические структуры данных. Массивы. Очереди.

2. Задачи: 3,13

**Вариант №4**

1. Динамические структуры данных. Массивы Деревья.

2. Задачи: 4,14

**Вариант №5**

1. Объектно-ориентированное программирование (ООП): основные понятия.

2. Задачи: 5,15

**Вариант №6**

1. Объектно-ориентированное программирование (ООП): инкапсуляция.

2. Задачи: 6,16

**Вариант №7**

1. Объектно-ориентированное программирование (ООП): наследование.

2. Задачи: 7,17

**Вариант №8**

1. Объектно-ориентированное программирование (ООП): полиморфизм.

2. Задачи: 8,18

**Вариант №9**

1. Объектно-ориентированное программирование (ООП): общие правила создания объектов, их свойств и методов.

2. Задачи: 9,19

**Задачи к контрольной работе № 2**

* + - 1. Дан указатель P1 на вершину стека (если стек пуст, то P1 = nil). Извлечь из стека все элементы и вывести их значения. Вывести также количество извлеченных элементов N (для пустого стека вывести 0). После извлечения элементов из стека освобождать память, которую они занимали.
      2. Даны указатели P1 и P2 на вершины двух непустых стеков. Переместить все элементы из первого стека во второй (в результате элементы первого стека будут располагаться во втором стеке в порядке, обратном исходному) и вывести адрес новой вершины второго стека. Операции выделения и освобождения памяти не использовать.
      3. Дан набор из 10 чисел. Создать очередь, содержащую данные числа в указанном порядке (первое число будет размещаться в начале очереди, последнее — в конце), и вывести указатели P1 и P2 на начало и конец очереди.
      4. Дан набор из 10 чисел. Создать две очереди: первая должна содержать числа из исходного набора с нечетными номерами (1, 3, …, 9), а вторая — с четными (2, 4, …, 10); порядок чисел в каждой очереди должен совпадать с порядком чисел в исходном наборе. Вывести указатели на начало и конец первой, а затем второй очереди.
      5. Дан набор из 10 чисел. Создать две очереди: первая должна содержать все нечетные, а вторая — все четные числа из исходного набора (порядок чисел в каждой очереди должен совпадать с порядком чисел в исходном наборе). Вывести указатели на начало и конец первой, а затем второй очереди (одна из очередей может оказаться пустой; в этом случае вывести для нее две константы nil).
      6. Дано число D и указатели P1 и P2 на начало и конец очереди (если очередь является пустой, то P1 = P2 = nil). Добавить элемент со значением D в конец очереди и вывести новые адреса начала и конца очереди.
      7. Дано число D и указатели P1 и P2 на начало и конец очереди, содержащей не менее двух элементов. Добавить элемент со значением D в конец очереди и извлечь из очереди первый (начальный) элемент. Вывести значение извлеченного элемента и новые адреса начала и конца очереди. После извлечения элемента из очереди освободить память, занимаемую этим элементом.
      8. Даны указатели P1 и P2 на вершины двух непустых стеков. Перемещать элементы из первого стека во второй, пока значение вершины первого стека не станет четным (перемещенные элементы первого стека будут располагаться во втором стеке в порядке, обратном исходному). Если в первом стеке нет элементов с четными значениями, то переместить из первого стека во второй все элементы. Вывести адреса новых вершин первого и второго стека (если первый стек окажется пустым, то вывести для него константу nil). Операции выделения и освобождения памяти не использовать.
      9. Дан указатель P1 на вершину непустого стека. Создать два новых стека, переместив в первый из них все элементы исходного стека с четными значениями, а во второй — с нечетными (элементы в новых стеках будут располагаться в порядке, обратном исходному; один из этих стеков может оказаться пустым). Вывести адреса вершин полученных стеков (для пустого стека вывести nil). Операции выделения и освобождения памяти не использовать.
      10. Дан указатель P1 на вершину стека (если стек пуст, то P1 = nil). Также дано число N (> 0) и набор из N чисел. Описать тип TStack — запись с одним полем Top типа PNode (поле указывает на вершину стека) — и процедуру Push(S, D), которая добавляет в стек S новый элемент со значением D (S — входной и выходной параметр типа TStack, D — входной параметр целого типа). С помощью процедуры Push добавить в исходный стек данный набор чисел (последнее число будет вершиной стека) и вывести адрес новой вершины стека.
      11. Создать меню с командами Size, Color, Paint, Quit.

При запуске приложения сделать недоступной команду Paint. При выборе команды. Quit завершать работу приложения. При выборе команды Size открывать окно диалога, содержащее:

* два поля ввода типа TEdit;
* группу из трех флажков (Red, Green, Blue) типа TCheckBox;
* кнопку типа TButton.

Обеспечить возможность:

* ввода длин сторон прямоугольника в пикселах в поля ввода;
* выбора его цвета с помощью флажков.

После задания параметров команда Paint должна стать доступной.

При выборе команды Paint в главном окне приложения нарисовать прямоугольник заданных размера и сочетания цветов или выдать сообщение, если заданные размеры превышают размер окна.

* + - 1. Создать меню с командами Input, Calc, About.

При выборе команды Input открывать окно диалога, содержащее:

* три поля ввода типа TEdit с метками Number 1, Number 2, Number 3;
* группу из двух флажков (Summa, Multipl.) типа TCheckBox;
* кнопку типа TButton.

Обеспечить возможность ввода трех чисел и выбора режима вычислений с помощью флажков, обеспечивающих подсчет суммы и/или произведения чисел. При выборе команды Calc открывать окно диалога с результатами.

* + - 1. Создать меню с командами Input color, Change, Exit, Help.

При выборе команды Exit завершать работу приложения. При выборе команды Input color открывать окно диалога, содержащее:

* три поля ввода типа TEdit с метками Red, Green, Blue;
* группу из двух флажков (Left, Right) типа TCheckBox;
* кнопку типа TButton.

Обеспечить возможность ввода RGB-составляющих цвета. При выборе команды Change изменять цвет фона главного окна на заданный (левая, правая или обе половины окна в зависимости от установленных флажков).

* + - 1. Создать меню с командами Input color, Change, Clear.

При выборе команды Input color открывать окно диалога, содержащее:

* список для выбора цвета типа TUstBox;
* группу из трех переключателей (Red, Green, Blue) типа TRadioGroup;
* кнопку типа TButton.

Обеспечить возможность:

* выбора цвета;
* ввода режима, определяющего закрашиваемую область (все окно, его верхняя или нижняя половина).
  + - 1. Создать меню с командами Input, Work, Exit.

При выборе команды Exit завершать работу приложения. При выборе команды Input открывать окно диалога, содержащее:

* три поля ввода типа TEdit с метками Radius, Height, Density;
* группу из двух флажков (Volume, Mass) типа TCheckBox;
* кнопку типа TButton.

Обеспечить возможность:

* ввода радиуса, высоты и плотности конуса;
* выбора режима с помощью флажков, обеспечивающих подсчет объема и/или массы конуса.

При выборе команды Work открывать окно сообщений с результатами.

* + - 1. Создать меню с командами Input, Calc, Draw, Exit.

При выборе команды ***Exit*** приложение должно завершать работу. При выборе команды ***Input*** открывать окно диалога, содержащее:

* поле ввода типа ***TEdit*** с меткой ***Radius***;
* CD группу из двух флажков (Square, Length) типа TCheckBox;
* кнопку типа TButton.

Обеспечить возможность:

* ввода радиуса окружности;
* выбора режима с помощью флажков, обеспечивающих подсчет площади круга и/или длины окружности.

При выборе команды ***Calc*** открывать окно сообщений с результатами. При выборе команды ***Draw*** в центре главного окна рисовать круг введенного радиуса или выдавать сообщение, что рисование невозможно (если диаметр превышает размеры рабочей области).

* + - 1. Создать меню с командами Begin, Help, About.

При выборе команды ***Begin*** открывать окно диалога, содержащее:

* поле ввода типа ***TEdit*** с меткой ***input***;
* метку типа ***TLabel*** для вывода результата;
* группу из трех переключателей (2, 16, 8) типа ***TRadioGroup***;
* две кнопки типа ***TButton*** — ***Do*** и ***ОК***.

Обеспечить возможность:

* ввода числа в поле ***input***;
* выбора режима преобразования с помощью переключателей, обеспечивающих перевод в двоичную, шестнадцатеричную или восьмеричную систему счисления.

При щелчке на кнопке ***Do*** выводить результат перевода.

* + - 1. Написать приложение Windows, которое выполняет анимацию изображения.

Создать меню с командами ***Show picture***, ***Choose***, ***Animate***, ***Stop***, ***Quit***.

При выборе команды ***Quit*** завершать работу приложения. При выборе команды ***Show*** ***picture*** рисовать в центре экрана объект, состоящий из нескольких графических примитивов. При выборе команды ***Choose*** открывать окно диалога, содержащее:

* поле ввода типа ***TEdit*** с меткой ***Speed*** для ввода скорости движения объекта;
* группу ***Direction*** из двух переключателей (***Up***-***Down***, ***Left***-***Right***) типа ***TRadloGroup*** для выбора направления движения;
* кнопку типа ***TButton***.

При выборе команды ***Animate*** перемещать объект в выбранном направлении до края окна и обратно с заданной скоростью, при выборе команды ***Stop*** — прекращать движение.

* + - 1. Написать приложение Windows, которое по заданным в файле исходным данным выводит информацию о портативных компьютерах.

Создать меню с командами ***Notebooks***, ***Your choice***, ***Quit***.

При запуске приложения сделать недоступной команду ***Your choice***. При выборе команды ***Quit*** завершать работу приложения.

При запуске приложения читать из файла исходные данные. Файл сформировать самостоятельно; каждая строка файла должна содержать тип компьютера, цену (***Price***) и емкость жесткого диска (***Hard drive***).

При выборе команды ***Show*** открывать окно диалога, содержащее:

* окно типа ***TStringGrid*** с введенной информацией;
* заголовки граф типа ***TLabel***;
* поле типа ***TEdit*** для ввода минимальной емкости диска;
* поле типа ***TEdit*** для ввода максимальной приемлемой цены;
* группу из двух переключателей (***Hard*** ***drive***, ***Price***) типа ***TRadioGroup***;
* кнопки ***OK*** и ***Cancel*** типа ***TBitBtn***.

После ввода всех данных сделать доступной команду ***Your*** ***choice***. При выборе команды ***Your choice*** открывать окно диалога, содержащее список компьютеров, удовлетворяющий введенным ограничениям и упорядоченный по отмеченной характеристике.

* + - 1. Написать приложение Windows, которое по заданным в файле исходным данным строит график или столбиковую диаграмму.

Создать меню с командами ***Input data***, ***Choose***, ***Line***, ***Bar***, ***Quit***.

При запуске приложения сделать недоступной команды ***Line*** и ***Ваг***. При выборе команды ***Quit*** завершать работу приложения. При выборе команды ***Input data*** читать из файла исходные данные (файл сформировать самостоятельно). При выборе команды ***Choose*** открывать окно диалога, содержащее:

список для выбора цвета графика типа ***TListBox***;

группу из двух переключателей (***Line***,-***Bar***) типа ***TRadioGroup***;

кнопку типа ***TButton***.

Обеспечить возможность ввода цвета и выбора режима: построение графика (***Line***) или столбиковой диаграммы (***Ваг***). После указания параметров сделать доступной соответствующую команду меню.

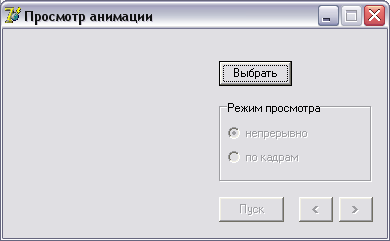
По команде ***Line*** или ***Ваг*** в главном окне приложения строить выбранным цветом график или диаграмму, причем окно должно содержать заголовок графика или диаграммы, наименование и градацию осей, а изображение — занимать все окно и масштабироваться при изменении размеров окна.

**Дополнительные задачи \*\*\***

Не обязательны для решения.

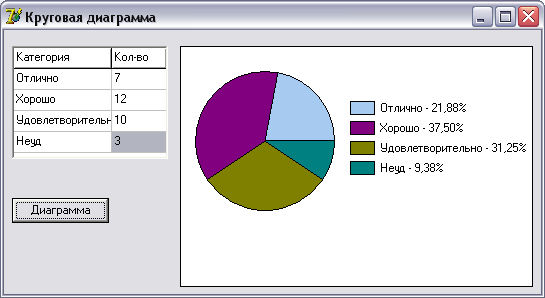
**Задача №1**

Напишите программу, используя которую можно просмотреть видеоклип. Клип должен воспроизводиться в диалоговом окне программы. Для выбора клипа (AVI-файла) используйте стандартное диалоговое окно Открытие файла. Рекомендуемый вид формы программы приведен на рис.

****

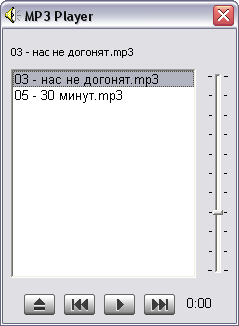
**Задача №2**

Напишите программу, которая строит диаграмму, отображающую результаты экзамена. Для ввода исходных данных используйте компонент stringGrid. Пример окна программы во время ее работы приведен на рисунке. Если данные в ячейки не введены, выходит соответствующее сообщение.

****

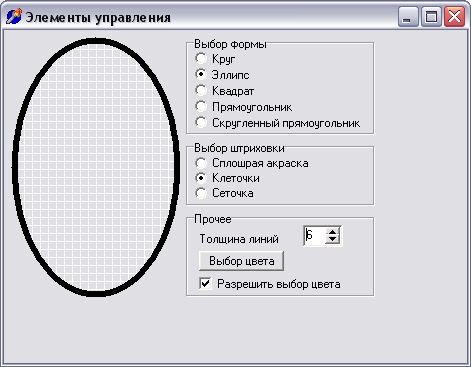
**Задача №3**

Напишите программу МРЗ Player. Программа должна обес­печивать возможность выбора каталога, в котором находятся МРЗ-файлы, а также регулировку громкости звука. Для отображения списка МРЗ-файлов используйте компонент ListBox, а для управления медиаплеером — кнопки speecffiutton. Рекомендуемый вид формы программы приведен на рис.

****

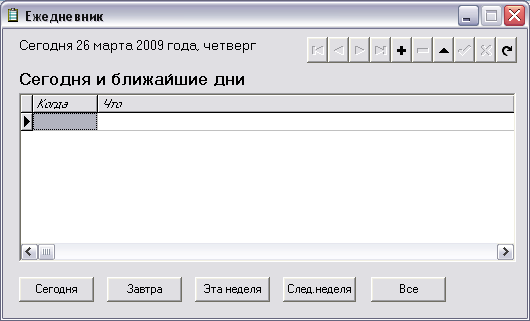
**Задача №4**

Написать программу вид которой представлен ниже.



**Задача №5**

Напишите программу – ежедневник. Рекомендуемый вид формы программы приведен на рисунке.

****

**Задача №6**

Написать программу «Текстовый редактор»

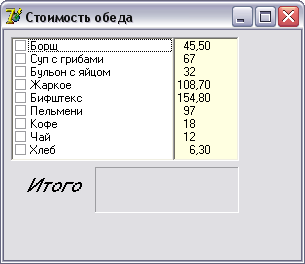
**Задача №7**

Написать программу построения графика функции вида ax2+bx+c.

Коэффициенты а, в, с вводятся пользователем.

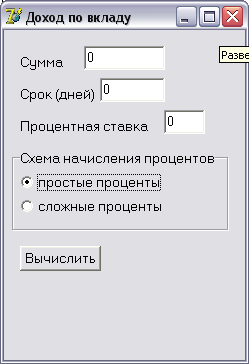
**Задача №8**

Написать программу расчета стоимости обеда. Рекомендуемый вид формы программы приве­ден на рисунке.

****

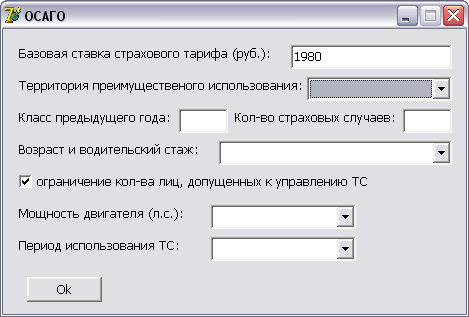
**Задача №9**

Напишите программу, которая вычисляет доход по вкладу. Программа должна обеспечивать расчет простых и сложных про­центов. Простые проценты начисляются в конце срока вклада, сложные — ежемесячно и прибавляются к текущей (накоплен­ной) сумме вклада, в следующем месяце проценты начисляются на новую сумму. Рекомендуемый вид формы программы приве­ден на рисунке

****

**Задача №10**

Напишите программу, которая позволяет рассчитать тариф ОСАГО (обязательное страхование гражданской ответственности владельца транспортного средства). Рекомендуемый вид формы приведен на рисунке.

****

**Форма промежуточного контроля**

**2 семестр - экзамен**

**ВОПРОСЫ К ЭКЗАМЕНУ**

**(2 семестр)**

1. Динамические структуры данных. Основные сведения о ссылочном типе данных (указателях).
2. Динамические структуры данных. Динамические массивы.
3. Динамические структуры данных. Линейные списки.
4. Динамические структуры данных. Стеки.
5. Динамические структуры данных. Очереди.
6. Динамические структуры данных. Деревья.
7. Объектно-ориентированное программирование (ООП): основные понятия, описание объектов, экземпляры объектов.
8. Среда Delphi. Структура проекта. Особенности реализации проектов (этапы создания проекта, интерфейс, использование шаблонов приложений).
9. Среда Delphi. Объектная модель: описание класса, наследование методов, свойства.
10. Среда Delphi. Работа с формами (характеристики формы, организация взаимодействия форм, особенности модальных форм).
11. Среда Delphi. Обзор библиотеки визуальных компонентов (VCL). Компоненты для работы с текстом.
12. Среда Delphi. Обзор библиотеки визуальных компонентов (VCL). Кнопки и переключатели.
13. Среда Delphi. Организация диалога с пользователем (процедуры и функции, реализующие диалоги, компоненты - стандартные диалоги).
14. Среда Delphi. Работа с меню (главное меню, контекстное меню, конструктор меню, динамическая настройка меню, синхронизация управляющих элементов, модификация системного меню).
15. Среда Delphi. Организация приложений (создание многодокументных приложений, особенности многодокументных приложений, создание одноэкземплярного приложения, особенности консольного приложения, запуск других приложений).

**Учебно-методическое и информационное обеспечение дисциплины** *[[2]](#footnote-2)*

**Печатные издания:**

1. Павловская, Татьяна Александровна. Паскаль. Программирование на языке высокого уровня : учебник / Павловская Татьяна Александровна. - Санкт-Петербург : Питер, 2008. - 393 с. : ил. - (Учебник для вузов).

2. Павловская, Татьяна Александровна. Паскаль. Программирование на языке высокого уровня: практикум : учеб. пособие / Павловская Татьяна Александровна. - Москва ; Санкт-Петербург, 2007. - 317 с. : ил. - (Учебное пособие).

**Издания из электронной библиотеки:**

1. Яковлева, Лидия Леонидовна. Информатика и программирование : учеб. пособие. В 2 ч. Ч. 1 / Яковлева Лидия Леонидовна.- Чита: ЗабГУ, 2014. - 213 с

2. Белов, В.В. - Программирование в Delphi: процедурное, объектно-ориентированное, визуальное [Электронный ресурс] : Учебное пособие для вузов / Белов В.В., Чистякова В.И. - 2-е изд., стереотип. - М. : Горячая линия - Телеком, 2014.

3. Трофимов, Валерий Владимирович. Алгоритмизация и программирование : Учебник / Трофимов Валерий Владимирович; Трофимов В.В. - отв. ред. - М. : Издательство Юрайт, 2017. - 137. - (Бакалавр. Академический курс. Модуль.).

4. Черпаков, Игорь Владимирович. Теоретические основы информатики : Учебник и практикум / Черпаков Игорь Владимирович; Черпаков И.В. - М. : Издательство Юрайт, 2017. - 353. - (Бакалавр. Академический курс).

***Базы данных, информационно-справочные и поисковые системы***

1. <http://ru.wikipedia.org/wiki/> - Всемирная электронная энциклопедия Википедия (Россия)
2. <http://window.edu.ru/> - электронная библиотека (единое окно доступа к образовательным ресурсам).
3. <http://pascalabc.net> - Web-среда разработки на ProgrammingABC.NET

Ведущий преподаватель доцент кафедры ПИМ Яковлева Л.Л.

Заведующий кафедрой д.э.н., профессор, Глазырина И.П.

***Приложение 1***

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

**КОНТРОЛЬНАЯ РАБОТА №1**

По дисциплине: «Информатика и программирование»

1 семестр

Вариант № \*\*\*

|  |
| --- |
| Выполнил: студент (ка)  группы \*\*\*\*  \*\*\*\*\*\*\*\*\*\*\* |
| Проверил: доцент кафедры ПИМ Яковлева Л.Л. |

Чита 2018 г

***Приложение 2***

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

**КОНТРОЛЬНАЯ РАБОТА №2**

По дисциплине: «Информатика и программирование»

2 семестр

Вариант № \*\*\*

|  |
| --- |
| Выполнил: студент (ка)  группы \*\*\*\*  \*\*\*\*\*\*\*\*\*\*\* |
| Проверил: доцент кафедры ПИМ Яковлева Л.Л. |

Чита 2018 г

1. Язык программирования высокого уровня [↑](#footnote-ref-1)
2. Внимание! Доступ к электронной библиотеке ЗабГУ: <http://mpro.zabgu.ru/MegaPro/Web>.

   Номер читательского билета можно получить в библиотеке (приходить с зачёткой). [↑](#footnote-ref-2)