

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Иркутский государственный университет путей сообщения»
Сибирский колледж транспорта и строительства

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

ОП.04 Основы алгоритмизации и программирования

по специальности

09.02.06 Сетевое и системное администрирование

базовая подготовка среднего профессионального образования

Иркутск 2022

Электронный документ выгружен из ЕИС ФГБОУ ВО ИргУПС и соответствует оригиналу

Подписант ФГБОУ ВО ИргУПС Трофимов Ю.А.

00a73c5b7b623a969ccad43a81ab346d50 с 08.12.2022 14:32 по 02.03.2024 14:32 GMT+03:00

Подпись соответствует файлу документа



Практическая работа №1

Формализация поставленной задачи. Составление блок-схем линейных и разветвляющихся алгоритмов

Цель работы: Овладеть техникой составления алгоритмов.

Задачи работы: Научиться составлять линейные и разветвляющиеся алгоритмы.

Задание: Составить алгоритмы, используя предложенные задания.

Ознакомьтесь с теоретическим материалом, необходимым для выполнения работы:

Алгоритм — совокупность последовательных шагов, схема действий, приводящих к желаемому результату.

Разработать алгоритм означает разбить задачу на определенную последовательность шагов. От разработчика алгоритма требуется знание особенностей и правил составления алгоритмов.

Основные особенности и свойства алгоритмов:

1. Наличие ввода исходных данных.
2. Наличие вывода результата выполнения алгоритма.
3. Дискретность.
4. Формальность.
5. Определенность (точность).
6. Понятность.
7. Результативность (конечность).
8. Корректность.
9. Массовость.
10. Эффективность.

Алгоритмы можно записывать по-разному. Форма записи, состав и количество операций алгоритма зависит от того, кто будет исполнителем этого алгоритма.

Способы описания алгоритма:

1. Формульная запись
2. Табличная запись
3. Развернутая словесная
4. На алгоритмическом языке
5. Графический (в виде блок-схемы)
6. На языке программирования

Основным элементарным действием в вычислительных алгоритмах является присваивание значения переменной величине. Если значение константы определено видом ее записи, то переменная величина получает конкретное значение только в результате присваивания.

Присваивание – это операция, которая значение выражения, стоящее справа от символа «:=» запоминает в переменной или элементе массива, стоящем слева. При присваивании происходит преобразование типов данных, если они не совпадают.

Линейными называются алгоритмы, в которых все действия осуществляются последовательно друг за другом, при этом каждая команда выполняется только один раз строго после той команды, которая ей предшествует. Линейный алгоритм составляется из команд присваивания, ввода, вывода и обращения к вспомогательным алгоритмам.

Разветвляющимся называется алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи, в зависимости от выполнения условий. Каждое из возможных направлений дальнейших действий называется ветвью. В блок-схемах разветвление реализуется специальным блоком «Решение». Этот блок предусматривает возможность двух выходов. В самом блоке «Решение» записывается логическое условие, от выполнения которого зависят дальнейшие действия.

Циклическим называется алгоритм, в котором некоторая часть операций (тело цикла – последовательность команд) выполняется многократно, т.е. более одного раза.

Различают:

- 1.циклы с известным числом повторений (или со счетчиком);
- 2.циклы с неизвестным числом повторений (циклы с предусловием и циклы с постусловием).

В любом цикле должна быть переменная, которая управляет выходом из цикла, т.е. определяет число повторений цикла.

В циклах со счетчиком известно число повторений цикла, т.е. оно является фиксированным числом. В этом случае переменная, которая считает количество повторений (шагов) цикла, называется счетчиком цикла (или параметром цикла, или управляющей переменной цикла).

Циклы с предусловием – это такие циклы, в которых до начала выполнения тела цикла проверяется условие выполнения следующего шага цикла. Если значение этого условия истинно (т.е. условие выполняется), то выполняется тело цикла.

Цикл с постусловием также используется при неизвестном заранее количестве повторений цикла, но в отличие от цикла с предусловием здесь условие на выход из цикла проверяется после того, как выполнились операторы тела цикла.

Задание 1: Составить линейные алгоритмы, используя предложенные задания.

Рассмотрите примерные упражнения

1. Составление алгоритма в табличной форме для вычисления выражения :

$$y = \frac{8x - 1}{4x + 2}$$

Исходные данные: x.

Результат: y.

Последовательность шагов и описание действий:

1. $a := 8x$
2. $b := a - 1$
3. $c := 4x$
4. $d := c + 2$
5. $y := b/d$

2. Составление алгоритма вычисления площади поверхности цилиндра с диаметром D и высотой H в табличной форме.

Исходные данные: D, H.

Результат: S.

Последовательность шагов и описание действий:

1. $a := \pi D^2$
2. $b := a / 4$
3. $c := \pi D H$
4. $S := b + c$

3. Вычислить факториал числа n ($c=n!$), который вычисляется по формуле $c=1*2*3*4*\dots*n$. Алгоритм представить в словесной форме.

Исходные данные: n.

Результат: c.

Алгоритм:

1. Полагаем $c=1$ и переходим к следующему пункту.
2. Полагаем $i=1$ и переходим к следующему пункту.
3. Полагаем $c=i*c$ и переходим к следующему пункту.
4. Проверяем, равно ли i числу n. Если $i=n$, то вычисления прекращаем. Если i больше n, то увеличиваем i на 1 и переходим к пункту 3.

Выполнить самостоятельно

Составьте алгоритмы решения следующих задач:

1. По заданным формулам составить вычислительные алгоритмы в виде таблиц:

$$S = \frac{\pi D^2}{4} + \frac{\pi D L}{2}$$

(S – площадь боковой поверхности конуса, D – диаметр основания, L – образующая);

$$f = \frac{1}{2\pi\sqrt{LC}}$$

(f – частота собственных колебаний в контуре, L – индуктивность катушки, C – емкость конденсатора).

2. Дано значение x . Получить значения $-2x+3x^2-4x^3$ и $1+2x+3x^2-4x^3$. Позаботиться об экономии операций.

3. Дано значение a . Не используя никаких функций и никаких операций, кроме умножения, получить значение a^8 за три операции и a^{10} за четыре операции.

4. Составить алгоритм для вычисления пути, пройденного лодкой, если ее скорость в стоячей воде v км/ч, скорость течения реки v_1 км/ч, время движения по озеру t_1 ч, а против течения реки – t_2 ч. Использовать словесный способ записи алгоритма.

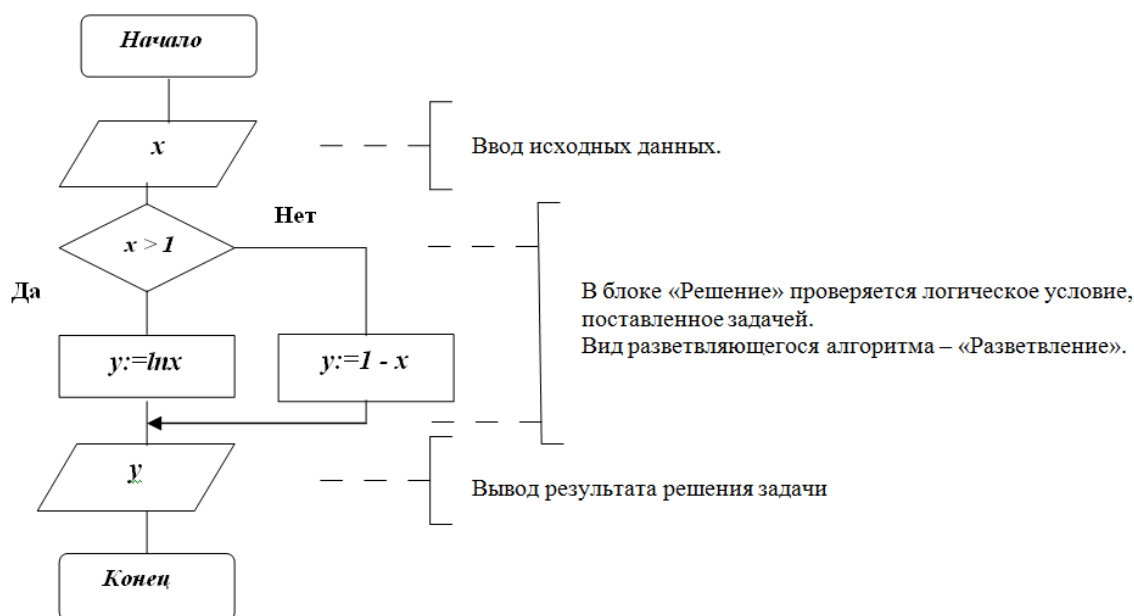
Задание 2: Составить разветвляющиеся алгоритмы, используя предложенные задания.

Рассмотрите примерные упражнения:

1. Составьте блок-схему вычисления значения функций:

$$y = \begin{cases} \ln x, & x > 1, \\ 1 - x, & x \leq 1 \end{cases}$$

Исходные данные: x . Результат: y .



Выполнить самостоятельно

Составьте алгоритмы решения следующих задач:

1. Изобразите блок-схему простого диалогового алгоритма, который обращается к пользователю с просьбой ввести сначала строку имя, а затем строку настроение. В результате диалога может появиться следующий совместный текст:

Программа> Здравствуйте! Как Ваше имя?

Пользователь> Гаврик

Программа> Доброе утро, Гаврик! Как настроение?

Пользователь> так себе

Программа> У меня тоже так себе, Гаврик!

2. Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень – отрицательные.

3. Составьте блок-схему вычисления значения функций:

$$y = \begin{cases} \frac{5x^2 + 2}{x + 4}, & \text{если } x > -4 \\ 3x^2 + 7, & \text{если } x \leq -4 \end{cases}$$

4. Составьте программу, заменяющую меньшее из двух данных чисел суммой, а большее – произведением этих чисел.

5. Составить блок-схему алгоритма вычисления площади треугольника 3-мя различными способами на выбор пользователя.

Задание 3: Составить циклические алгоритмы, используя предложенные задания.

Рассмотрите следующие упражнения

1. Составьте алгоритм вычисления суммы первых 20 членов последовательности с общим членом $a = S + (k + 1) / (2k + 1)^2$, используя циклы с предусловием и постусловием. Для циклического накапливания сумм при составлении соответствующих алгоритмов используется предписание стандартного вида: $\text{сумма} := \text{сумма} + \text{слагаемое}$. Если повторять такое предписание требуемое количество раз, изменяя соответствующим образом слагаемое, то и будет получена искомая сумма. Понятно, что сумма перед началом работы цикла должна иметь нулевое значение. В схеме, изображенной ниже, роль суммы выполняет переменная S, а роль слагаемого – формула общего члена последовательности $(k + 1)(2k + 1)$. Изменение слагаемого достигается увеличением в каждом обороте цикла номера члена k на единицу. Словесная запись этого алгоритма:

Цикл с постусловием:

1. начало
2. $k := 1; S := 0$
3. $S := S + (k + 1) / (2k + 1)^2$
4. $k := k + 1$
5. если $k \leq 20$ идти к п.3

6. вывод S

7. Конец

Цикл с предусловием:

1. начало

2. $k := 1; S := 0$

3. если $k > 20$ идти к п.6

4. $S := S + (k + 1) / (2k + 1)^2$

5. $k := k + 1$; идти к п.3

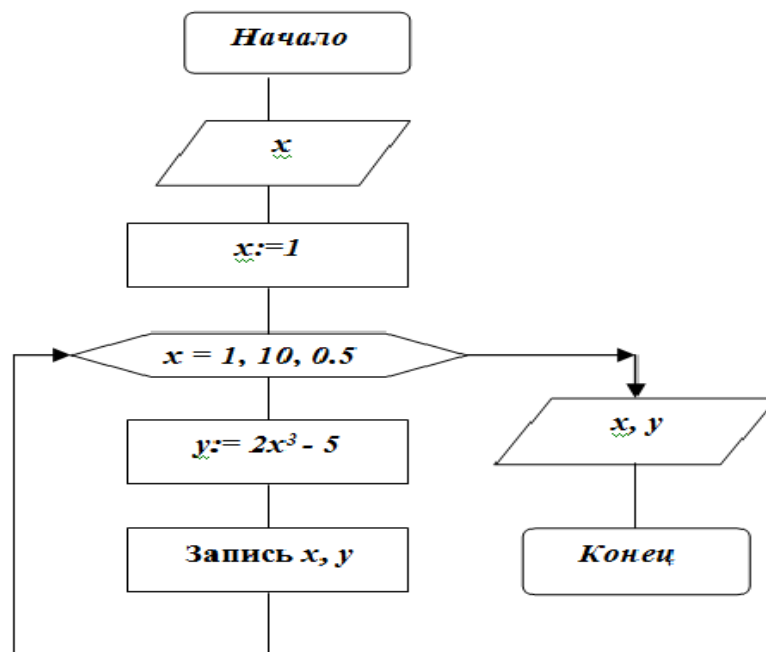
6. вывод S

7. Конец

2. Составьте блок-схему алгоритма для вычисления значений функции для значений x , начиная с $x=1$ до $x=10$ с шагом $h=0,5$.

Исходные данные: x .

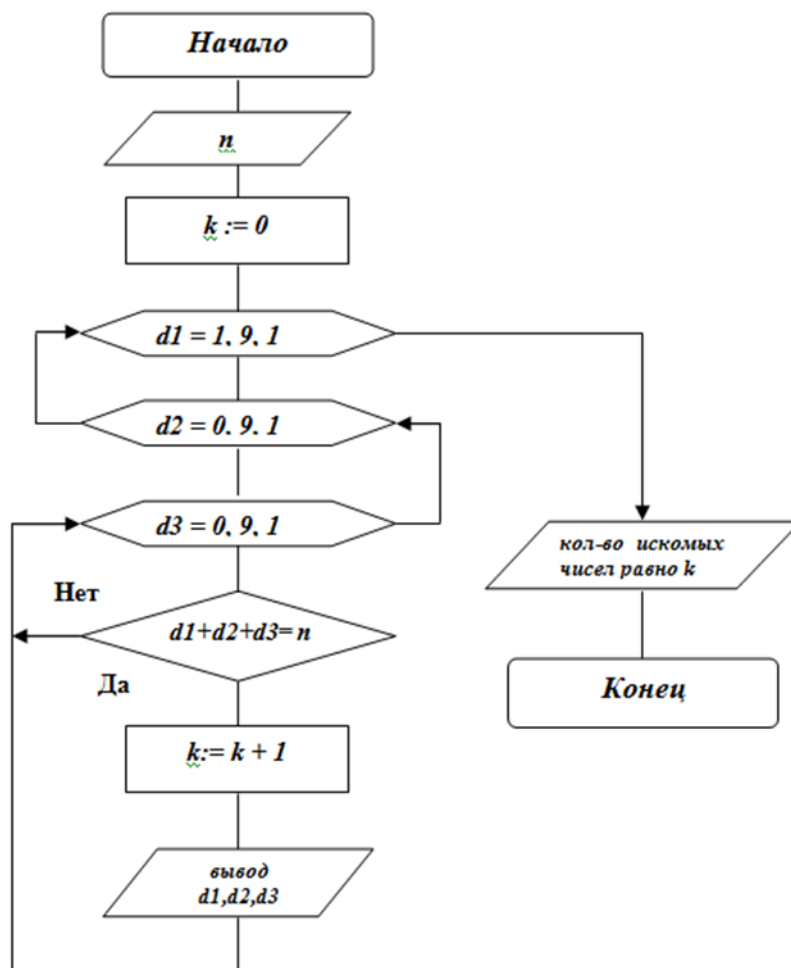
Результат: x, y .



3. Составьте алгоритм для определения k – количества трехзначных натуральных чисел, сумма цифр которых равна n ($1 < n < 27$). Операции деления не использовать. В алгоритме приняты обозначения: $d1$ - левая, $d2$ - средняя, $d3$ – правая цифры числа.

Исходные данные: n .

Результат: $k, d1, d2, d3$.



Выполнить самостоятельно

Составьте алгоритмы решения следующих задач:

1. Составьте блок-схемы алгоритмов для задачи из Упражнения 1.
2. Составьте блок-схему алгоритма для решения следующей задачи: начав тренировки, спортсмен пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?
3. Изобразите блок-схему алгоритма определения максимального числа в последовательности из n произвольных чисел.
4. Составьте блок-схему алгоритма вычисления суммы квадратов первых n чисел натурального ряда.
5. Найти сумму первых n членов ряда

$$y = 1 + \frac{x}{2} + \frac{2 * x}{3} + \frac{3 * x}{4} + \dots \text{при } |x| < 1$$

6. Найти все делители целого положительного числа (кроме 1 и самого числа).

Инсталляция и конфигурирование компилятора Pascal. Навыки работы с HELP. Работа с готовыми примерами

Цель работы:

Научиться инсталляции и конфигурированию компилятора Pascal. Закрепить на практике теоретические знания по работе с HELP в среде Turbo Pascal

Инсталляция и конфигурирование компилятора Pascal.

Для инсталляции и конфигурирования компилятора Pascal скачать архив. Он содержит все необходимое для запуска Borland Pascal под ОС Windows (32- и 64-разрядными версиями), в том числе русификатор (переход на русский язык и обратно с помощью клавиши Ctrl). Настоятельно не рекомендуется брать Borland или Turbo Pascal из других источников, так как по умолчанию Borland и Turbo Pascal игнорирует многие ошибки в программах, а скомпилированные программы могут не работать на достаточно быстрых компьютерах.

- Если при раскрытии на полный экран (Alt+Enter) искажаются цвета и окно BP из синего становится красным, то следует в конец файла dos\dosbox.conf добавить строки:
[sdl]
output = opengl
- Если наблюдаются проблемы с работой арифметики с плавающей запятой, то проверьте, что в настройках Options -> Compiler снята галочка с пункта 8087/80287, в разделе Numeric Processing.

Исходные данные (задание):

Задание1 Взять из HELP пример рисования окружности, перенести его в среду Turbo Pascal и запустить на выполнение.

Задание2 Взять из HELP пример рисования линии, перенести его в среду Turbo Pascal и запустить на выполнение.

Задание3 Взять из HELP пример рисования прямоугольника, перенести его в среду Turbo Pascal и запустить на выполнение.

Порядок выполнения

- 1.Загрузить Turbo Pascal
- 2.Найти HELP
- 3.Найти пример
- 4.Выделить его
- 5 Загрузить в буфер
- 6.Поместить в Turbo Pascal
- 7.Запустить на выполнение

8. На экране должна появиться нужная фигура (окружность, линия и т.д.)

Указания по работе с HELP

Все действия лучше выполнять с помощью горячих клавиш.

CTRL/F1 - Найти HELP

CTRL/INS- Загрузить в буфер

SHIFT/INS- Поместить в Turbo Pascal

CTRL/F9- Запустить на выполнение

Практическая работа №4

Программирование линейных алгоритмов. Использование различных типов исходных и выходных данных

Цель работы:

- Закрепить на практике теоретические знания по созданию программ с линейными алгоритмами, закрепить на практике навыки работы в среде Turbo Pascal
- научиться записывать на языке Pascal математические выражения;
- научиться использовать стандартные функции языка Паскаль
- научиться работать с целыми типами данных;
- научиться работать с вещественными типами данных

Исходные данные (задание):

Задание Написать в среде Turbo Pascal программы, реализующие следующие линейные алгоритмы, запустить их на выполнение.

$b = \left(1 + \operatorname{tg}^2 \frac{z}{2} \right)$	$b = 1 + \frac{z^2}{3 + z^2 / 5}$
$a = \frac{1 + \sin^2(x + y)}{2 + \left x - 2x / (1 + x^2 y^2) \right } + x$	$a = \ln \left \left(y - \sqrt{ x } \right) \right \left(x - \frac{y}{z + x^2 / 4} \right)$
$b = \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right)$	$f = 6,673 \cdot 10^{-8} \cdot \frac{m_1 \cdot m_2}{r^2}$
$t = \frac{xyz - 3,3 \left x + \sqrt[4]{y} \right }{10^7 + \sqrt{\lg 4}}$	$b = e^{ x-y } \ln(1+e) \log_2 \operatorname{tg} 2$
$y = \frac{\beta + \sin^2 \pi^4}{\cos 2 + \operatorname{ctg} \gamma }$	$y = \sqrt[8]{x^8} + 8^x$

Задачи: 1 Написать в среде Turbo Pascal программы, реализующие следующие линейные алгоритмы, запустить их на выполнение

1. Написать программу, которая для заданного целого числа a печатает следующую таблицу:

$$\begin{array}{c} a \\ a^3 \quad a^6 \\ a^6 \quad a^3 \quad a \end{array}$$

2. Вычислить длину окружности, площади круга и объема шара одного и того же заданного радиуса.

3. Написать программу вычисления периметра и площади прямоугольного треугольника по длинам двух катетов.

4. Написать программу, которая по координатам трех вершин некоторого треугольника вычисляет его площадь и периметр.

5. Найти площадь кольца, внутренний радиус которого равен 20, а внешний – заданному числу r ($r > 20$).

Порядок выполнения

1. Составить в тетради алгоритм
2. Написать программу
3. Отладить программу
4. Протестировать программу
5. Записать в тетради текст программы
6. Записать в тетради входные и выходные данные

Программирование задач с операторами условия и выбора

Цель работы:

1. научиться работать с оператором условия IF THEN ELSE;
2. научиться работать с оператором выбора CASE

Задание: Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение.

1. Написать программу вычисления Y:

$$y = \begin{cases} x & , x \geq 1 \text{ или } x < -2 \\ 2x + 16 - x & , \text{остальных случаях} \end{cases}$$

2. Написать программу вычисления X:

$$x = \begin{cases} 14 & m = d, \quad l < k \\ 15 & m = d, \quad l \geq k \\ 16 & m \neq d \end{cases}$$

3. Составить программу для подсчета числа букв А, Б, В - в предложении. Предложение вводится отдельными символами до тех пор пока не введен символ (.).

4. В строке введенных символов подсчитать количество символов С, D, Р, считая концом ввода символ (*).

5. По введенному номеру месяца напечатать наименование времени года и наименование месяца.

6. Улица Байкальская разбита на участки:

с 1 по 100 дом - 1 участок
с 101 по 203 дом - 2 участок
с 204 по 308 дом - 3 участок
остальные дома - 4 участок.

По введенному номеру дома выдать № участка.

7. Лежит ли введенное значение X на отрезке в интервале [4,5.4].

8. В пятиэтажном доме на каждом этаже расположено по четыре квартиры. Составить программу, которая по номеру этажа, печатает номера квартир на этом этаже.

9. Написать программу нахождения наибольшего числа из введенных чисел, считая концом ввода 0.

Практическая работа №6

Программирование задач с оператором цикла FOR

Цель работы: Научиться работать с оператором цикла FOR. Научиться работать с оператором цикла WHILE (цикл с предусловием). Научиться работать с оператором цикла Repeat Until(цикл с постусловием).

Задание 1: Программирование задач с оператором цикла FOR. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение.

1. Составить программу определения разрядности введенного целого числа.
2. Подсчитать сумму отрицательных чисел последовательности.
3. Составить программу подсчитывающую сумму цифр вводимого натурального числа.
4. Имеется одномерный массив из 15 чисел (a1, a2, a3 ... a15). Составить программу их ввода. Упорядочить массив по убыванию.
5. Ввести 5-тизначное число. Вывести последовательность цифр начиная с конца.

Пример: ввод - 42891
вывод - 1 9 8 2 4

Практическая работа №7

Программирование задач с оператором цикла WHILE

Цель работы: Научиться работать с оператором цикла WHILE (цикл с предусловием).

Задание 2: Программирование задач с оператором цикла WHILE. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение.

1. Подсчитать ax^2+bx+c , при $a=18.5$, $b=-0.5$, $c=134$, x изменяется от -1 до 0.6 с шагом 0.2. Значения выдавать на каждом шаге итерации.

2. Подсчитать сумму положительных чисел последовательности.

3. Подсчитать $w=(a+bx)\sqrt{x+1}$, при $a=2.8$, $b=-0.3$, x изменяется от 1 до 3 с шагом 0.5. Выдавать значения на каждом шаге итерации

4. Имеется одномерный массив из 20 чисел ($B_1, B_2, B_3 \dots B_{20}$). Составить программу их ввода. Упорядочить массив по возрастанию.

5. Подсчитать

$$zz = \begin{cases} fv/(f-6)x & x \geq 13.6 \\ (f+v)xx/v+3 & x < 4.8 \\ x/vf & \text{в остальных случаях} \end{cases}$$

Практическая работа №8

Программирование задач с оператором цикла FOR

Цель работы: Цель работы: Научиться работать с оператором цикла Repeat Until(цикл с постусловием).

Задание.3 Программирование задач с оператором цикла RepeatНаписать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

Перевернуть число 12345 54321

```
var
c,b,x : integer;
begin
writeln('Введите число');
readln(x);
if x < 9 then
writeln('Число должно быть больше 9')
else
repeat
c:=x div 10;
b:=x mod 10;
x:=c;
write(b);
until c < 9;
writeln(c);
readln;
end.
```

1.Подсчитать ax^2+bx+c , при $a=18.5$, $b=-0.5$, $c=134$, x изменяется от -1 до 0.6 с шагом 0.2. Значения выдавать на каждом шаге итерации.

2. Подсчитать сумму положительных чисел последовательности.

3. Подсчитать $w=(a+bx)\sqrt{x+1}$, при $a=2.8$, $b=-0.3$, x изменяется от 1 до 3 с шагом 0.5. Выдавать значения на каждом шаге итерации

4. Имеется одномерный массив из 20 чисел ($B_1, B_2, B_3 \dots B_{20}$). Составить программу их ввода. Упорядочить массив по возрастанию.

5. Подсчитать

$$zz = \begin{cases} fv / (f - 6)x & x \geq 13.6 \\ (f + v)xx / v + 3 & x < 4.8 \\ x / vf & \text{в остальных случаях} \end{cases}$$

Практическая работа №9

Обработка одномерных массивов

Цель работы: Научиться писать программы с использованием одномерных и двумерных массивов.

- научиться описывать одномерные массивы;
- научиться вводить одномерные массивы;
- научиться генерировать одномерные массивы;
- научиться выводить одномерные массивы;
- осуществлять поиск минимального и максимального элементов массива;
- упорядочивать массивы по возрастанию и убыванию.

Задание Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Вычисляя значения переменной $x=8d+f$ при всех значениях $d=1,2,3$ и $f=-3, 3, -6$. Создать одномерный массив. Вывести значения элементов этого массива и значения d, f .

2. Создать одномерный массив из случайно сгенерированных чисел (от 1 до 36). Подсчитать сумму нечетных чисел элементов массива. Выдать на экран массив и найденную сумму.

3. Из случайно сгенерированных чисел (от 1 до 25) создать одномерный массив. Выдать массив на экран в виде:

a1			
	a2		
		a3	
			a4

4. Вычисляя значения переменной $Z=3A/(9+X)$ при всех значениях $A=0,12,20$ и $X=-2,-3,-8$. Создать одномерный массив. Вывести значения элементов этого массива на экран:

$z(a1,x1)$	$z(a1,x2)$	$z(a1,x3)$
$z(a2,x1)$	$z(a2,x2)$	$z(a2,x3)$
$z(a3,x1)$	$z(a3,x2)$	$z(a3,x3)$

5. Ввести данные для формирования одномерного массива $X[10]$. Поменять местами элементы массива, введя их номера с клавиатуры.

Практическая работа №10

Обработка двумерных массивов

Цель работы: Научиться писать программы с использованием одномерных и двумерных массивов.

- научиться описывать двухмерные массивы;
- научиться вводить двухмерные массивы;
- научиться генерировать двухмерные массивы;
- научиться выводить двухмерные массивы;
- осуществлять поиск минимального и максимального элементов массива;
- упорядочивать массивы по возрастанию и убыванию.

Задание Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Транспонировать матрицу и подсчитать количество равных элементов

```
program pr1;
const N1=10;N2=12;
var A,A1: array[1..N1,1..N1] of integer;
    B,B1: array[1..N2,1..N2] of integer;
    i,j,k,k1: integer;

begin

randomize;

for I:=1 TO N1 Do begin
for J:=1 TO N1 Do begin
  A[I,J]:=random(10);
  A1[J,i]:=A[i,j];
  end;
end;
for I:=1 TO N2 Do begin
for J:=1 TO N2 Do begin
  B[I,J]:=random(10);
  B1[j,i]:=B[i,j];
  end;
end;
k:=0;
for I:=1 to N2 do begin
for J:=1 to N2 do begin
if B1[i,j]=B[i,j] then inc(k);
end; end;
k1:=0;
for I:=1 to N1 do begin
for J:=1 to N1 do begin
if A1[i,j]=A[i,j] then inc(k1);
end; end;write (k:4,k1:4);
readln;
```

2. Создать трехмерный массив из случайно сгенерированных вещественных чисел, заменяя те элементы массива на 0, которые > 55.0 .
3. Ввести вещественные числа для формирования массива $A[3,3]$, подсчитать сумму диагональных элементов этого массива.
4. Ввести вещественные числа для формирования массива $A[4,4]$. Переставить колонки этого двумерного массива: 1 на место 2, 2 на место 3, 3 на место 4, 4 на место 1.
5. Ввести данные для формирования двумерного массива $L[3,3]$, формируя вначале строки. Поменять местами строки и колонки: 1 строку перенести в 1 колонку, 2 строку перенести во 2 колонку, а 3 строку поместить в 3 колонку.
6. Сгенерировать двумерный массив. Вычислить сумму элементов обратной диагонали.

Практическая работа №11

Программирование задач сортировки массивов

Цель работы: Научиться писать программы по упорядочиванию массивов по возрастанию и убыванию

Исходные данные (задание):

Задание Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

```
var i,j,n,buf : integer;
a : array [1..15] of integer;
begin
n:=15;
for i:=1 to n do
readln(a[i]);
for i:=1 to n do
for j:=1 to n-1 do
if a[i]>a[j+1]then
begin
buf := a[i];
a[i] := a[j+1];
a[j+1] := buf;
end;
for i:=1 to n do
write(a[i]:4);
writeln;
readln;
end.
```

- 1 . Разобраться с алгоритмом сортировки методом вставки.
- 2 . Разобраться с алгоритмом сортировки методом пузырька
- 3 . Разобраться с алгоритмом сортировки методом Шелла

Практическая работа №12

Использование стандартных функций и процедур для работы со строками

Цель работы:

Научиться писать программы по программированию строковых массивов

Задание Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Заменить в предложении одно слово на другое

```
program slovo;  
var t,a,b,k,s,c:string;  
i,j:integer;  
begin  
  write('Введите исходный текст ');  
  readln(t);  
  write('Введите слово которое нужно заменить ');  
  readln(a);  
  write('Введите слово для замены ');  
  readln(b);  
  t:=t+' '  
  i:=1;  
  s:=' '  
  while i<>0 do  
  begin  
    i:=pos(' ',t);  
    c:=copy(t,1,i-1);  
    if c=a then s:=s+b+' ' else s:=s+c+' '  
    t:=copy(t,i+1,length(t)-i);  
  end;  
  writeln('Новый текст',s:10);  
  readln;  
end.
```

2. Определить, сколько раз в тексте встречается заданная буква.

3. Подсчитать, сколько слов в тексте начинается на заданную букву

Программирование стандартных процедур и функций

Цель работы: Научиться работать с использованием стандартных процедур.
Задание1. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ y }}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$	$z = \frac{ x - y }{1 + xy }$
$a = \frac{3 + e^{y-1}}{1 + x^2 y - \operatorname{tg} z }$	$a = (1 + y) \frac{x + y / (x^2 + 4)}{e^{-x-2} + 1 / (x^2 + 4)}$
$b = 1 + y - x + \frac{(y - x)^2}{2} + \frac{ y - x ^3}{3}$	$b = \frac{1 + \cos(y - 2)}{x^4 / 2 + \sin^2 z}$
$a = y + \frac{x}{y^2 + \left \frac{x^2}{y + x^3 / 3} \right }$	$a = \frac{2 \cos(x - \pi / 6)}{1 / 2 + \sin^2 y}$
$b = \left(1 + \operatorname{tg}^2 \frac{z}{2} \right)$	$b = 1 + \frac{z^2}{3 + z^2 / 5}$
$a = \frac{1 + \sin^2(x + y)}{2 + x - 2x / (1 + x^2 y^2) } + x$	$a = \ln \left \left(y - \sqrt{ x } \right) \right \left(x - \frac{y}{z + x^2 / 4} \right)$

Задание 2 Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Ввести x и вычислить значения всех известных процедур и функций.
2. Нарисовать их графики по заданному примеру(график SIN(x))

ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИЯ SIN(x)

USES graph;

var DRIver, mode: integer;

```

x:real;yg,yg,i:integer;
BEGIN
DRIVER:=DETECT;
INITgraph(driver,mode,'c:\tp\bgi');
setcolor(2);
line(10,90,200,90);
line(10,20,10,160 );
x:=0;
for i:=0 to 190 do
begin xg:=10+round(95/pi*x);
yg:=90-round(50*sin(x));
putpixel(xg,Yg,yellow);
x:=x+pi/95
end;
outtextxy(15,30,'y');
outtextxy(205,90,'x');
outtextxy(130,40,'y=sin(x)');
readln;
closeGraph;
end.

```

3. Определить чётность X через ODD и mod
4. Найти код ASCII заданного символа
5. Найти символ по заданному коду ASCII

Программирование процедур и функций пользователя

Цель работы: Научиться работать с использованием процедур и функций пользователя

Задание 2. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение:

1. Паскаль сам по себе не предоставляет функции, вычисляющей площадь круга по фактическому параметру – диаметру круга. Однако такую функцию легко определить самостоятельно. После этого функцию, например, `circle` можно использовать в программе точно так же, как ранее использовались функции `sqr` или `trunc`.

В первой строке определения функции параметр `d` как бы говорит «Делай то, что делается со мной, но используй значение, которое будет на моем месте». При вызове функции на место `d` ставится число, которое таким образом возводится в квадрат, затем результат умножается на 3.14159 и делится на 4.0. Параметр `d` – это формальный параметр, тогда как конкретное число – фактический параметр. В основной программе, обращающейся к функции `circle`, можно использовать имя `d` как имя переменной (или любого другого объекта), не опасаясь помех со стороны функции.

Также приведены функции, вычисляющие площади прямоугольника и треугольника; их параметрами являются длины сторон.

Все эти функции могут быть вызваны из программы в любой последовательности и любое количество раз.

Обратите внимание на то, что функции вызываются с фактическими параметрами `x`, `y`, `z`, тогда как формальные параметры в их определениях – `a`, `b`, `c`, `d`. Переменная `a` в основной программе никак не связана с формальным параметром `a` в функциях. Точно так же отсутствует связь переменной `x` в основной программе и локальной переменной `x` в функции `triangle`.

Определенные здесь функции различаются по числу параметров. В функциях может быть любое фиксированное число параметров; определить функцию с переменным числом параметров (например, как в случае `read(a)`, `read(a, b)`, `read(a, b, c)`) нельзя – такой возможностью пользуется только сам Pascal.

Программа на языке Паскаль:

```
var
  choice: char;
  a, x, y, z: real;

function circle(d: real): real;
  const pi = 3.14159;
  begin
    circle := pi * sqr(d) / 4.0;
  end;

function rectangle(a, b: real): real;
```

```

begin
    rectangle := a * b
end;

function triangle(a, b, c: real): real;
var x: real;
begin
    x := (a + b + c) / 2;
    triangle := sqrt(x * (x - a) * (x - b) * (x - c))
end;

begin
    repeat
        read(choice);
        case choice of
            'c': begin
                readln(x);
                a := circle(x)
            end;
            'r': begin
                readln(x, y);
                a := rectangle(x, y)
            end;
            't': begin
                readln(x, y, z);
                a := triangle(x, y, z)
            end
        end;
        writeln('Square: ', a:8:2);
    until choice = 'q';

    readln
end.

```

2. Написать функцию, которая вычисляет объём цилиндра. Параметрами функции должны быть радиус и высота цилиндра.
3. Написать функцию, которая сравнивает два целых числа и возвращает результат сравнения в виде одного из знаков $>$, $<$ или $=$.
4. Написать функцию, которая вычисляет ab . Числа a и b могут быть дробными положительными числами.
5. Написать функцию *Dohod*, которая вычисляет доход по вкладу. Исходными данными для функции являются: величина вклада, процентная ставка (годовых) и срок вклада (количество дней)

Программирование задач с рекурсией

Цель работы: Научиться работать с использованием рекурсивных алгоритмов

Исходные данные (задание):

Задание 1 Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

```
program faktorial;
uses crt;
var n,i,fak: integer;
begin
  clrscr;
  write ('введите число=');
  readln (n);
  fak:=1;
  for i:=2 to n do begin
    fak:=fak*i;
  end;
  write ('факториал числа=',fak);
  readln (fak);
end.
```

Задание 2 Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Написать рекурсивную программу вычисления максимального числа Фибоначи, ближайшего к заданному n по недостатку.
2. Написать рекурсивную программу поиска индекса минимального элемента массива
3. Написать рекурсивную программу поиска минимального элемента массива

Программирование задач с использованием данных типа записи

Цель работы: Научиться программировать записи.

- научиться описывать структуры типа записи;
- научиться вводить структуры типа записи;
- научиться выводить структуры типа записи;

Исходные данные (задание):

Задание Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Создать массив из 4 записей. Структура записи:

Ф.И.О. студента

возраст студента

характеристика (был ли студент в армии)

Выбрать в этом массиве записи и выдать на экран Ф.И.О. студентов, которые были в армии

2. Создать массив из 3 записей. Структура записи:

наименование отделения

наименование группы

количество неуспевающих

количество пропусков уроков

Выбрать в этом массиве записи и выдать на экран наименование групп, в которых больше всего неуспевающих и пропусков.

3. На конкурсе красоты регистрируются участники. Создать массив из 5 записей. Структура записи:

Ф.И.О. участника конкурса

рост

вес

объем талии

Выбрать в этом массиве записи и выдать на экран Ф.И.О. тех участников, параметры которых соответствуют таким эталонам:

рост 165-175

вес 56-59

объем талии 60-64.

4. Создать структуру записи:

Ф.И.О. автора

название книги

название издательства

Выбрать книги введенного с клавиатуры определенного издательства.

5. Создать структуру записи:

страна

столица

численность населения столицы

Выбрать столицы с населением более 8 миллионов.

Программирование задач работы с текстовыми файлами

Цель работы:

- научиться описывать файлы;
- научиться записывать информацию в файлы;
- научиться читать информацию из файлов.

Исходные данные (задание):

Задание Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Программу формирующую записи изменить таким образом, чтобы все сформированные записи хранились во внешнем файле, для этого создать программу со следующими режимами в виде процедур:

- первоначальное создание файла на диске с двумя или тремя записями;
- дозапись созданного файла произвольным числом записей;
- просмотр созданного файла;
- выбор необходимых записей по условию;
- выход в среду Паскаля.

Просмотр созданного файла сделать в виде таблицы с выводом шапки для каждого поля файла.

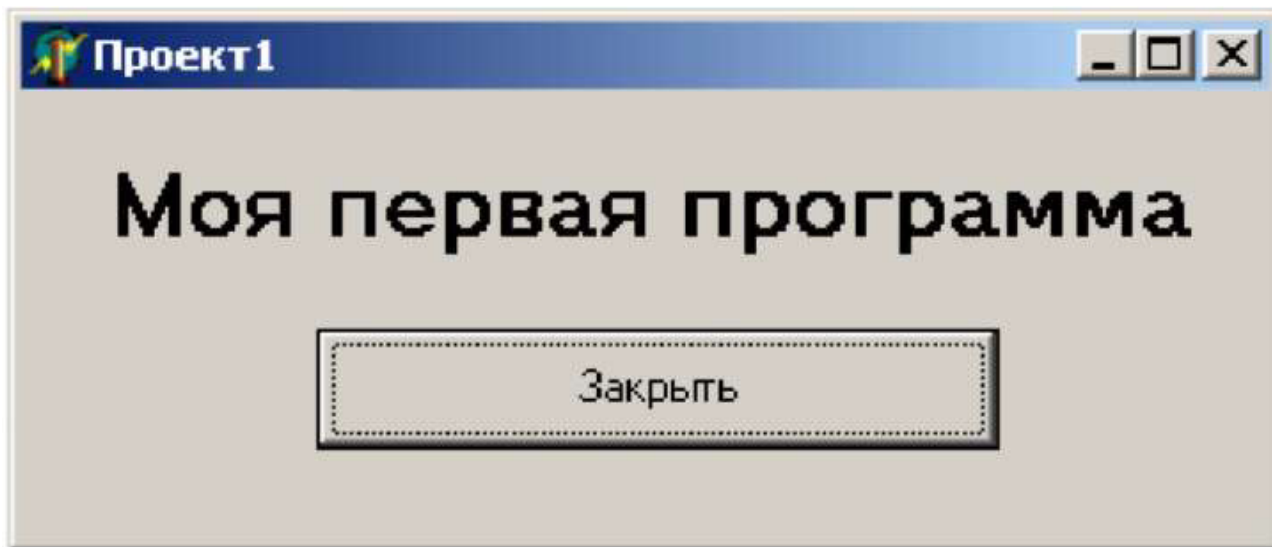
2. Создать для предыдущей программы режим работы в меню с помощью окон, закрашенных в разный цвет, предусмотреть режим перемещения курсора и выбор пунктов меню.

Практическая работа №19

Создание консольного приложения

Задание 1 Цель работы - создать программу, выполняющую следующие действия:

1. После запуска программы появляется окно.



2. Для выхода из программы необходимо щелкнуть мышью на кнопке "Закреть". Описание плана разработки программы

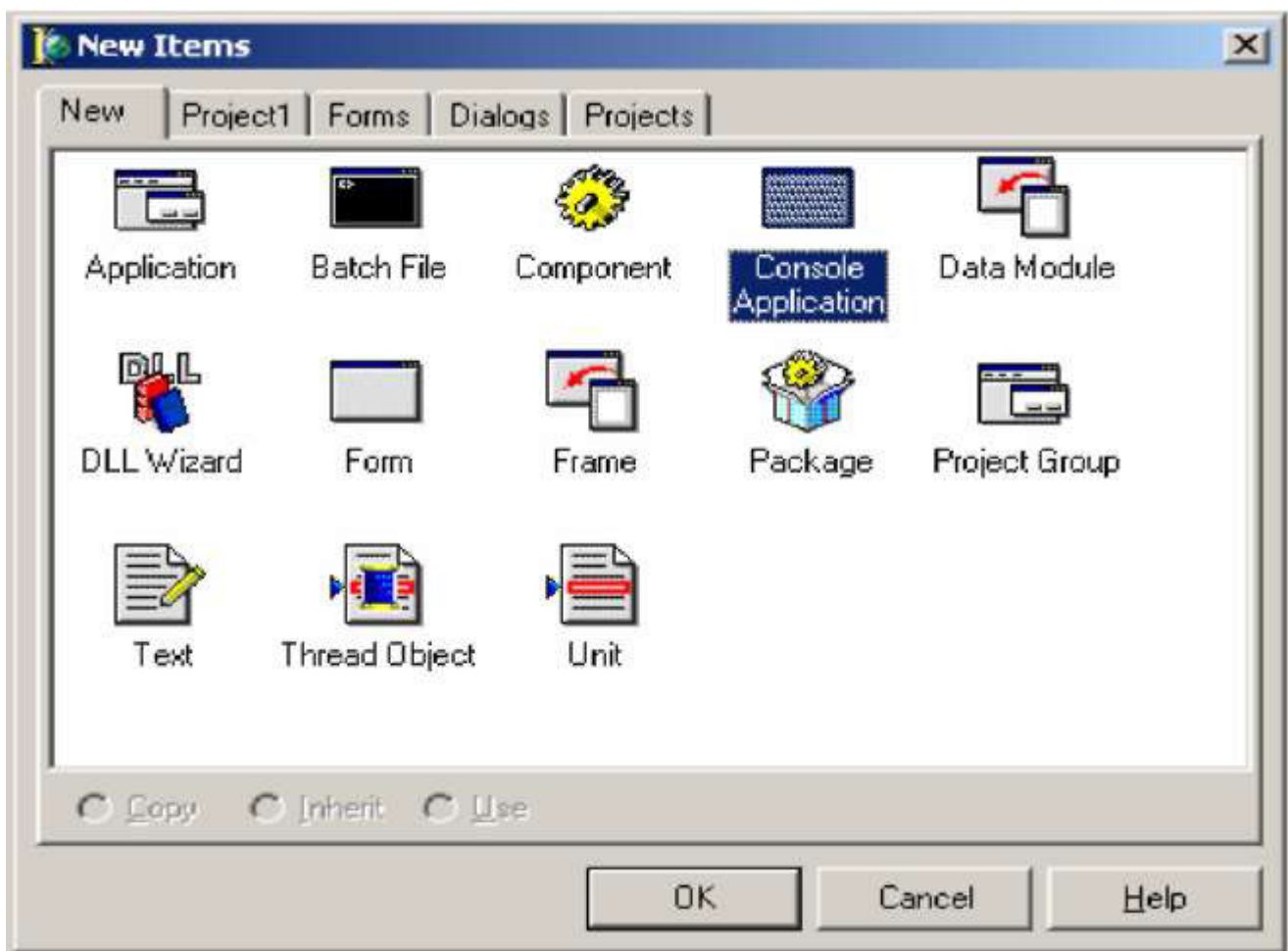
1. Открыть новый проект.
2. Разместить на форме экземпляры компонентов: метку Label и кнопку Button.
3. Выполнить следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ имя события	Действие
Form1	Properties	Caption (надпись)	Установка имени формы "Проект1"
Label1 (Вкладка Standard)	Properties	Caption	Ввод текста надписи "Моя первая программа"
		AutoSize (Автоподбор)	Ввод значения свойства: True
		Font - Color	Выбрать цвет: clPurple
Button1 (Вкладка Standard)	Properties	Caption	Установка имени кнопки "Закреть"
	Events	OnClick	Close;

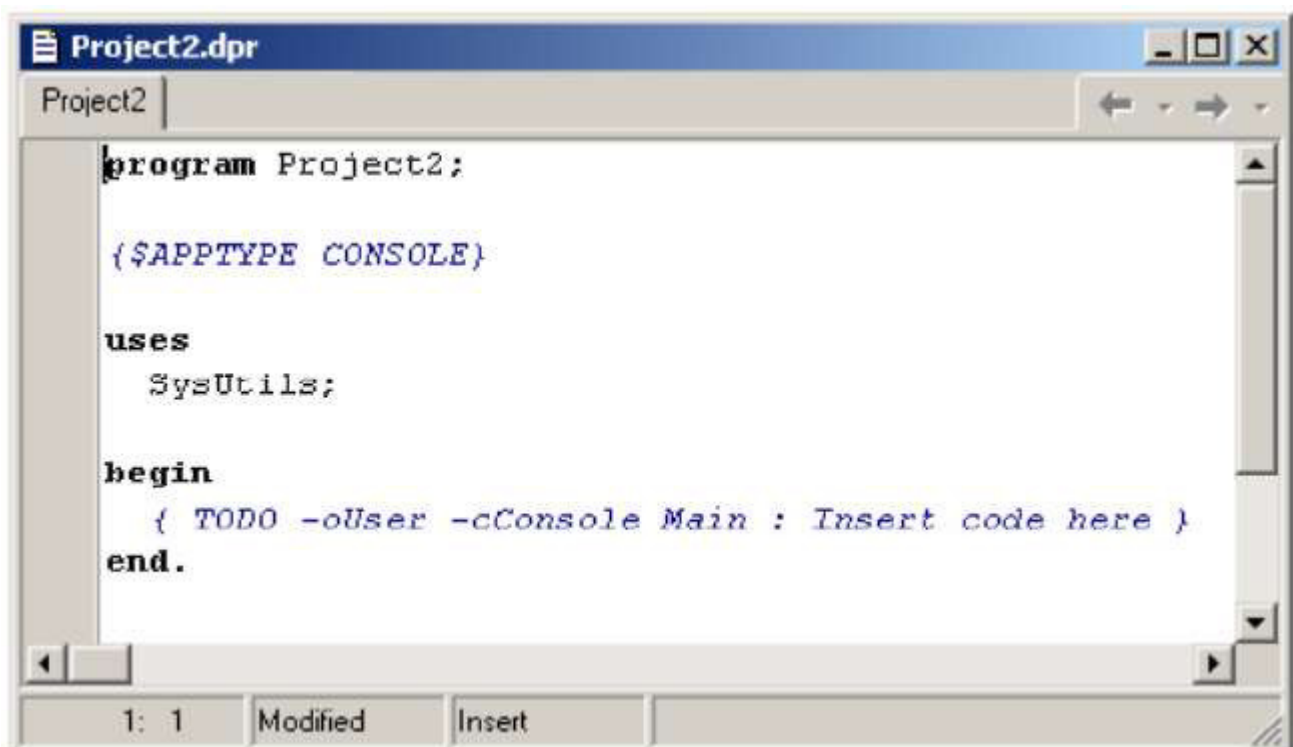
4. Сохраните проект, запустите и протестируйте его.

Задание 2 Цель работы - создать консольную программу.

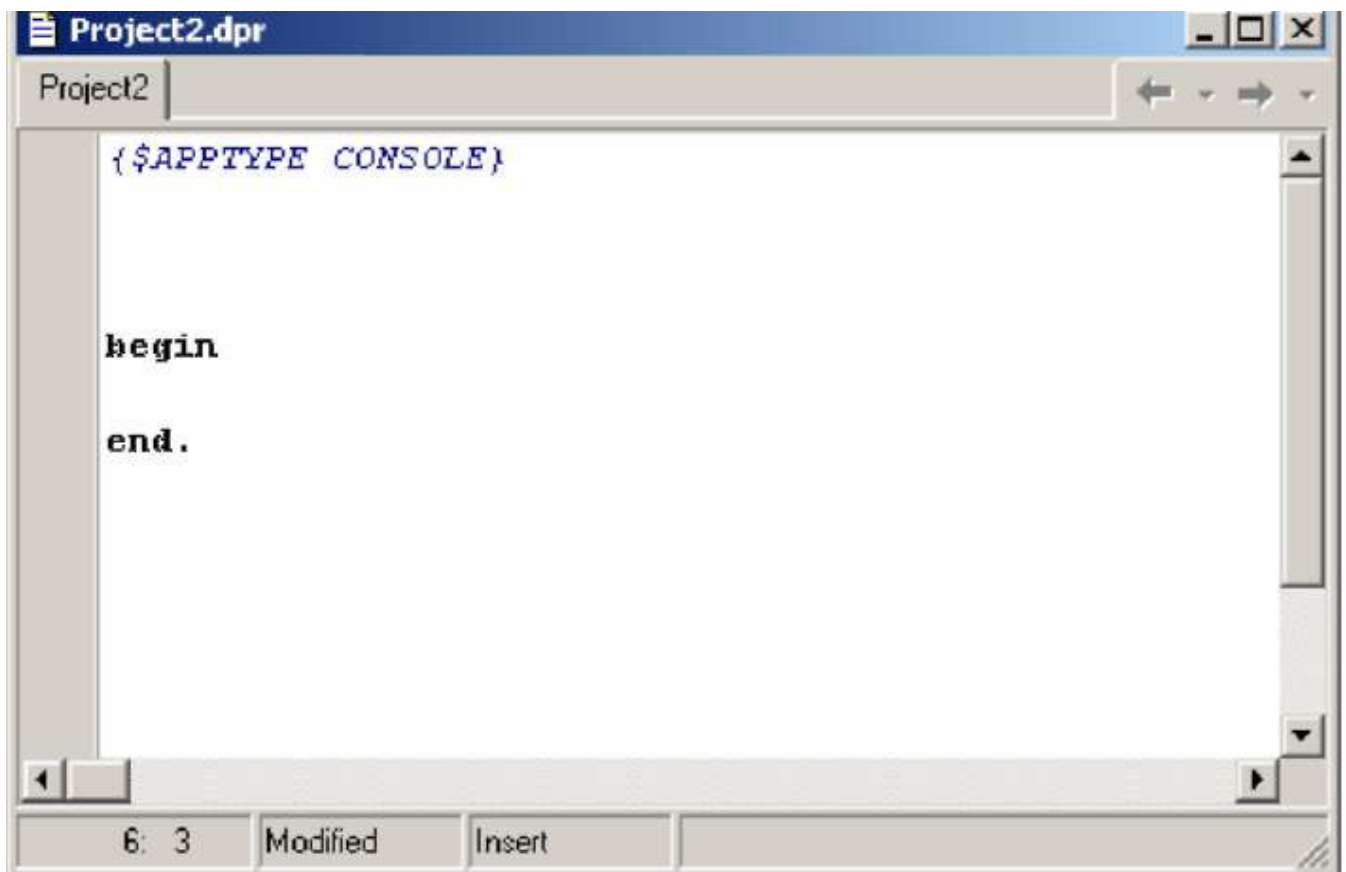
1. Дать команду главного меню File > New > Other (Файл > Новый > Другое), и в диалоговом окне на закладке New выбрать значок Console Application (Консольное приложение).



2. После нажатия на кнопку ОК экран примет следующий вид:



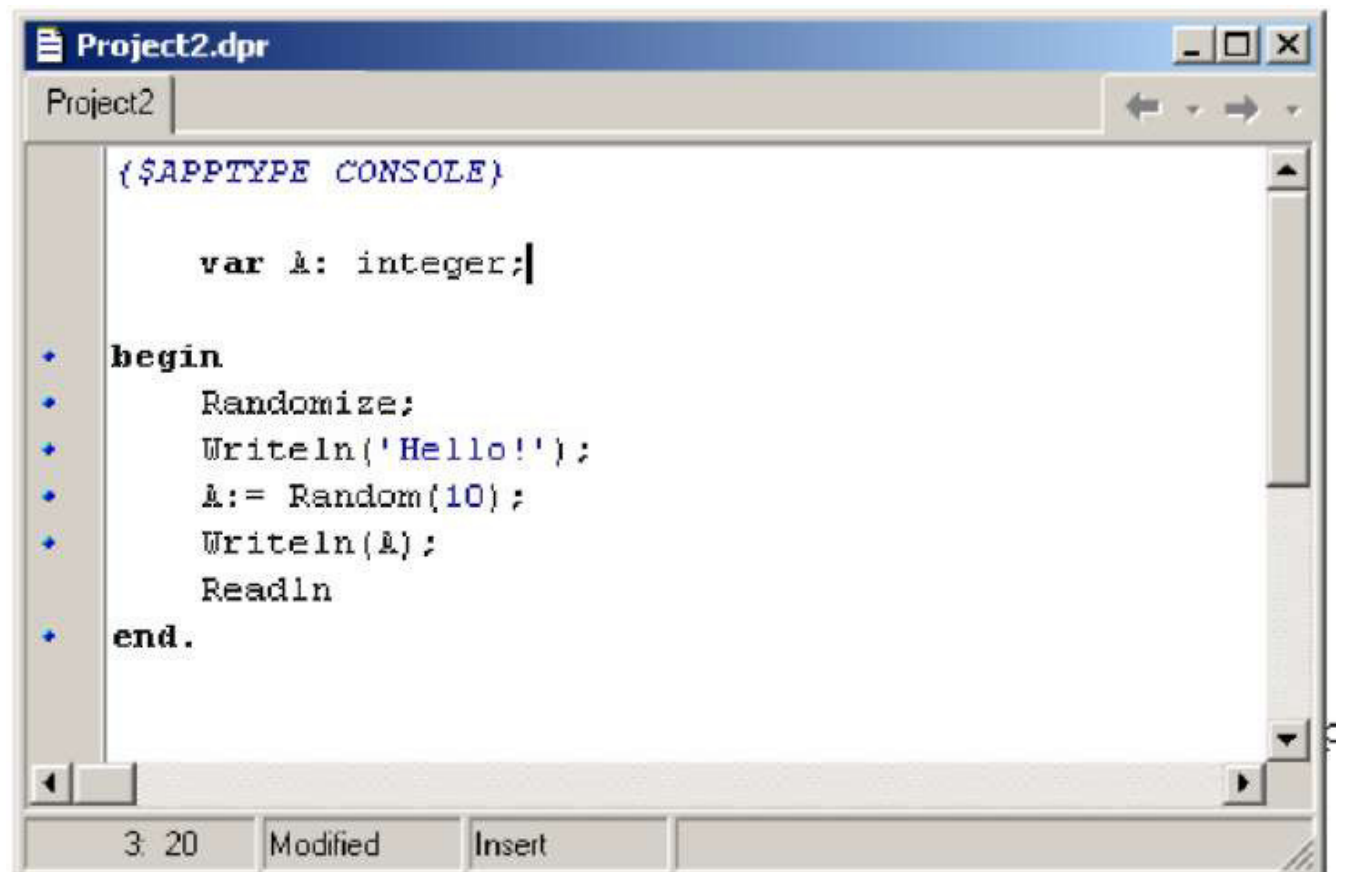
3. Текст был сгенерирован Delphi автоматически. Он представляет собой шаблон создаваемого приложения. Ничего изменять в этом тексте в принципе не надо - он соответствует готовой программе. Но внесем некоторые изменения.



The screenshot shows the Delphi IDE with the 'Project2.dpr' file open. The code editor contains the following text:

```
{$APPTYPE CONSOLE}  
  
begin  
  
end.
```

The status bar at the bottom indicates line 6, column 3, in 'Modified' mode.

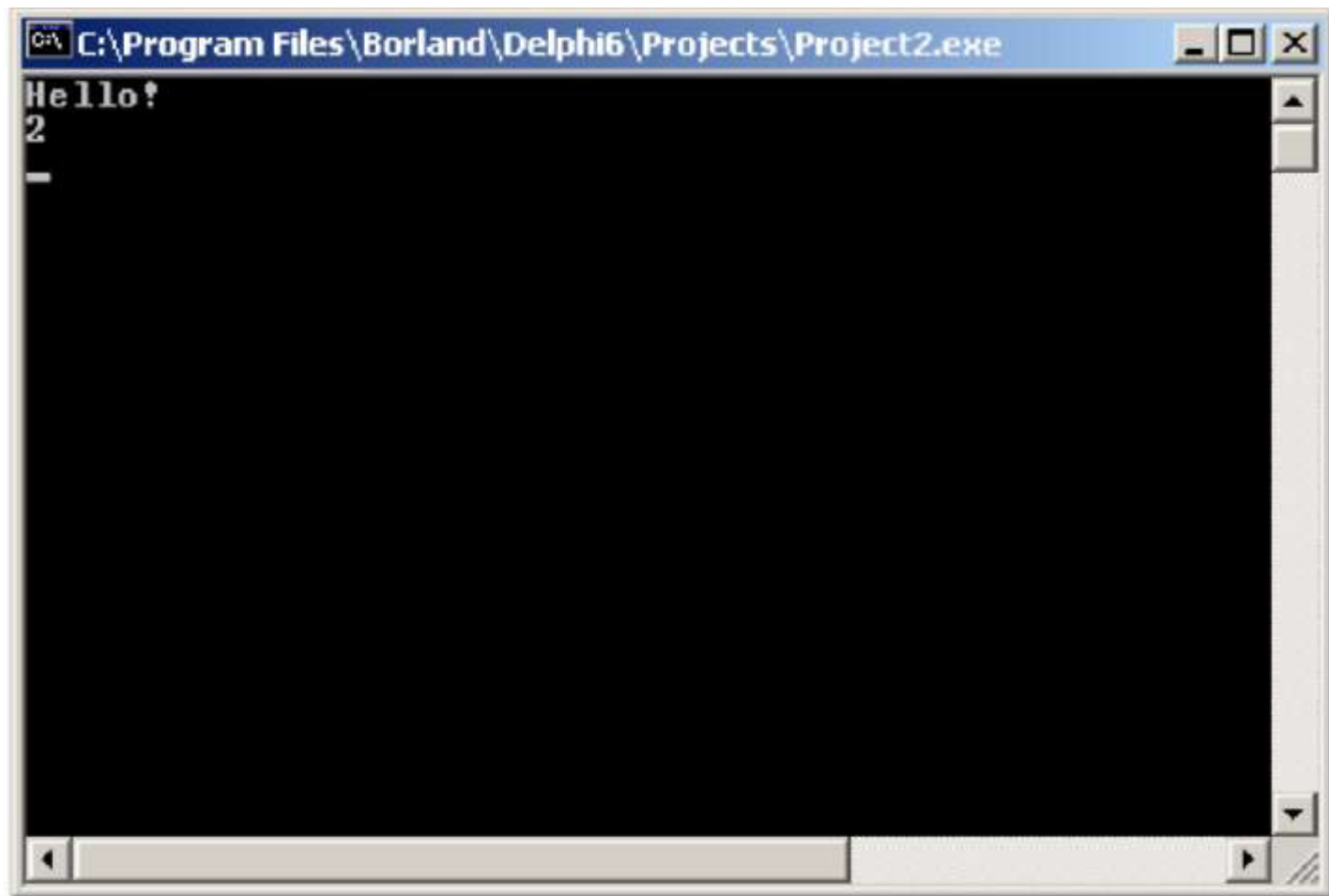


The screenshot shows the Delphi IDE with the 'Project2.dpr' file open. The code editor contains the following text:

```
{$APPTYPE CONSOLE}  
  
    var A: integer;  
  
* begin  
*     Randomize;  
*     Writeln('Hello!');  
*     A:= Random(10);  
*     Writeln(A);  
*     Readln  
* end.
```

The status bar at the bottom indicates line 3, column 20, in 'Modified' mode.

5. Выбрать команду Run. Получить результат. Нажать Enter для выхода в окно кода.



Практическая работа №20

Работа с меню

Цель работы -

Создать программу, выполняющую следующие действия:

1. После запуска программы в окне изображается строка меню (Файл, Выход).
2. При выборе пункта меню Файл появляются пункты меню (Рисунки, Выход).
3. При выборе пункта меню Рисунки появляется вложенное меню, состоящее из двух пунктов (Облака, Лес).



4. По щелчку правой кнопки мыши появляется контекстное меню.
5. Выбрать по пункту другой рисунок



6. Для выхода из программы необходимо щелкнуть мышью на закрывающей кнопке в строке заголовка.

7. Если выбрать любой из пунктов Выход, работа программы завершается.

Описание плана разработки программы

1. Открыть новый проект.

2. Разместить на форме экземпляры компонентов: панель Panel, рисунок Image, диалоговое окно OpenFileDialog.

3. Выполнить следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ имя события	Действие
Form1	Properties	Caption	Установка имени формы "Мое меню"
	Events	OnMouseDown	<pre>var p:TPoint; begin p.X :=X; p.Y :=Y; p := ClientToScreen (p);</pre>

			PopupMenu1.Popup (p.X, p.Y); end;
Запустить редактор меню (дважды щелкнуть на значке меню на форме)			
Form1.MainMenu1	Properties (в окне Object Inspector не выбран никакой объект)	Caption	Ввести текст пункта меню -Файл, и нажать Enter. Система присвоит ему имя N1
Между существующими и будущими пунктами меню можно переключаться с помощью щелчка мыши или курсорных клавиш.			
Form1.MainMenu1	Properties	Caption	Ввести текст пункта меню -Выход, и нажать Enter. Система присвоит ему имя N2.
	Events (щелкнуть на пункте Выход в строке меню)	N2Click	Close;
Щелкните на пункте Файл. Редактор меню создал еще одну заготовку под этим пунктом. Это заготовка для меню, которое откроется при выборе пункта Файл в работающей программе. Используя заготовки, создайте в этом меню два пункта: Рисунки (система присвоит ему имя N3) и Выход (N4). Выберите в редакторе меню пункт Рисунки и нажмите комбинацию клавиш Ctrl + Вправо.			
N4: TMenuItem	Events	OnClick	Выберем из раскрывающегося списка уже существующую процедуру-обработчик N2Click
Form1.MainMenu1	Properties	Caption	Ввести текст пункта меню -Облака, и нажать Enter. Система присвоит ему имя N5.
N5: TMenuItem	Events (выбрать в строке меню на форме пункт Облака)	OnClick	Image1.Picture.LoadFromFile ('C:\Windows\Облака.bmp');
Form1.MainMenu1	Properties	Caption	Ввести текст пункта меню - Лес, и нажать Enter. Система присвоит ему имя N6.
N6: TMenuItem	Events (выбрать в строке меню на форме пункт Лес)	OnClick	Image1.Picture.LoadFromFile ('C:\Windows\Лес.bmp');
Закройте окно редактора меню и убедитесь, что теперь строка меню появилась в основной форме программы.			

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ имя события	Действие
PopupMenu (Вкладка Standard)	Properties	Caption	Ввести текст пункта меню -Облака, и нажать Enter. Система присвоит ему имя N7.
		Caption	Ввести текст пункта меню - Лес, и нажать Enter. Система присвоит ему имя N8.
N7	Events	OnClick	Выберем из раскрывающегося списка уже существующую процедуру-обработчик N5Click
N8	Events	OnClick	Выберем из раскрывающегося списка уже существующую процедуру-обработчик N6Click
Image (Вкладка Additional)	Properties	Stretch	Присвоить значение True

4. Сохраните проект, запустите и протестируйте его.

Листинг подпрограммы

```

procedure TForm1.N2Click (Sender: TObject);
begin
Close;
end;
procedure TForm1.N5Click (Sender: TObject);
begin
Image1.Picture.LoadFromFile ('C:\Windows\Облака.bmp');
end;
procedure TForm1.N6Click (Sender: TObject);
begin
Image1.Picture.LoadFromFile ('C:\Windows\Лес.bmp');
end;

```

procedure TForm1.FormMouseDown

```

(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState;
X, Y: Integer); var p:TPoint;
begin
p.X :=X;
p.Y :=Y;
p := ClientToScreen (p);

```

```
PopupMenu1.Popup (p.X, p.Y);  
end;
```

Практическая работа №21

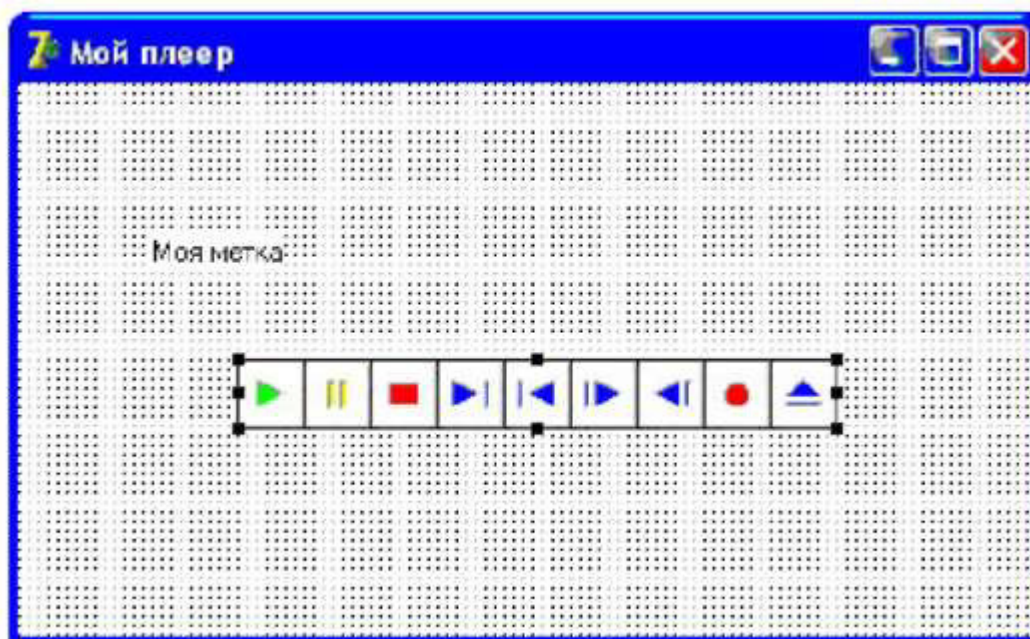
Создание приложения «Музыкальный плеер»

Цель работы - создать программу, выполняющую следующие действия:

1. После запуска программы в окне изображается музыкальный проигрыватель.
2. По щелчку мышью на кнопке "Play" воспроизвести выбранную мелодию.
3. Для выхода из программы необходимо щелкнуть мышью на закрывающей кнопке в строке заголовка.

Описание плана разработки программы

1. Открыть новый проект.
2. Разместить на форме экземпляры компонентов: медиаплеер MediaPlayer, рисунок Image.



3. Выполнить следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/имя события	Действие
Form1	Properties	Caption	Установка имени формы "Музыкальный проигрыватель"
MediaPlayer1 (Вкладка System)	Properties	AutoOpen	Выберите значение True из раскрывающегося списка
		FileName	Указать не полный путь к файлу, а относительно местоположения программы. Например, "Prim.wav", то программа будет этот файл искать в папке, где сама находится.

4. Сохраните проект, запустите и протестируйте его.

5. Создать проигрывателю картинку-фон, например:



6. Сохранить картинку-фон в файле, например, Фон.Бшр

7. Вставить в плеер в качестве фона.

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/имя события	Действие
Image1	Properties	Picture AutoSize	С помощью кнопки Load выберите ваш файл Фон.Бтр и нажмите ОК. В результате на месте пунктирной каемки на форме появится этот рисунок из файла. Выбрать значение True

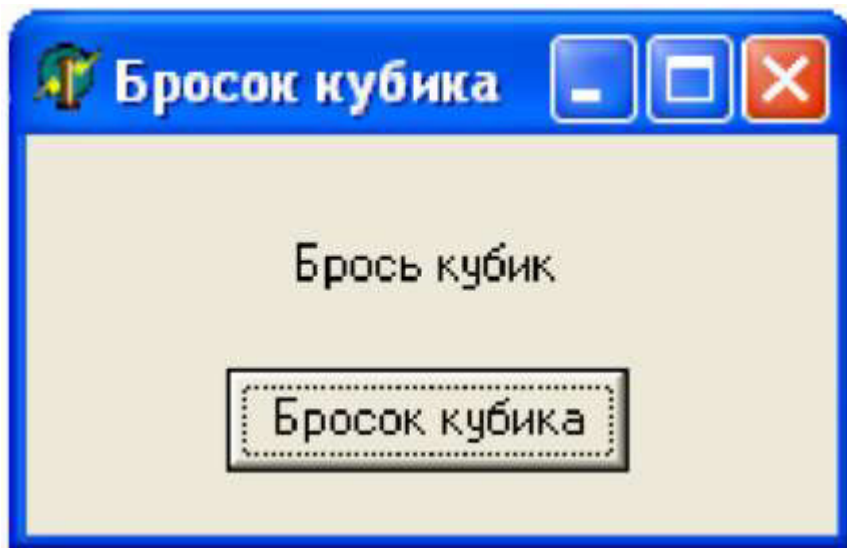
8. Подобрать размер формы под размер изображения, само изображение установите в верхний левый угол, панель с кнопками медиаплеера поставить так, чтобы гармонировала с рисунком фона, например:



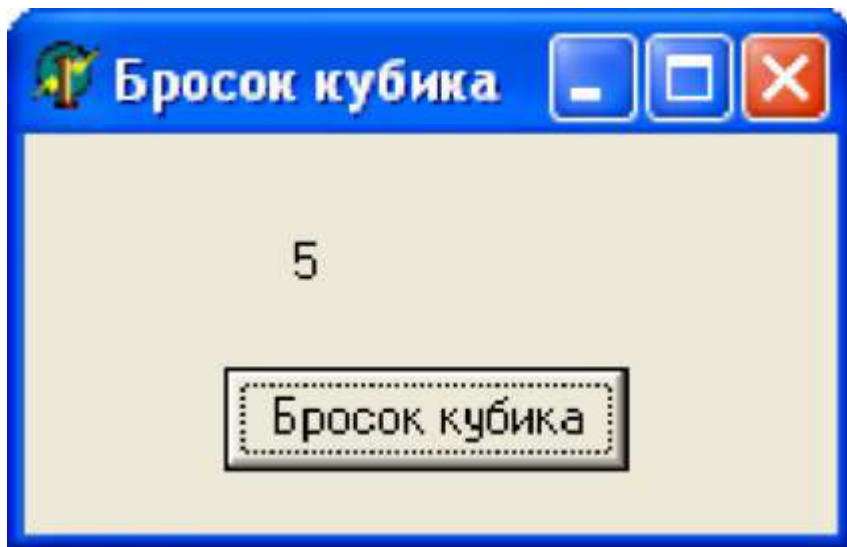
Практическая работа №22

Случайный выбор из списка

Цель работы - создать программу, выполняющую следующие действия: 1. После запуска программы появляется надпись "Брось кубик".



2. По щелчку мышью на кнопке "Бросок кубика" появляется сообщение, выдающее числа-очки в диапазоне 0 - 6.



3. Для выхода из программы необходимо щелкнуть мышью на закрывающей кнопке в строке заголовка.

Описание плана разработки программы

1. Открыть новый проект.
2. Разместить на форме экземпляры компонентов: метку Label и кнопку Button.
3. Выполнить следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ имя события	Действие
Form1	Properties	Caption	Установка имени формы "Бросок кубика"
Label1	Properties	Caption	Ввод текста надписи "Брось кубик"
Button1	Properties	Caption	Установка имени кнопки "Бросок кубика"
	Events	OnClick	Label1.Caption := IntToStr (n) ;

4. Сохраните проект, запустите и протестируйте его. Фрагмент программы

```
procedure TForm1.Button1Click(Sender: TObject); var n: Integer;
begin
n := random (6) + 1 ;
```

Label1.Caption := IntToStr (n) ; end; end.

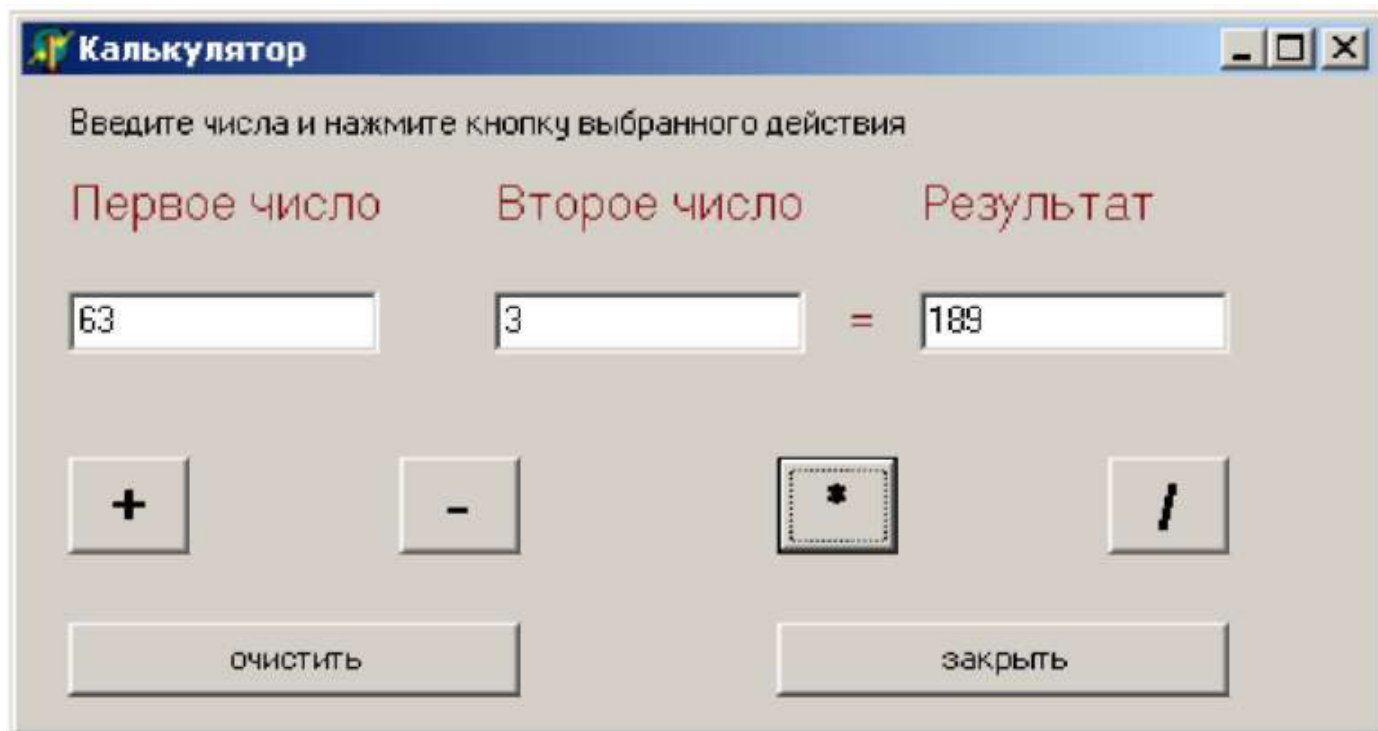
```
5. Создать функцию, отображающую числовые значения граней. Function Kubic (Sides:Integer): Integer;
begin
If Sides >= 1 then begin
Result := random (Sides) +1 ; end else begin
Result :=0;
end;
End;
```

Практическая работа №23

Разработка приложения «Обычный калькулятор»

Цель работы - создать программу, выполняющую действия обычного калькулятора. Описание плана разработки программы

1. Открыть новый проект.
2. Разместить на форме экземпляры компонентов: Label, Edit, Button.



3. Сохраните проект, запустите и протестируйте его.

Листинг подпрограммы

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Edit1.Text := ''; Edit2.Text := ''; Edit3.Text := '';
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
  Edit1.Text := ''; Edit2.Text := ''; Edit3.Text := '';
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
  close;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  a := StrToFloat (Edit1.Text);
```

```

b := StrToFloat (Edit2.Text);
c := a + b;
Edit3.Text := FloatToStr (c);
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
a := StrToFloat (Edit1.Text);
b := StrToFloat (Edit2.Text);
c := a - b;
Edit3.Text := FloatToStr (c);
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
a := StrToFloat (Edit1.Text);
b := StrToFloat (Edit2.Text);
c := a * b;
Edit3.Text := FloatToStr (c);
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
a := StrToFloat (Edit1.Text);
b := StrToFloat (Edit2.Text);
if b = 0 then Edit3.Text := 'division by zero' else begin
c := a / b;
Edit3.Text := FloatToStr (c);
end;
end;
end;

```

Задание для самостоятельного выполнения

1. Создать программу, выполняющую действия простого инженерного калькулятора.



```

Фрагмент подпрограммы (для вычисления функции Cos(x)) if Edit1.Text <> '' then begin
a := StrToFloat (Edit1.Text);
c := cos (a);
end;
if Edit2.Text <> '' then
begin
b := StrToFloat (Edit2.Text);
c := cos (b);
end;

```

Edit3.Text := FloatToStr (c); Добавить на форму кнопки для вычисления функций Tg(x), Log ab.

2. Создать программу, выполняющую перевод чисел из десятичной системы счисления в двоичную систему и обратно.

Form1

Десятичная сис. счисления	Ok	Двоичная сис. счисления
34		100010
Двоичная сис. счисления	Ok	Десятичная сис. счисления
1010101		85

Exit Clear

Практическая работа №24

Нахождение минимального и максимального числа в массиве

Цель работы - создать программу, которая находит минимальное и максимальное числа в введенном массиве.

Описание плана разработки программы

1. Открыть новый проект.
2. Разместить на форме экземпляры компонентов: Button, Edit, Label.
3. Выполнить следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ имя события	Действие
Form1	Properties	Caption	Установка имени формы "Поиск"
	Events	OnCreate	Очистить значения свойств Text текстовых полей
Button1	Properties	Caption	Введите название "Очистить"
	Events	OnClick	Очистить значения свойств Text текстовых полей
Button2	Properties	Caption	Введите название "Закрыть"
	Events	OnClick	Обработка события закрытия формы
Button3	Properties	Caption	Введите название "Поиск"
	Events	OnClick	Обработка события нахождения минимального и максимального чисел во введенном массиве
Edit1	Properties	Caption	Очистить значение свойства Text
Edit2	Properties	Caption	Очистить значение свойства Text
Edit3	Properties	Caption	Очистить значение свойства Text

4. Введите переменные

s, ss: string ; a: array [1..15] of integer; l, j, k, max, min, p, code : integer.

5. Сохраните проект, запустите и протестируйте его. Листинг подпрограмм

```
procedure TForm1.FormCreate(Sender: TObject);
begin
Edit1.Text := ''; Edit2.Text := ''; Edit3.Text := '';
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```

Edit1.Text := ''; Edit2.Text := ''; Edit3.Text := '';
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
close;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
s :=Edit1.Text;
s := concat (s, #32);
i := 0;

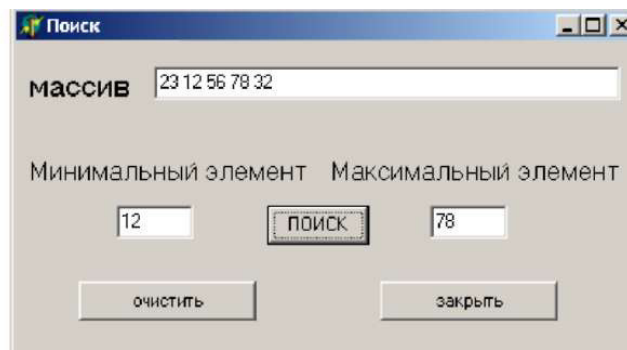
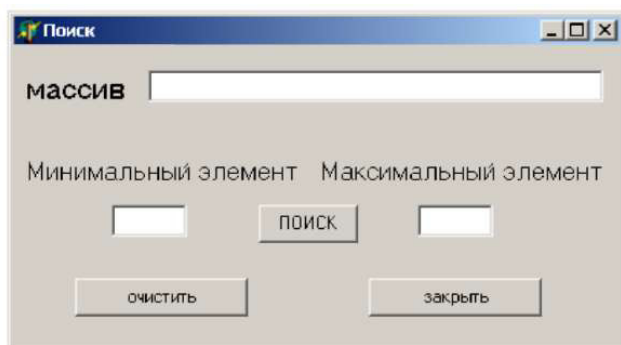
```

while Length(s) > 0 do begin i :=

```

p := pos (#32,s);
ss := copy (s,1,p-1);
Val (ss,k,code);
a[i] := k;
delete(s,1,p);
end;
max := a[1]; For j := 1 to i do
if max <
a[j] then max := a[j]; min := a[1]; For j := 1 to i do
if min >
a[j] then min := a[j]; Edit3.Text := IntToStr (max);
Edit2.Text := IntToStr (min);
end;

```



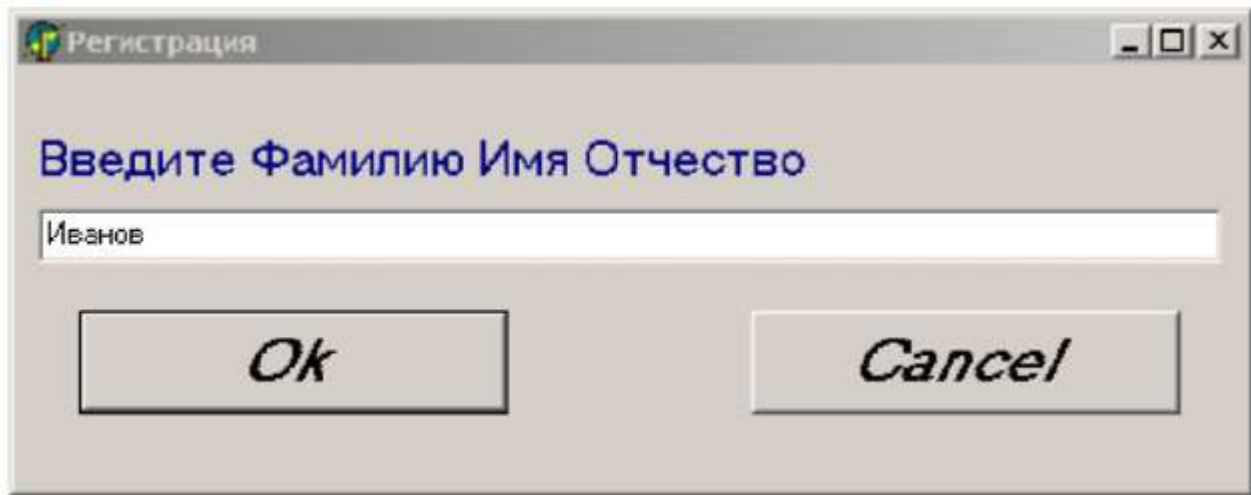
Практическая работа №25-26

Разработка приложения «Тест»

Цель работы - создать программу, которая тестирует учащегося по информатике и математике.

Проект должен содержать последовательность форм, реализующих диалог с тестируемым учащимся.

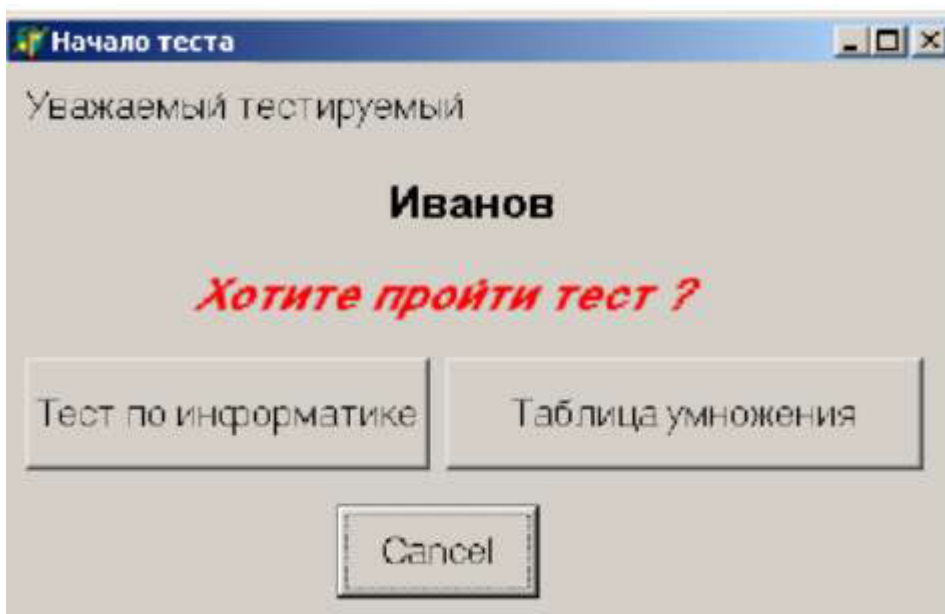
На первой форме происходит регистрация учащегося.



Фрагмент программы (unit1): uses Unit2; {\$R *.dfm}

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Close;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
  Form2.Label3.Caption:=Form 1.Edit1.Text;
  Form2.ShowModal;
end;
```

На второй форме предлагается выбрать один из тестов.

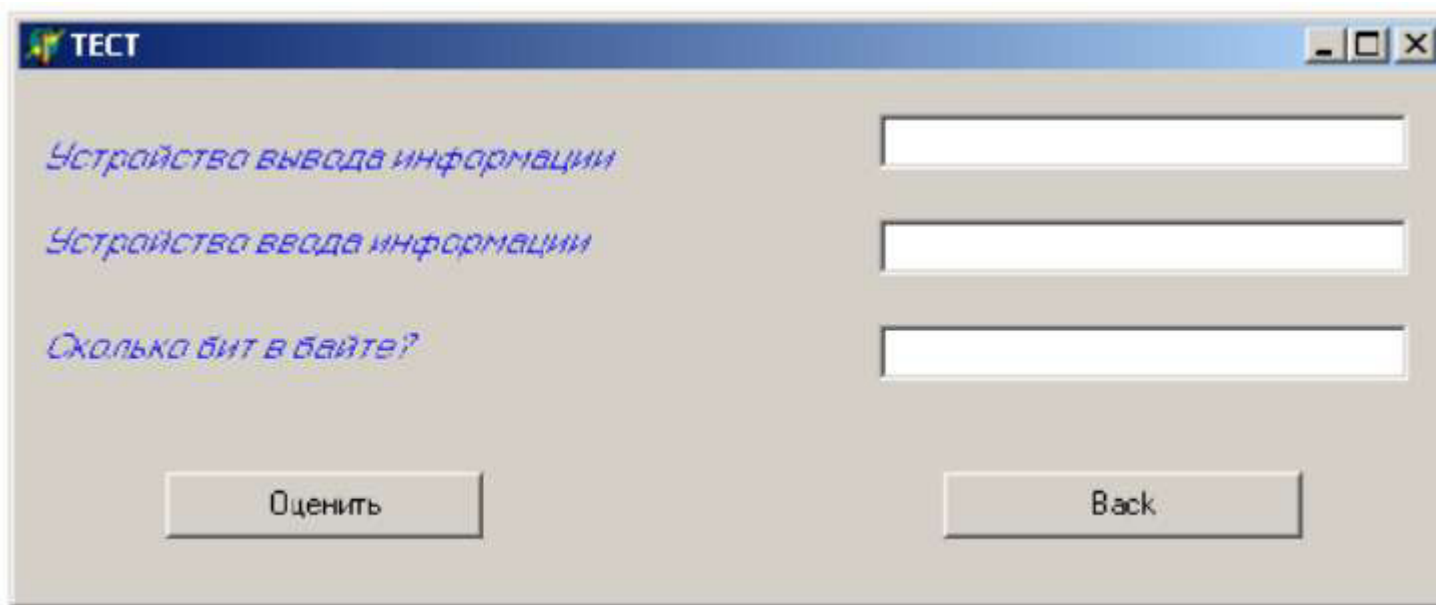


Фрагмент программы (unit2):

```
uses Unit3, Unit6; {$R *.dfm}
```

```
procedure TForm2.Button1Click(Sender: TObject);  
begin  
Form3.ShowModal;  
end;  
procedure TForm2.Button2Click(Sender: TObject);  
begin  
Form2.Close;  
end;  
procedure TForm2.Button3Click(Sender: TObject);  
begin  
Form6.Edit2.Text:="";  
Form6.ShowModal;  
end;
```

На третьей форме предлагается проути тест по информатике.



Фрагмент программы (unit3): uses Unit4, Unit2; {\$R *.dfm}

```
procedure TForm3.Button1Click(Sender: TObject);  
begin  
k:=0;
```

```
if (Form3.Edit1.Text='монитор') or (Form3.Edit1.Text='Монитор') or  
(Form3.Edit1.Text='МОНИТОР') then k:=k+1; if (Form3.Edit2.Text='Клавиатура') or  
(Form3.Edit2.Text='клавиатура')
```

```

or (Form3.Edit2.Text='КЛАВИАТУРА') then k:=k+1; if Form3.Edit3.Text='8' then k:=k+1;
Form4.Label2.Caption:=IntToStr(k); if k=0 then Form4.Label1.Caption:='Очень плохо' else
if k=1 then Form4.Label1.Caption:='Плохо' else
if k=2 then Form4.Label1.Caption:='Хорошо' else

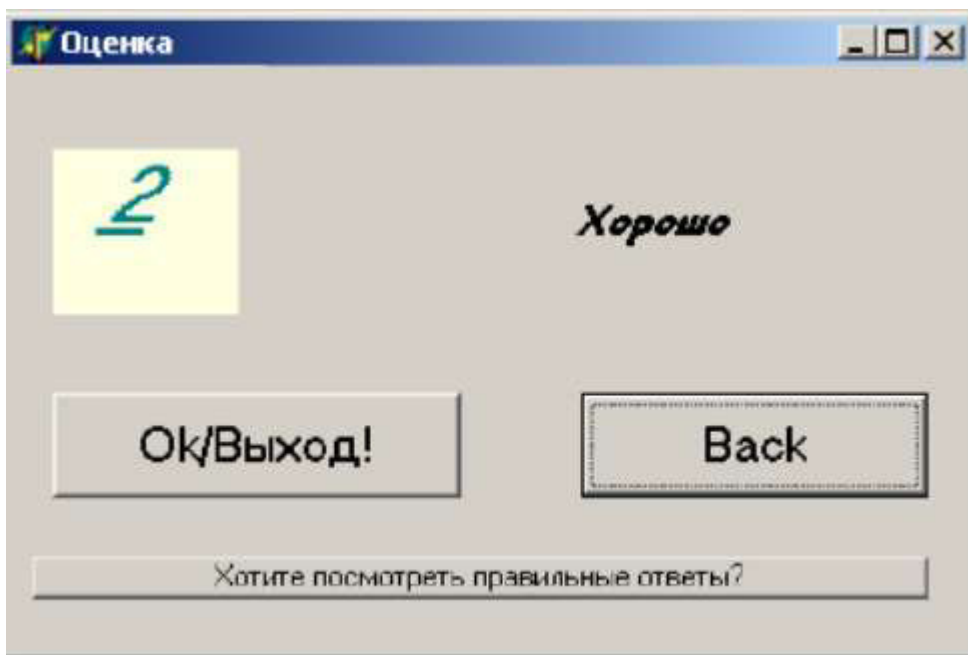
```

```

if k=3 then Form4.Label1.Caption:='Очень хорошо';
Form3.Edit1.Text:='';
Form3.Edit2.Text:='';
Form3.Edit3.Text:='';
Form4.ShowModal;
end;
procedure TForm3.Button2Click(Sender: TObject);
begin
Form3.Close;
end;

```

На следующей форме отображается результат тестирования и предложение о просмотре ответа.



Фрагмент программы (unit4): uses Unit1, Unit5, Unit3, Unit2; {\$R *.dfm}

```

procedure TForm4.Button2Click(Sender: TObject);
begin
Form4.Close;
end;
procedure TForm4.Button1Click(Sender: TObject);
begin
Form4.Close;
Form3.Close;

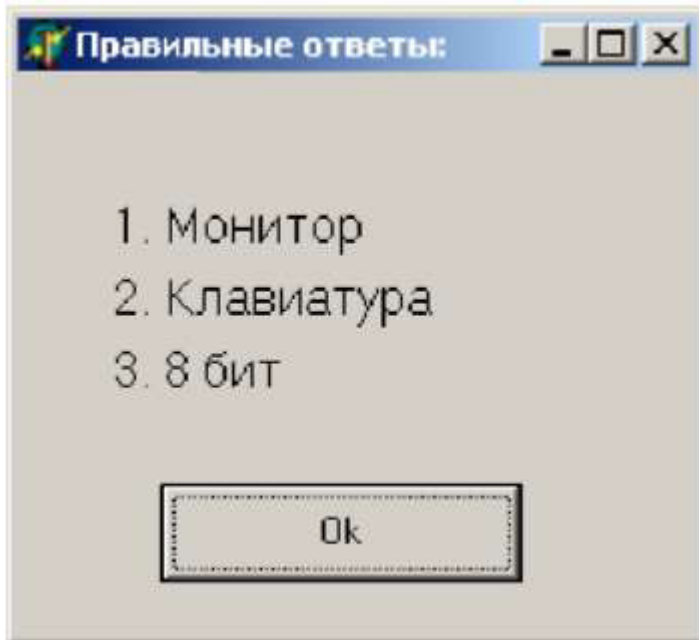
```

```

end;
procedure TForm4.Button3Click(Sender: TObject);
begin
Form5.ShowModal;
end;

```

На следующей форме отображаются правильные ответы.



Фрагмент программы (unit5):

```

uses Unit4, Unit3, Unit2, Unit1;

```

```

{$R *.dfm}

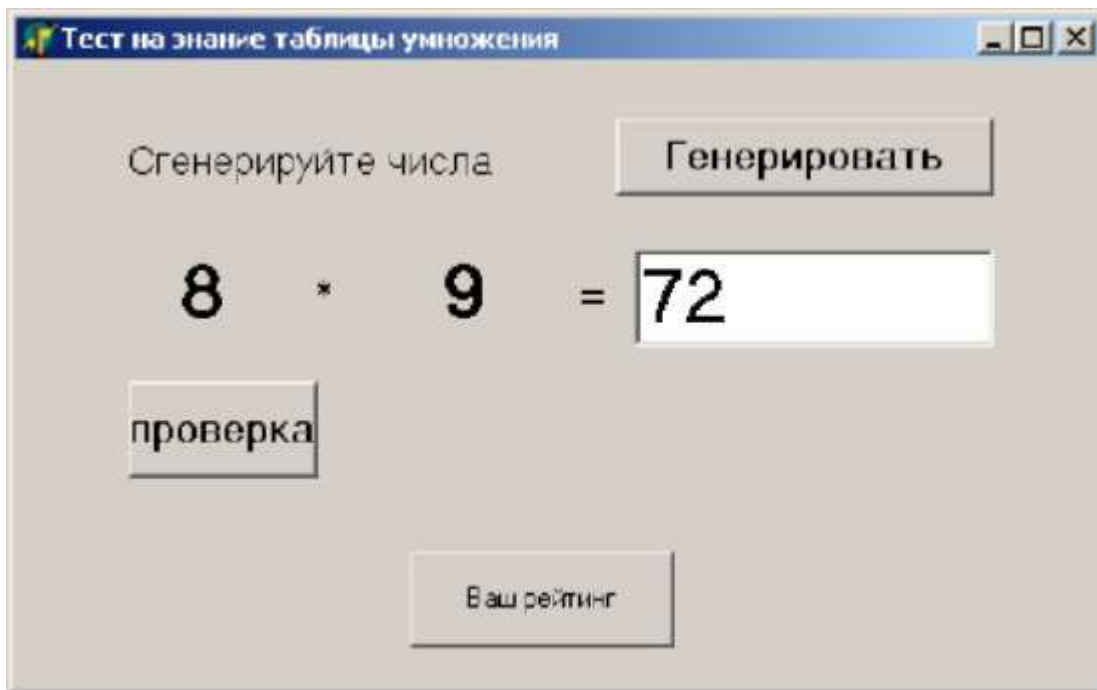
```

```

procedure TForm5.Button1Click(Sender: TObject);
begin
Form4.Close;
Form3.Close;
Form2.Close;
Form1.Close;
Form5.Close;
end;

```

Если учащийся выбрал тест по математике, то ему предлагается проверить свои знания таблицы умножения. На следующей форме случайным образом выбираются числа. Учащийся должен ввести значение произведения в текстовое поле. С помощью кнопки "проверка" выясняется правильность введенного ответа. Если ответ правильный, то можно сгенерировать следующий пример. После нескольких примеров можно проверить свой рейтинг.



Фрагмент программы (unit6): uses Unit7, Unit1; {\$R *.dfm}

```

procedure TForm6.Button1Click(Sender: TObject);
var n,i:integer;
begin
  randomize;
  a:=random(10)-0;
  b:=random(10)-0;
  Form6.Label1.Caption:=IntToStr(a);
  Form6.Label2.Caption:=IntToStr(b);
  Form6.Edit2.Text:='-'; Form6.Label6.Caption:='';
end;
procedure TForm6.Button2Click(Sender: TObject);
begin
  if (a*b=StrToInt(Form6.Edit2.Text)) then begin
    Form6.Label6.Caption:='Правильно';
    m:=m+1;
    r:=r+1;
    q:=q+1;
  end else begin
    Form6.Label6.Caption:='Не правильно'; r:=r-1;
    q:=q+1;
  end;
  Form6.Label1.Caption:=''; Form6.Label2.Caption:='';
end;
procedure TForm6.Button3Click(Sender: TObject); var
c:real;

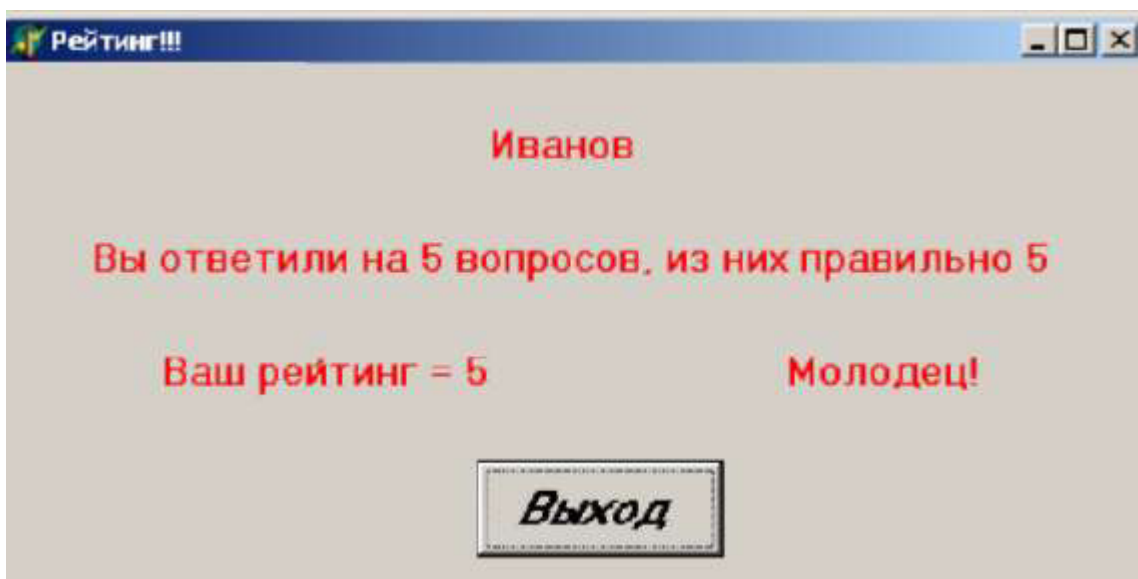
```

```

begin
Form7.Label2.Caption:=Form1.Edit1.Text;
Form7.Label1.Caption:='Вы ответили на '+IntToStr(q)+' вопросов, из них правильно '+IntToStr(
m);
Form7.Label4.Caption:='Ваш рейтинг = '+IntToStr(r);
c:=m/q;
if c=0 then Form7.Label3.Caption:='Очень плохо' else if (c>0)and(c<0.5) then Form7.Label3.Ca
ption:='Плохо' else if c=0.5 then Form7.Label3.Caption:='Надо доучить' else if (c>0.5) and (
c<1) then Form7.Label3.Caption:='Хорошо' else if c=1 then Form7.Label3.Caption:='Молодец!';
Form7.ShowModal;
end;

```

При нажатии на кнопку "Ваш рейтинг" на следующей форме появляется результаты рейтинга. Тестирование можно завершить.



Фрагмент программы (unit7): uses Unit6; {\$R *.dfm}

```

procedure TForm7.Button1Click(Sender: TObject);
begin
Form7.Close;
Form6.Close;
end;

```

Обратите внимание на подключение модулей в строке Uses. Таким образом, происходит обращение к соответствующей форме. Для отображения формы используется функция

```
function ShowModal: Integer;
```

Данная функция позволяет показывать форму в работе режима диалога.

Перечень учебных изданий, интернет - ресурсов, дополнительной литературы:

Основная литература:

Фризен, И.Г. Основы алгоритмизации и программирования : учеб. пособие / И.Г. Фризен. — М. : ФОРУМ : ИНФРА-М, 2017. — 392 с. — (Среднее профессиональное образование). - Режим доступа: <http://znanium.com/catalog/product/559358> (ЭБС znanium.com - Договор № 4971 от 11.01.2021 г. действителен до 31.12.2021 г.)

Дополнительная литература:

Голицына, О.Л. Основы алгоритмизации и программирования : учебное пособие /, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2020. — 431 с. — (Среднее профессиональное образование). - Текст : электронный. - URL: <http://znanium.com/catalog/product/1072040> (ЭБС znanium.com - Договор № 4971 от 11.01.2021 г. действителен до 31.12.2021 г.)

Интернет-источники:

Научно-технический и научно-производственный журнал «Информационные технологии» [Электронный ресурс]. — Режим доступа: URL: <http://www.novtex.ru/IT/>

Национальный Открытый Университет «ИНТУИТ» [Электронный ресурс]. — Режим доступа: URL: <http://www.intuit.ru/>

Журнал CHIP [Электронный ресурс]. — Режим доступа: URL: <http://www.ichip.ru/>

Журнал "ComputerBild" [Электронный ресурс]. — Режим доступа: URL: <http://www.computerbild.ru>