Лабораторная работа №5

**Тема:** Процедуры​​ и ​​функции.

**Цель:** Изучить условия использования пользовательской функции и процедуры на примере решения задач.

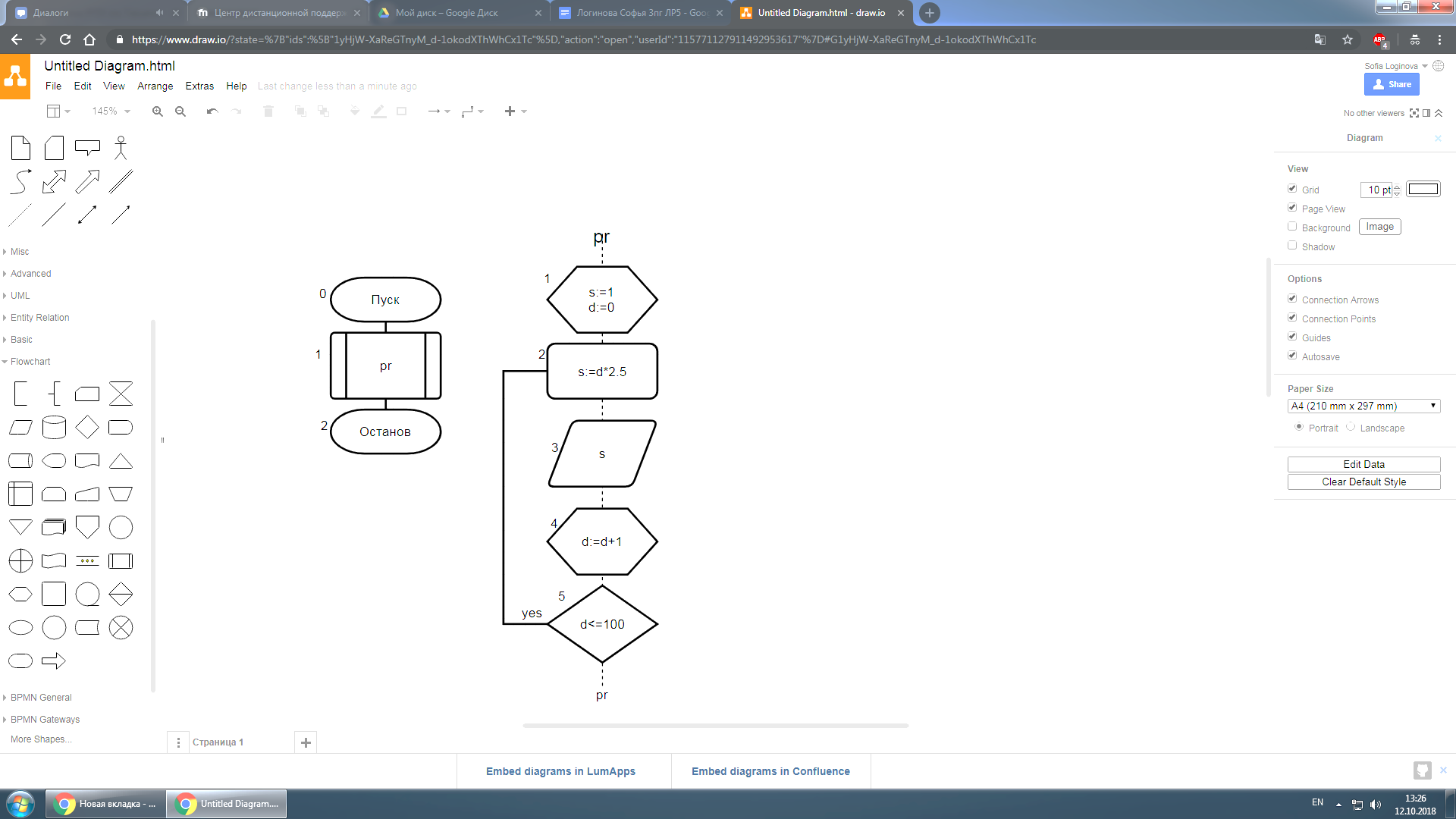
**Оборудование:** ПК, материалы лекций, компилятор Pascal ABC.

Задание №1.

1. **Постановка задачи:**

Перевести дюймы в сантиметры от 0 до 100 дюймов. (1 дюйм=2.5 см). Результаты вывести в виде таблицы. Операторы для формирования вывода таблицы оформить в виде пользовательской

процедуры.

1. **Математическая модель:** s:=d\*2.5;
2. **Блок-схема:**  
   
3. **Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Описание** | **Тип** |
| s | Результат(сантиметры) | real |
| d | Счетчик цикла(дюймы) | integer |

1. **Код программы:**

**program** z1;

**procedure** pr;

**var** s: real;

d: integer;

**begin**

s:=1;

Writeln ('|------|-----------|');

Writeln('|Дюймы | Сантиметры|');

Writeln ('|------|-----------|');

**for** d:=0 **to** 100 **do begin**

s:=d\*2.5;

writeln('|', d:6, '|', s:11:2, '|');

**end**;

Writeln ('|------|-----------|');

**end**;

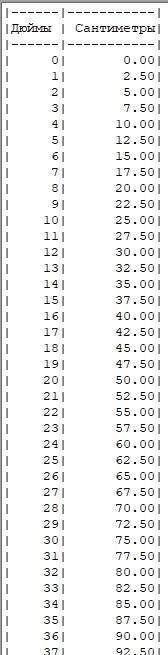
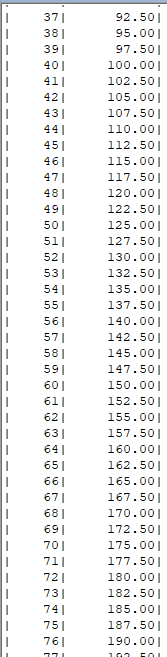
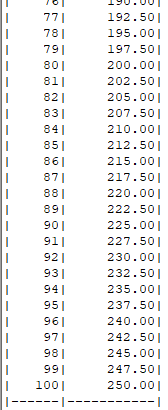
**begin**

pr();

readln();

**end**.

1. **Результат выполнения работы:**

1. **Анализ результатов вычисления:**

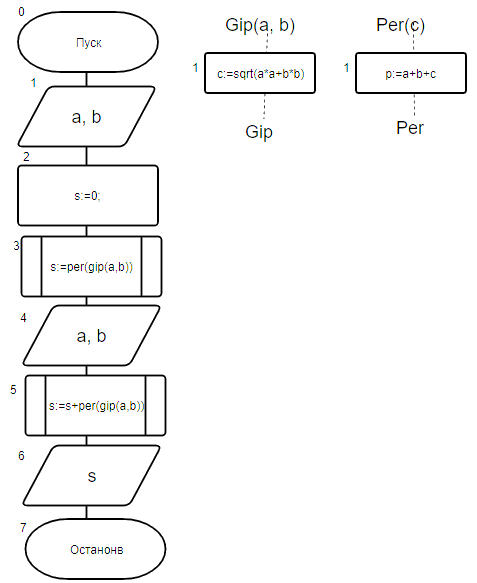
В ходе выполнения задания было изучено введение пользовательской процедуры в основное тело программы, а также принцип ее работы.

Задание №2

1. **Постановка задачи**:

Даны катеты двух прямоугольных треугольника. Найти сумму их периметров, определив две функции: для расчета гипотенузы и периметра прямоугольных треугольников.

1. **Математическая модель:***c2=a2+b2  
   P=a+b+c*
2. **Блок-схема:**



1. **Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Смысл** | **Тип** |
| s | Выходные данные(сумма) | Real |
| a, b | Входные данные(катеты) | Real |
| c | Результат работы процедуры(гипотенуза) | Real |
| p | Результат работы процедуры(периметр) | Real |

1. **Код программы:**

**program** z2;

**var** s, a, b: real;

**Function** Gip(a, b: real): real;

**var** c: real;

**begin**

c:=sqrt(a\*a+b\*b);

Gip:=c;

**end**;

**Function** Per(c: real): real;

**var** p: real;

**begin**

p:=a+b+c;

Per:=p;

**end**;

**begin**

writeln('Vvedite katety 1 treygolnika');

readln(a, b);

s:=0;

s:=per(gip(a,b));

writeln('Vvedite katety 2 treygolnika');

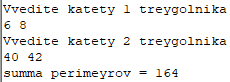
readln(a, b);

s:=s+per(gip(a,b));

writeln('summa perimeyrov = ', s);

**end**.

1. **Результат выполнения работы:**



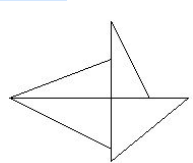
1. **Анализ выполнения вычислений:**

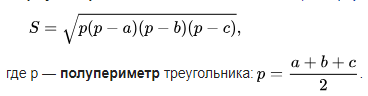
В ходе выполнения задания было изучено введение пользовательской функции в основное тело программы, а также принцип ее работы на примере решения задачи с нахождением гипотенузы прямоугольного треугольника, а также его периметра.

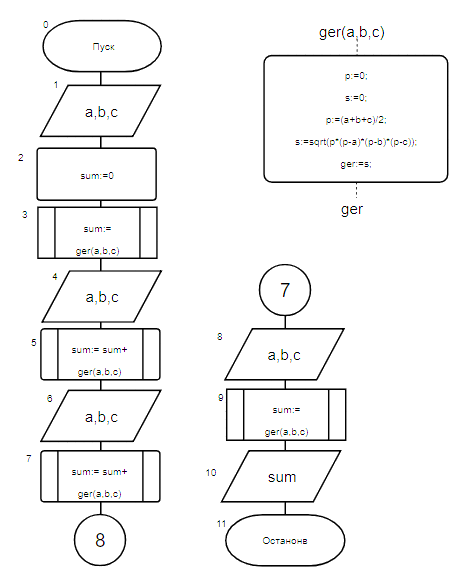
Задание №3

1. **Постановка задачи:**

Вычислить площадь фигуры, заданной сторонами. Фигура не является прямоугольником, а треугольники, которые ее составляют, не являются прямоугольными.



1. **Математическая модель**:
2. **Блок-схема:**



1. **Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Смысл** | **Тип** |
| sum | Выходные данные (сумма площадей) | real |
| a, b, c | Входные данные (стороны треугольников) | real |
| s | Результат работы процедуры (площадь) | real |
| p | Результат работы процедуры (полупериметр) | real |

1. **Код программы:**

**program** z3;

**var** sum, a, b, c: real;

**Function** ger(a, b, c: real): real;

**var** s, p: real;

**begin**

p:=0;

s:=0;

p:=(a+b+c)/2;

s:=sqrt(p\*(p-a)\*(p-b)\*(p-c));

ger:=s;

**end**;

**begin**

writeln('vvedite storoni 1 treygolnika');

readln(a, b, c);

sum:=0;

sum:=ger(a,b,c);

writeln('vvedite storoni 2 treygolnika');

readln(a, b, c);

sum:=sum+ger(a,b,c);

writeln('vvedite storoni 3 treygolnika');

readln(a, b, c);

sum:=sum+ger(a,b,c);

writeln('vvedite storoni 4 treygolnika');

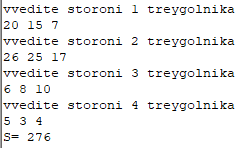
readln(a, b, c);

sum:=sum+ger(a,b,c);

writeln('S= ', sum);

**end**.

1. **Результат выполнения работы:**



1. **Анализ выполненных вычислений:**

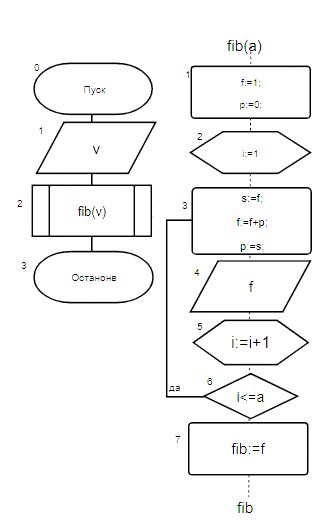
Для выполнения данного задания потребовалась функция, в которой производится расчет площади треугольников по формуле Герона. Подобное введение позволило в теле основной программы избежать одних и тех же вычислений четыре раза (при вводе пользователем данных для нового треугольника).

Задание №4

1. **Постановка задачи:**

С клавиатуры вводится число. Вывести на экран столько элементов ряда Фибоначчи, сколько указал пользователь. Вычисление ряда организовать в функцию. Например, если на ввод поступило число

6, то вывод должен содержать шесть первых чисел ряда Фибоначчи: 1 2 3 5 8 13.

1. **Математическая модель:**
2. **Блок-схема:**
3. **Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Смысл** | **Тип** |
| v | Входные/выходные данные (количество элементов ряда Фибоначчи) | Integer |
| i, a | Счетчики цикла | Integer |
| f | Определяет элемент ряда | Integer |
| s, p | Сохраняют предыдущие два элемента ряда | Integer |

1. **Код программы:**

**program** z4;

**var** v: integer;

**Function** fib(a: integer): integer;

**var** i, s, f, p: integer;

**begin**

f:=1;

p:=0;

**for** i:=1 **to** a **do begin**

s:=f;

f:=f+p;

p:=s;

write(f,' ');

**end**;

fib:=f;

**end**;

**begin**

readln(v);

fib(v);

**end**.

1. **Результат выполнения работы:**



1. **Анализ выполненных вычислений:**

В ходе выполнения задания основной проблемой было сохранение предыдущих двух элементов без использования массива. В результате, в тело цикла было введено две дополнительные переменные, отвечающие за сохранение предыдущих элементов, необходимых для вычисления ряда Фибоначчи. Использование функции позволило в теле основной программы использовать только ввод пользователем количества элементов, необходимых к выводу, и вызов самой функции.

**Вывод:**

В данной лабораторной работе были изучены пользовательские функции и процедуры, которые позволяют организовать повторяющиеся вычисления в единый блок, что существенно ускоряет работу программы и не требует использования большего числа переменных.