

Среда Программирования ABC Pascal

Задание

1. Ознакомиться с презентацией на тему «Среда программирования языка ABC Pascal»
2. Установить программу Pascal ABC себе на компьютер. Скачать ее можно, перейдя по ссылке: <http://pascalabc.net/ssyilki-dlya-skachivaniya>

Система программирования

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ.

В них входят:

- **компилятор или интерпретатор**
- **интегрированная среда разработки**
- **средства создания и редактирования текстов программ**
- **обширные библиотеки стандартных программ и функций**
- **средства отладки, помогающие устранять ошибки в программе**
- **«дружественная» к пользователю диалоговая среда;**
- **многооконный режим работы**
- **мощные графические библиотеки**
- **встроенный ассемблер**
- **встроенная справочная служба**

Система программирования PascalABC

Система PascalABC является мультипарадигменной, т.е. в ней можно программировать в процедурном, объектно-ориентированном и функциональном стилях.

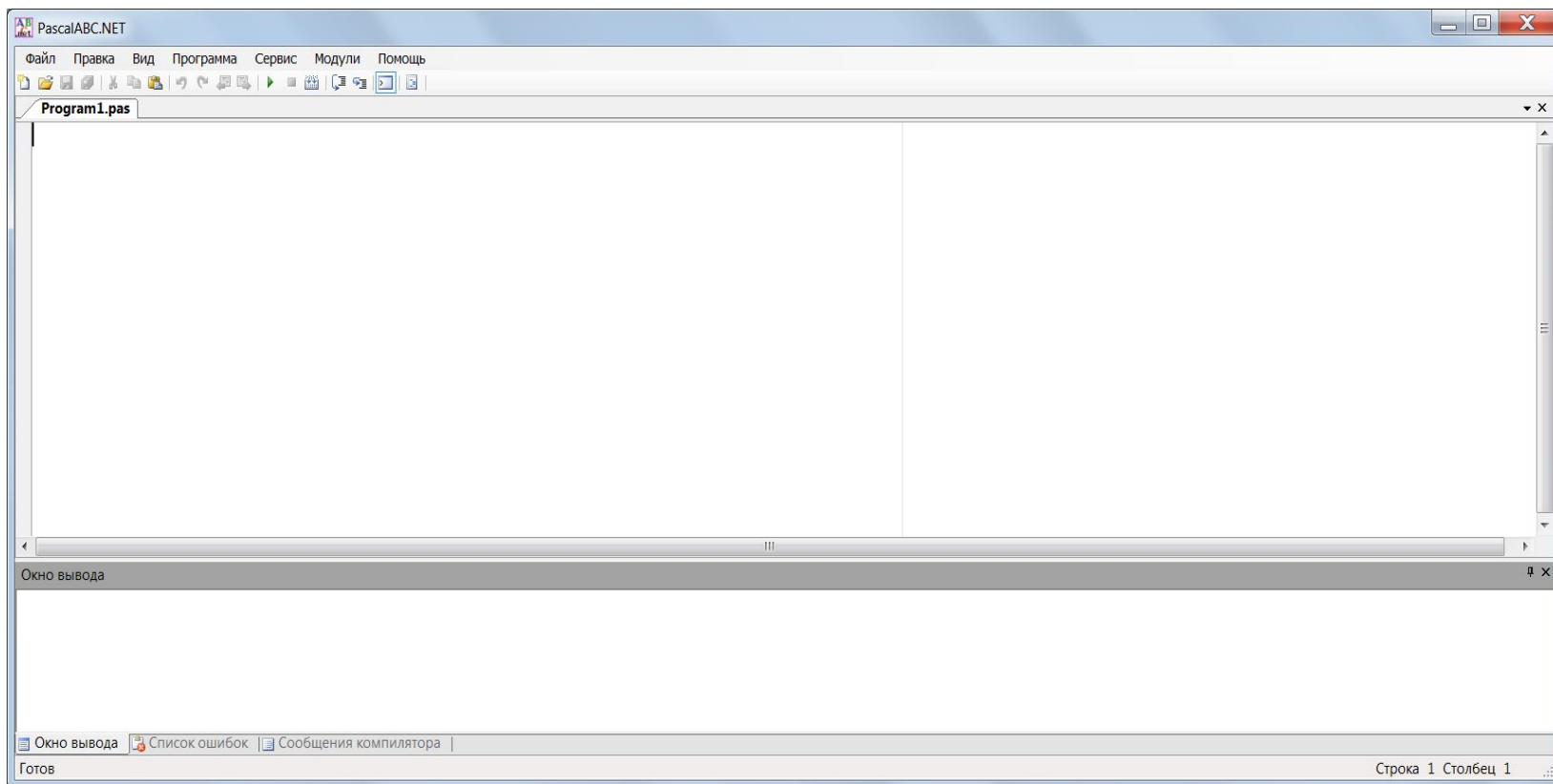
В состав системы входят следующие стандартные модули:

- GraphABC – растровая графика
- ABCObjects – векторная графика
- FormsABC – создание простых оконных приложений
- Arrays – работа с одно- и двумерными динамическими массивами
- Collections – содержит упрощенные классы коллекций
- исполнители Робот и Чертёжник (школьная информатика)

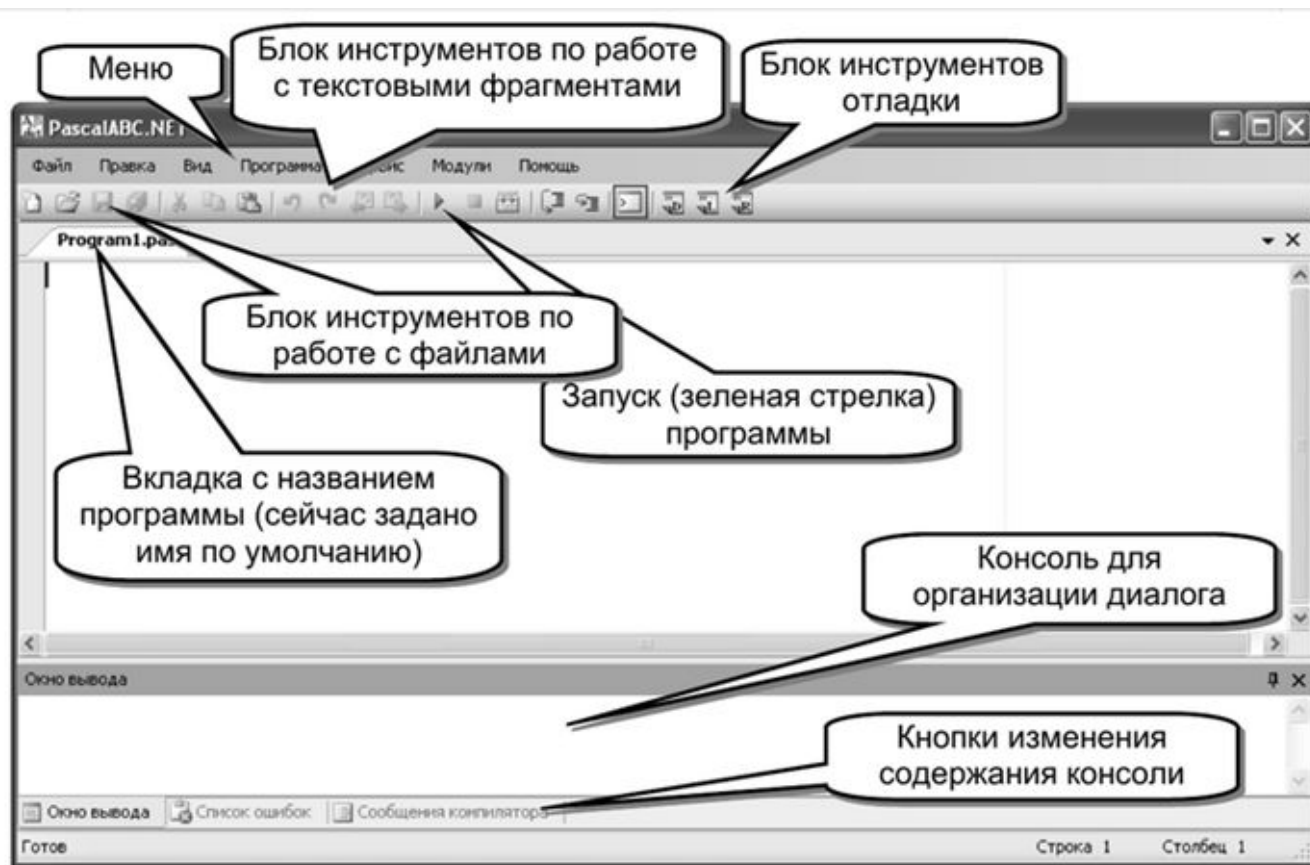
Установка и запуск системы PascalABC

После установки ярлык программы  появится на Рабочем столе и в меню «Пуск».

Дважды кликнув по этому ярлыку, запускаем программу:



Структура окна среды Pascal ABC



Структура окна среды Pascal ABC

Алфавит языка Pascal

Алфавит ЯП – это совокупность всех допустимых символов, которые можно использовать в этом языке.

Алфавит языка Pascal включает в себя следующие символы:

- прописные и строчные буквы латинского алфавита от A до z, а также символ подчеркивания `_`, который тоже считается буквой.

Прописные и строчные буквы равнозначны!

- арабские цифры 0 1 2 3 4 5 6 7 8 9
- специальные одиночные знаки + - * / = < > . , : ; ^ \$ # @
- специальные парные знаки () [] { } ‘ ‘
- составные знаки <= >= <> .. (* *) (..)

Также используются буквы русского алфавита, но только при вводе/выводе текста, заключенного в апострофы (`'`) или в комментариях к программе.

Структура программы на языке Pascal

program имя программы; {заголовок программы}

uses {раздел подключения модулей}

Label {раздел описания меток}

Const {раздел описания констант}

Type {раздел описания типов}

Var {раздел описания переменных}

Function ...; {раздел описания функций}

Procedure ...; {раздел описания процедур}

} Блок
описаний

BEGIN

...

END.

} Раздел операторов

Структура программы на языке Pascal

Первая строка – это **заголовок программы**, не обязателен.

Далее следует **раздел подключения модулей**, где указываются через запятую подключаемые к программе модули. Также может отсутствовать.

Далее идет **блок описаний**, состоящий из пяти разделов:

1. описание меток
2. описание констант
3. описание типов
4. описание переменных
5. описание процедур и функций

В этом списке только раздел описания переменных является обязательным, остальные могут отсутствовать.

Структура программы на языке Pascal

Далее со служебного слова **BEGIN** начинается раздел операторов, которые отделяются друг от друга точкой с запятой «;».

Конструкция **begin ... end** называется операторными скобками, операторы, находящиеся внутри этой конструкции, считаются одним составным оператором.

Вся программа завершается словом **END** с точкой.

Идентификаторы

Идентификаторы – это имена переменных, констант, меток, типов, модулей, процедур и функций.

Имена задает разработчик программы.

На идентификаторы накладываются ограничения:

- нельзя использовать служебные слова
- имя должно начинаться с буквы и может содержать латинские буквы, цифры и знаки подчеркивания

Пример:

`a1, b_2, k123, _d` – идентификаторы

`1a, и2, @ru, integer, var` – не идентификаторы



Желательно выбирать имена, несущие смысловую нагрузку, например, **result, summa, price**. Это делает программу проще для понимания.

Константы

Константа (постоянная) – это величина, значение которой не изменяется при исполнении программы.

Раздел описания констант начинается со служебного слова **const**, после которого следуют элементы описания:

ИМЯ КОНСТАНТЫ = значение;

ИЛИ

ИМЯ КОНСТАНТЫ : тип = значение;

Пример:

```
const
```

```
    Pi = 3.14;
```

```
    Number = 10;
```

```
    Name = 'Victor';
```

```
    Cifra = ['0'..'9'];
```

```
    Mass: array [1..5] of integer = (1,3,5,7,9);
```

Компьютер "знает", чему равны константы **e** и **π**.

Переменные

Переменная – одно из главных понятий в программировании.

Переменная – это величина, значение которой меняется при исполнении программы.

После объявления переменных программа выделяет определенное (в зависимости от типа данных) количество ячеек в памяти для хранения этих переменных.

Имена переменных присваиваются этим ячейкам, в которых затем хранятся значения переменных.

Храниться они могут или до конца выполнения программы или до тех пор, пока переменной не присвоится другое значение.

Имя переменной не изменяется до конца программы, а значение переменной может меняться.



В ЯП Pascal обязательное объявление переменных защищает программы от ошибок и повышает их надежность.

Оператор присваивания

Как и во всех ЯП в Pascal есть оператор присваивания `:=`, служит для задания значения переменной.

Если переменная уже имела какое-то значение, то оно стирается, и переменной присваивается новое значение.

Синтаксис оператора:

Переменная := выражение



В некоторых ЯП символом присваивания является знак равенства `=`. Чтобы не путать его с оператором сравнения, в Pascal введено обозначение `:=`.

Ввод данных

Выполняется операторами **read** и **readln** (read line – прочти строку).

read (A1 , A2) ;

readln ;

readln (A1 , A2) ;

где **A1**, **A2** – переменные, которым последовательно присваиваются вводимые с клавиатуры значения.

При выполнении оператора **read** программа останавливается и ждет ввода значений переменных. Если в операторе указано две и более переменных, то при наборе они печатаются через пробел.

При выполнении оператора ввода без параметров **readln** выполняется переход на новую строку.

При выполнении оператора **readln (A1 , A2)** вводятся значения всех переменных, а затем выполняется переход на новую строку.

Вывод данных

Выполняется операторами **write** и **writeln** (write line – напиши строку).

write(A1,A2); {Вывод значений A1 и A2 в одну строку}

writeln; {Переход на новую строку}

writeln(A1,A2); {Вывод A1 и A2 и переход на новую строку}

Ввод числовых данных с форматом

write(A:L:D);

Простейшая программа

название алгоритма

```
program qq;  
begin { начало программы }  
      { тело программы }  
end.  { конец программы }
```

комментарии в скобках {}
не обрабатываются



Что делает эта программа?

Вывод на экран

```
program qq;  
begin  
▶ write('2+');  
▶ writeln('2=?'); { на новую строку}  
▶ writeln('Ответ: 4');  
end.
```

Протокол:

2+

Ответ: 4

Задание наберите программу
в Pascal ABC и проверьте ее
работоспособность

Задача 1.

Выведем на экран приветствие:
"Добрый день".

Program prim1;

```
begin    {Выводим текст}  
    writeln('Добрый день');  
end.
```

Задача 2.

Ввод с клавиатуры значения переменной N

```
program Inp;  
  uses Crt;  
  var  
    N: integer;  
begin  
  ClrScr;  
  write('Введите число с клавиатуры:');  
  readln(N);  
  writeln('Вы ввели число ', N);  
  readln  
end.
```

Задача 3. Расчет скорости тела при падении с башни

```
• Program Piza;  
  const crt;  
    G=9.8;  
  var  
    V,H,N:real;  
begin  
  
  clrscr;  
  write('Введите высоту башни:');  
  readln(H);  
  V:=Sqrt(2*G*H);  
  writeln('Скорость падения',V:6:3);  
  readln  
end.
```

Задания

«В»: Вывести на экран текст «лесенкой»

Вася

пошел

гулять

«С»: Вывести на экран рисунок из букв

```
Ж
ЖЖЖ
ЖЖЖЖЖ
ЖЖЖЖЖЖЖ
НН НН
ZZZZZ
```

Сложение чисел

Задача. Ввести с клавиатуры два числа и найти их сумму.

Протокол:

Введите два целых числа

компьютер

25 30

пользователь

25+30=55

компьютер считает сам!

?

1. Как ввести числа в память?
2. Где хранить введенные числа?
3. Как вычислить?
4. Как вывести результат?