

Лабораторная работа № 5
«Разработка алгоритмов решения сложных задач
методом пошаговой детализации и их
программная реализация»
по Разделу
«Разработка алгоритмов решения сложных задач
методом пошаговой детализации и их
программная реализация»

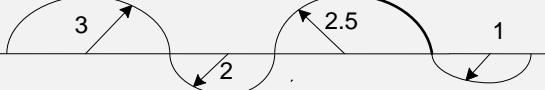
5.1 Общее задание

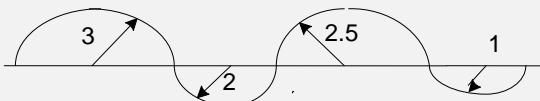
- 1) Изучить основные положения технологий структурного проектирования алгоритмов и метода пошаговой детализации, средства **MS Visio** для графической визуализации алгоритмов.
- 2) Выбрать вариант индивидуального задания из таблицы 5.1.
- 3) Проанализировать постановку задачи своего варианта индивидуального задания и, если необходимо, уточнить ее у преподавателя.
- 4) Провести формализацию решения задачи. Определить этапы ее решения с использованием метода пошаговой детализации.
- 5) Разработать схемы алгоритмов процедур на различных этапах метода пошаговой детализации и схему иерархии процедур.
- 6) Утвердить у преподавателя результаты выполнения п.п. 4-5.
- 7) Разработать программный код функций VC++ по алгоритмам п. 5.
- 8) Создать консольный проект, содержащий три файла исходного кода: файл с главной функцией **main**, файл с функциями ввода и вывода данных и файл с функциями, решающими предписанные задачи. Обмен данными между функциями должен осуществляться через параметры и возвращаемые значения, без использования глобальных переменных. Главная функция **main** должна содержать только операторы вызова разработанных функций.
- 9) Подготовить варианты исходных данных для тестирования проекта.
- 10) Выполнить проект с тестовыми исходными данными и получить результаты. Проверить правильность результатов.
- 11) Оформить отчет по работе в среде **MS Word**. Изобразить все схемы, используя средства **MS Visio**.
- 12) Представить преподавателю отчет по работе.
- 13) Ответить на замечания преподавателя по выполненной работе и на заданные им вопросы по теме.
- 14) Получить отметку о выполнении и защите работы.

5.2 Варианты индивидуальных заданий

Таблица 5.1 – Варианты индивидуальных заданий

№	Задача
1)	Определите периметры правильных n – угольников (10 – угольника, 50 – угольника, 100 – угольника), вписанных в окружность заданного радиуса R. Сторона правильного n-угольника $a = 2 \cdot R \cdot \sin \frac{180^\circ}{n}$; периметр n-угольника $p = n \cdot a$
2)	Определите длины всех медиан треугольника, заданного длинами сторон a,b,c. Медиана, проведенная к стороне a: $m_a = \sqrt{\frac{2b^2 + 2c^2 - a^2}{3}}$.
3)	Определите углы между тремя векторами, направленными из общей начальной точки с координатами (0,0) в конечные точки: точку (2;5); точку (7;6); точку (9;3). Угол между двумя векторами, проведенными из точки (0, 0) в точки (x_1, y_1) и (x_2, y_2) вычисляется по формуле $\alpha = \arccos\left(\frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2}}\right)$
4)	Вычислите площадь пятиугольника, заданного прямоугольными координатами своих вершин: A1(3;2), A2(9;6), A3(14;2), A4(10;-3), A5(7;-2) Использовать формулу площади треугольника: $S = \frac{1}{2} \cdot (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) $
5)	Определите длины всех биссектрис треугольника, заданного длинами сторон a, b, c. Биссектриса угла α вычисляется по формуле: $La = \frac{\sqrt{bc((b+c)^2 - a^2)}}{b+c}$.
6)	Вычислите R – расстояние между двумя точками А и В, заданными сферическими координатами. Соотношение между сферическими координатами и декартовыми $x = r \cdot \sin\theta \cdot \cos\phi;$ $y = r \cdot \sin\theta \cdot \sin\phi;$ $z = r \cdot \cos\theta;$ $0 \leq r < \infty; -\pi < \phi \leq \pi; 0 \leq \theta \leq \pi;$ $R = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$
7)	Вычислите значение площади полной поверхности треугольной пирамиды, если известны длины всех ребер: $ AB = 3; AD = 5; DB = 4; DC = \sqrt{35}; BC = \sqrt{19}; AC = 5.$ Для вычисления площади треугольника использовать формулу Герона: $S = \sqrt{p(p-a)(p-b)(p-c)}; \quad p = \frac{a+b+c}{2}$, где a, b, c - длины сторон треугольника.
8)	Определите высоту, на которой будет мяч, подброшенный вертикально вверх с высоты $y_0=1$ м и начальной скоростью $V_0=20$ м/сек через время $t=1$ сек, 3сек и 4сек. Высота в момент t $y(t) = y_0 + v_0 \cdot t - g \cdot t^2 / 2$, где $g = 9.8$ м/сек

№	Задача
9)	Определите площади правильных n - угольников (10-угольника; 50-угольника; 100-угольника), вписанных в окружность радиуса R. Сторона правильного n-угольника $a = 2R \sin \frac{180}{n}$; площадь n-угольника $S = 0.5 \cdot a \cdot n \cdot r$, где $r = R \cos \frac{180}{n}$ – радиус вписанной окружности.
10)	Определите площадь кольца, внутренний радиус которого равен R1; а внешний R2 ($R2 > R1$). Площадь круга радиуса R вычисляется по формуле: $S = \pi R^2$
11)	Вычислите полярные координаты 3-х точек, заданных прямоугольными координатами в правой полуплоскости. Формулы преобразования координат: $r = \sqrt{(x^2 + y^2)}; \varphi = \arctg\left(\frac{y}{x}\right)$
12)	Определите площадь каждого из 3-х секторов с радиусами R1, R2, R3 и с центральными углами α, β, γ . Площадь S сектора радиуса R с центральным углом δ (в градусах) равна $S = \pi R^2 \frac{\delta}{360}$
13)	Вычислите $y = \operatorname{sh}(x) + \operatorname{tg}(x+1) - \operatorname{tg}^2(2 + \operatorname{sh}(x-1))$ $\operatorname{sh}(x) = \frac{e^x - e^{-x}}{2}$
14)	Определите стороны a, b, c треугольника, заданного величинами своих углов α, β, γ ($\alpha + \beta + \gamma = 180^\circ$) и радиусом описанной окружности R. Применить теорему синусов $\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)} = 2R$
15)	Определите площадь каждого из трех кругов, ограниченных тремя окружностями, длины которых L1, L2, L3 известны. Площадь круга $S = \pi R^2$, длина окружности $L = 2\pi R$.
16)	Определите углы треугольника, длины сторон которого a, b, c заданы. Примените теорему половинного угла: $\operatorname{tg} \frac{\gamma}{2} = \sqrt{\frac{(p-a)(p-b)}{p(p-c)}}, \text{ где } p = \frac{a+b+c}{2}, \gamma - \text{угол, противолежащий стороне } c$
17)	Определите длины всех биссектрис треугольника, заданного длинами сторон a, b, c. Биссектриса угла α вычисляется по формуле: $L_a = \sqrt{bc((b+c)^2 - a^2)}$
18)	Определите общую длину дуги, образованной полуокружностями  Длина половины окружности $L = \pi R$
19)	Вычислите стороны треугольника А и В при условии, что заданы сторона С и углы треугольника α и β , противолежащие сторонам А и В, соответственно. Сторона треугольника вычисляется по формуле $A = C \frac{\sin(\alpha)}{\sin(\gamma)}$; $\gamma = 180^\circ - \alpha - \beta$, где α - угол, противолежащий стороне А, γ - угол, противолежащий стороне С.
20)	Вычислите значения медиан треугольника, сторонами которого являются медианы исходного треугольника со сторонами a, b, c. Длина медианы, проведенной к стороне a: $m_a = 0.5 \sqrt{2b^2 + 2c^2 - a^2}$

№	Задача
21)	Вычислите длину отрезка R, концы которого A и B заданы полярными координатами $(r_A; \varphi_A)$ и $(r_B; \varphi_B)$, по формуле: $R = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2},$ где $(x_A; y_A)$ и $(x_B; y_B)$ – декартовы координаты концов отрезка.
22)	Вычислите $y = ch(x+2) - 3ch(x) + tg 2(1-ch(2x-3))$ $ch(x) = \frac{e^x + e^{-x}}{2}$
23)	Определите длину дуги каждого из трех секторов с радиусами R_1, R_2, R_3 и с центральными углами $\alpha, \beta, \gamma.$ Длина дуги сектора радиуса R с центральным углом α (в градусах) равна $L = 2\pi R \cdot \frac{\alpha}{360}$
24)	Вычислите координаты точки пересечения двух прямых: $a_1x + b_1y = c_1$, вычисляемые по формулам: $x = \frac{\Delta x}{\Delta}; y = \frac{\Delta y}{\Delta}$, где $a_2x + b_2y = c_2,$ $\Delta = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}; \Delta x = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}; \Delta y = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}.$
25)	Определите площади правильных n - угольников (5-угольника 15-угольника; 25-угольника), вписанных в окружность радиуса R. Сторона правильного n-угольника $a = 2R \sin \frac{180}{n}$; площадь n-угольника $S = 0.5 \cdot a \cdot n \cdot r$, где $r = R \cos \frac{180}{n}$ – радиус вписанной окружности.
26)	Определите все углы треугольника при заданных значениях сторон a, b, c. Угол, лежащий против стороны a, вычисляется по теореме косинусов: $\alpha = \arccos(b^2 + c^2 - a^2)/(2 \cdot b \cdot c)$
27)	Вычислите определенный интеграл $y = \int_a^b f(x) dx$ для функции $f(x) = \sqrt{2x + 1}$ по приближенной формуле Симпсона $y \cong \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{6}\right) + f(b) \right]$
28)	Вычислите площади трех кругов S1, S2 и S3 с заданными диаметрами d1, d2 и d3 по формуле $\pi d^2/4$
29)	Вычислите площадь шестиугольника, заданного прямоугольными координатами своих вершин: A1(3;2), A2(9;6), A3(14;6), A4(14;2), A5(10;-3), A6(7;-2) Использовать формулу площади треугольника: $S = \frac{1}{2} \cdot (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) $
30)	Определите общую площадь фигуры, образованной $S = \frac{1}{2} \cdot (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) $ полуокругами  Площадь полуокруга $S = \frac{\pi R^2}{2}.$

5.3 Содержание отчёта

Титульный лист с указанием номера, темы и названия работы, группы и Ф.И.О студента, номера варианта, Ф.И.О преподавателя.

- 1) Общее и индивидуальное задания.
- 2) Формализация и уточнение задания.
- 3) Схемы алгоритмов процедур на различных этапах метода пошаговой детализации и схема иерархии процедур с пояснениями.
- 4) Программный код проекта.
- 5) Результаты выполнения проекта.
- 6) Доказательство правильности результатов выполнения проекта.

5.4 Пример выполнения задания

1) Индивидуальное задание

Вычислить:

$$y = \frac{3.5x^3 - 2.8x^2 + x + 6.7}{x^3 + 0.3x^2 - 1} \quad z = \frac{0.2x^2 - 2x + 0.76}{5.5x^3 + 0.4x^2 + 3x - 2.8}$$

2) Формализация задания

Из условия задачи видно, что выражения в чисителях и знаменателях обоих выражений для вычисления y и z представляют собой в общем случае полиномы (многочлены) 3-й степени:

$$P_3(x) = a_0x^3 + a_1x^2 + a_2x + a_3$$

причем в некоторых случаях отдельные коэффициенты $a_i=0$. Таким образом, для вычисления чиселей и знаменателей формул на нижнем уровне иерархии целесообразно использовать одну общую процедуру, вычисляющую значение многочлена по заданным значениям x и коэффициентов a_i . Для вычисления y и z эта процедура должна будет последовательно вызываться 4 раза.

Поскольку значения коэффициентов зафиксированы в условии задачи, будем считать, что единственным вводимым исходным данным задачи является значение переменной x , а выводимыми результатами ее решения – значения y и z .

3) Разработка алгоритмов решения задач

Начнем проектирование алгоритма методом «сверху вниз».

3.1) На самом верхнем (первом) уровне алгоритм решения задачи можно укрупненно представить в виде вызова главной процедуры с именем `main`

(рисунок 5.1), где **CalcYZ** (процедура) вычисления значений переменных **y** и **z** по заданному значению переменной **x**.

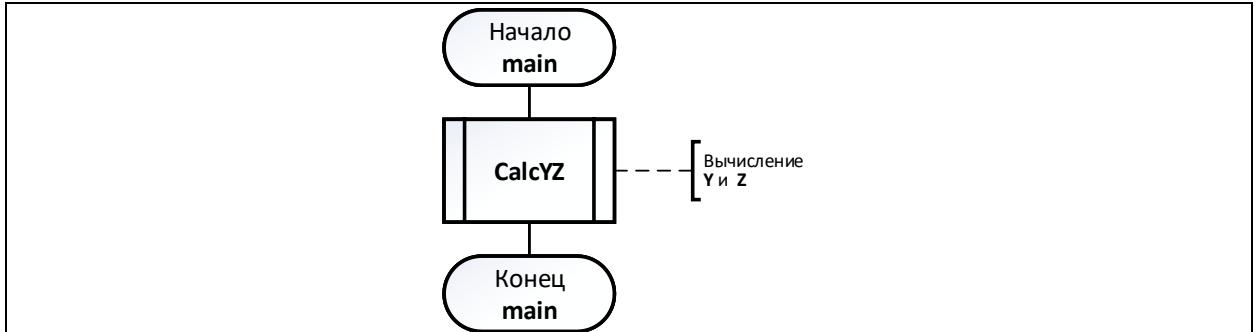


Рисунок 5.1 – Укрупненная схема алгоритма **main** решения задачи

3.2) На следующем, втором уровне, детализируем алгоритм процедуры **CalcYZ** путем представления его в виде последовательности следующих трех процедур (рисунок 5.2):

- процедуры ввода значения **x** с именем **GetX**;
- процедуры вычисления **y** и **z** с именем **CompYZ**;
- процедуры вывода вычисленных значений периметра и площади с именем **PutYZ**.

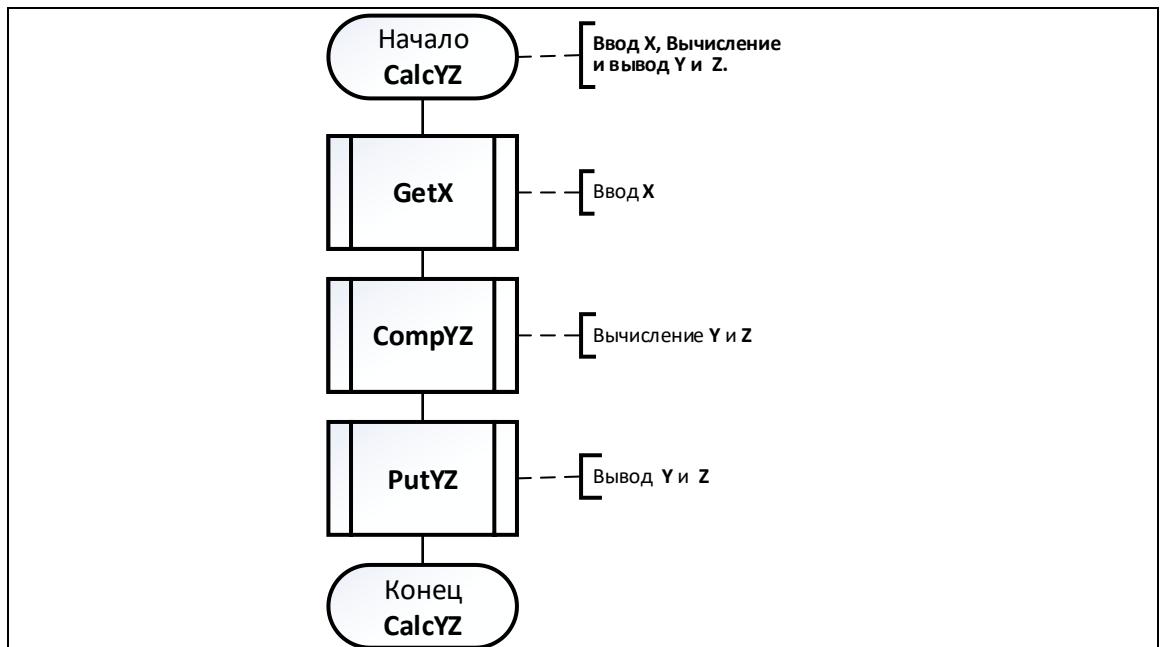


Рисунок 5.2 – Результат второго уровня детализации алгоритма

3.3) Перейдем к следующему, третьему уровню детализации. Процедуры **GetX** и **PutYZ** дальнейшей детализации не требуют, так как средства ввода-вывода имеются в любом языке программирования. Поэтому на следующем, третьем шаге детализируем алгоритм процедуры вычисления **y** и **z** **CompYZ** (рисунок 5.3).

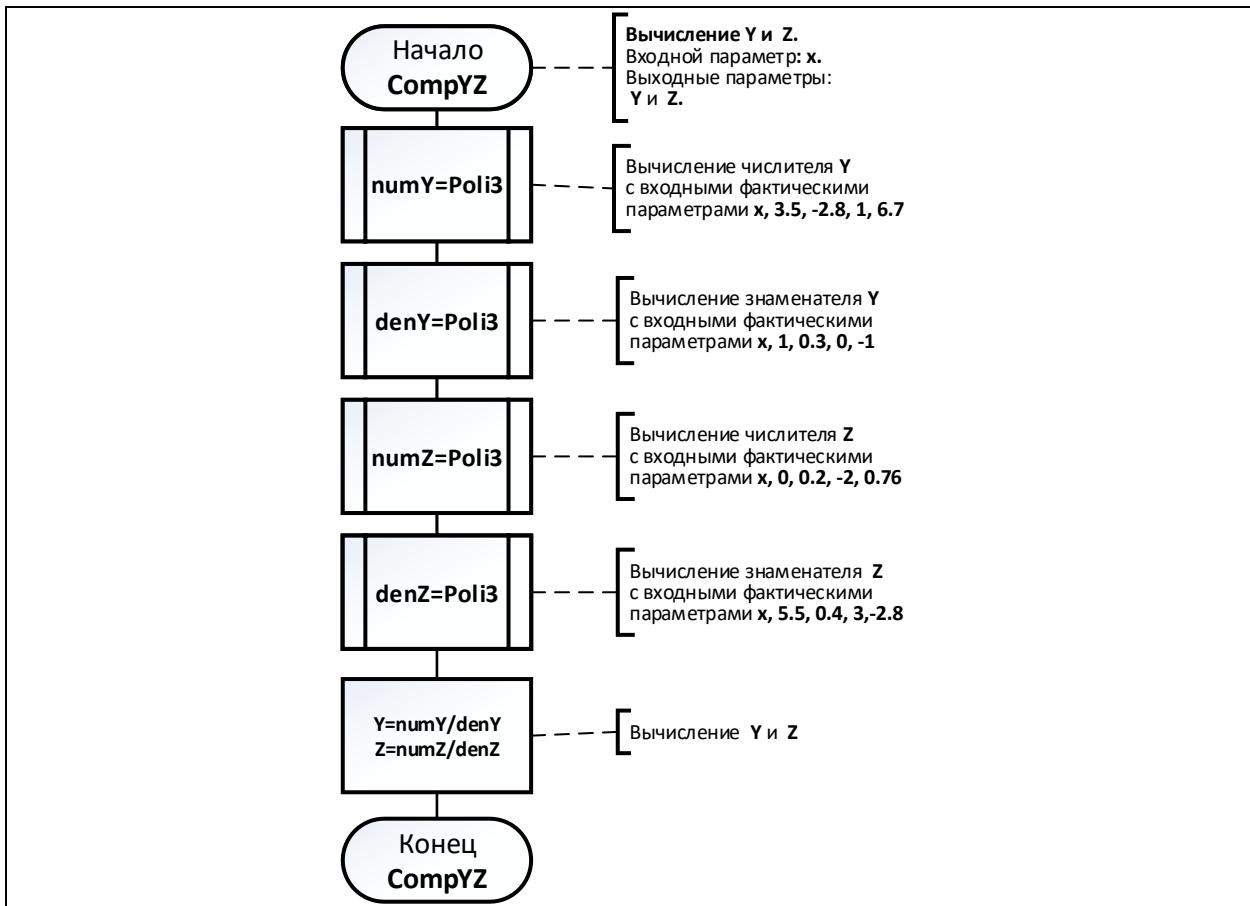


Рисунок 5.3 – Схема алгоритма процедуры **CompYZ**

Из расчетных формул видно, что для вычисления **y** и **z** необходимо вычислить значения двух числителей и двух знаменателей заданных выражений, которые представляют собой полиномы 3-й степени. Поэтому на этом уровне детализации воспользуемся одной и той же процедурой **Poli3**, которая будет вызываться из процедуры **CompYZ** 4 раза с различными параметрами.

3.4) На последнем, четвертом уровне детализации разработаем схему алгоритма процедуры **Poli3**, вычисляющей значение полинома 3-й степени по заданным значениям коэффициентов полинома и переменной **x** (рисунок 5.4).

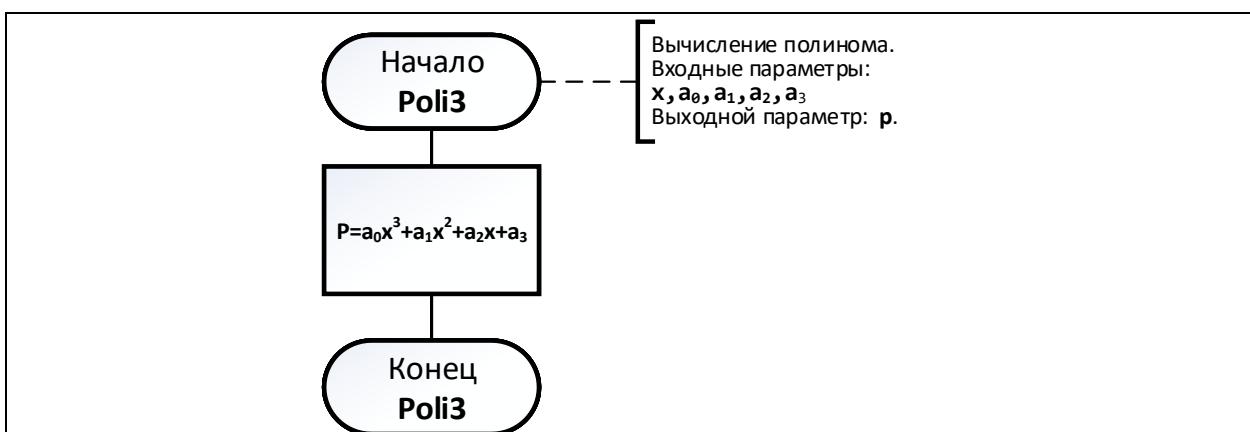


Рисунок 5.4 – Схема алгоритма процедуры **Poli3**

Схема иерархии процедур для решения задачи изображена на рисунке 5.5.

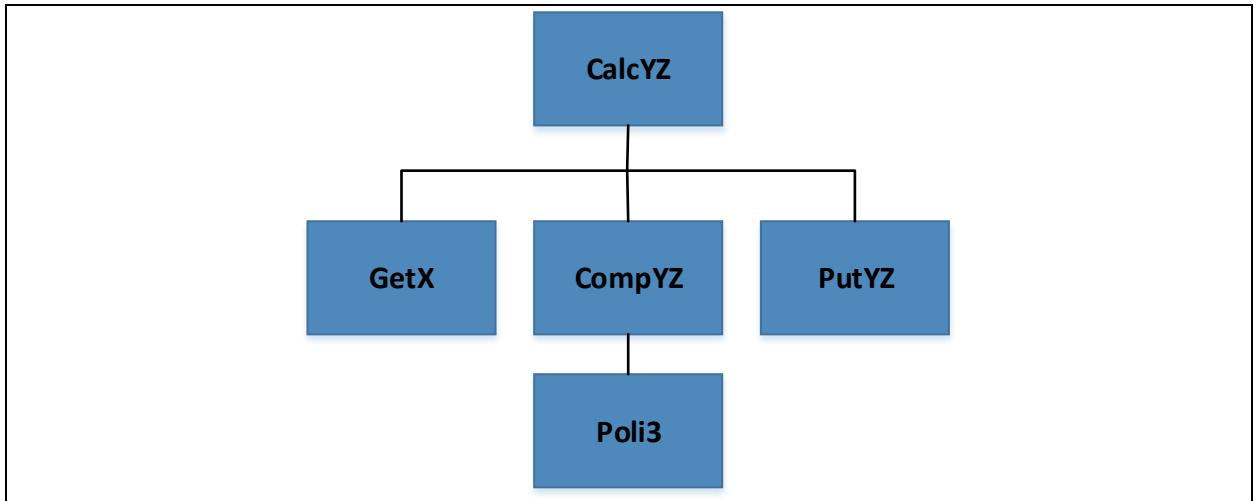


Рисунок 5.5 – Схема иерархии процедур для решения задачи

4) Разработка программного проекта

При разработке программного проекта внесем незначительное упрощение: избавимся от лишней “двухэтажности” и реализуем процедуру **CalcYZ** в главной функции проекта **main**.

Для решения нашей задачи создадим проект, в котором будет три файла исходного кода: помимо файла с главной функцией **main** создадим файл с функциями **GetX** и **PutYZ** для ввода и вывода и объединим в отдельный файл все остальные функции, участвующие в решении задачи.

Программный код проекта представлен ниже на рисунках 5.6 – 5.8.

В первый файл с именем **GetPut.cpp** (рисунок 5.6) запишем функции ввода исходных данных **GetX** и вывода результатов **PutYZ**. Обе эти функции типа **void**, так как не имеют возвращаемого значения. Функция ввода значения **x** **GetX** имеет один выходной параметр и ни одного входного параметра. Функция **PutYZ**, предназначенная для вывода **y** и **z**, имеет 2 входных параметра и ни одного выходного параметра. Так как функции этого файла используют объекты **cin** и **cout**, то в нем присутствуют директивы **#include <iostream>** и **using namespace std**.

Во второй файл с именем **Calc.cpp** (рисунок 5.7) запишем функции, выполняющие необходимые вычисления для решения задачи. Функция **CompYZ** определена как функция с одним входным параметром **x** и двумя выходными параметрами **y** и **z**. Возвращаемого значения функция не имеет.

Функция **Poli3**, вычисляющая значение многочлена 3-й степени, имеет пять входных параметров (значение **x** и коэффициенты многочлена **a_i**) и возвращаемое значение – вычисленную величину многочлена.

```

// Файл GetPut.cpp

#include <iostream>
using namespace std;

// Определение функции ввода X
void GetX(double& x)
{
    setlocale(LC_ALL, "rus");
    cout<<"Введите значение X: ";
    cin>>x;
}

// Определение функции вывода Y и Z
void PutYZ(double y, double z)
{
    setlocale(LC_ALL, "rus");
    cout<<"Значение Y= "<<y<<endl;
    cout<<"Значение Z= "<<z<<endl;
}

```

Рисунок 5.6 – Программный код файла с функциями **GetX** и **PutYZ**

```

// Файл Calc.cpp

// Прототип функции Poli3
double Poli3(double, double, double, double, double);

// Определение функции CompYZ для вычисления Y и Z
void CompYZ(double x, double& y, double& z)
{
    double numY = Poli3(x, 3.5, -2.8, 1., 6.7);
    double denY = Poli3(x, 1., 0.3, 0., -1.);
    double numZ = Poli3(x, 0., 0.2, -2., 0.76);
    double denZ = Poli3(x, 5.5, 0.4, 3., -2.8);
    y = numY/denY;
    z = numZ/denZ;
}

// Определение функции Poli3 для вычисления многочлена 3-й степени
double Poli3(double x, double a0, double a1, double a2, double a3)
{
    double p = a0*x*x*x + a1*x*x + a2*x +a3;
    return p;
}

```

Рисунок 5.7 – Программный код файла с функциями **CompYZ** и **Poli3**

Так как функция **CompYZ** определена перед вызываемой ею функцией **Poli3**, то в начале файла записан прототип функции **Poli3**.

В третьем файле с именем **main.cpp** (рисунок 5.8) находится главная функция **main**, программный код которой содержит только вызовы

разработанных функций ввода исходных данных, решения задачи и вывода результатов. В этом файле присутствует директива препроцессора `#include <iostream>`, необходимая для команды задержки закрытия консольного окна `system("PAUSE")`. Перед определением главной функции `main` в файл включены прототипы вызываемых в ней функций `GetX`, `CompYZ` и `PutYZ`.

```
// Файл main.cpp

#include <iostream>

// Прототипы вызываемых функции
void GetX(double&);
void PutYZ(double, double);
void CompYZ(double, double&, double&);

// Определение главной функции main
void main()
{
    double x, y, z;
    GetX(x);
    CompYZ(x, y, z);
    PutYZ(y,z);
    system("PAUSE");
}
```

Рисунок 5.8 – Программный код файла с главной функцией `main`

5) Исходные данные для тестирования проекта

Протестируем проект при двух значениях входной переменной `x`: `x=0` и `x=1`.

Расчет на калькуляторе значений `y` и `z` при `x=0` дает следующие результаты: `y=-6.7`, `z=-0.271429`.

Расчет на калькуляторе значений `y` и `z` при `x=1` дает следующие результаты: `y=28`, `z=-0.170492`.

6) Результаты выполнения проекта

Результаты выполнения проекта при `x=0` приведены на рисунке 4.9, при `x=1` – на рисунке 4.10.

```
Введите значение X: 0
Значение Y= -6.7
Значение Z= -0.271429
Для продолжения нажмите любую клавишу . . .
```

Рисунок 5.9 – Результаты выполнения проекта при `x=0`

```
Введите значение X: 1
Значение Y= 28
Значение Z= -0.170492
Для продолжения нажмите любую клавишу . . .
```

Рисунок 5.10 – Результаты выполнения проекта при $x=1$

7) Доказательство правильности результатов выполнения проекта

Правильность результатов выполнения проекта подтверждается их совпадением с результатами контрольных вычислений на калькуляторе для обоих вариантов исходных данных.

Приложение 5.1 – Создание схем алгоритмов с использованием графического редактора MS Visio

1) Назначение MS Visio

Основным назначением графического редактора *MS Visio* является визуализация данных: создание схем алгоритмов, иллюстраций, диаграмм и др. Далее будут рассмотрены в основном только те возможности *Visio*, которые необходимы для создания схем алгоритмов задач.

Процесс оформления схем алгоритмов достаточно прост. В *Visio* имеется большое количество готовых элементов, называемых фигурами или в соответствии с ГОСТом (ГОСТ 19.701-90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения) – блоками. Каждый из блоков имеет свое назначение. С помощью мыши эти элементы можно переносить в нужные места на рабочий лист. Затем выбранные блоки соединяются линиями, указывающими связи между ними (процесс выполнения алгоритма), и снабжаются поясняющими надписями. Готовую схему можно использовать как самостоятельный документ или поместить в качестве рисунка (иллюстраций) в документ, подготовленный в другом приложении.

Так как *Visio* является составной частью *MS Office*, то большинство кнопок, команд и инструментов имеют аналогичный смысл, а назначение кнопок и элементов меню, являющимися специфическими для *Visio*, комментируются всплывающими пояснениями.

К сожалению, *Visio* не обладает функциональными средствами редактора формул, поэтому при необходимости включения формул в блоки формируемой схемы алгоритма рекомендуется использовать встроенный редактор формул *MS Word* или какой-либо другой.

В данном кратком описании используется версия *Visio 2016*.

2) Создание и открытие документа

После запуска *Visio* на экране появится окно, представленное на рисунке П5.1. В этом окне приведены пиктограммы основных категорий шаблонов документов, с которыми способно работать *Visio*. Выбор категории шаблона осуществляется щелчком по нужному шаблону, после чего нужно уточнить свой выбор, щелкнув по одному из категорий шаблонов.

Поскольку нам необходимо разрабатывать схемы алгоритмов, необходимо выбрать категорию **Простая блок-схема**. После щелчка по пиктограмме **Простая блок-схема** появится окно, в котором необходимо нажать кнопку **Создать** (рисунок П5.2). В результате появится основное окно *Visio* (рисунок П5.3).

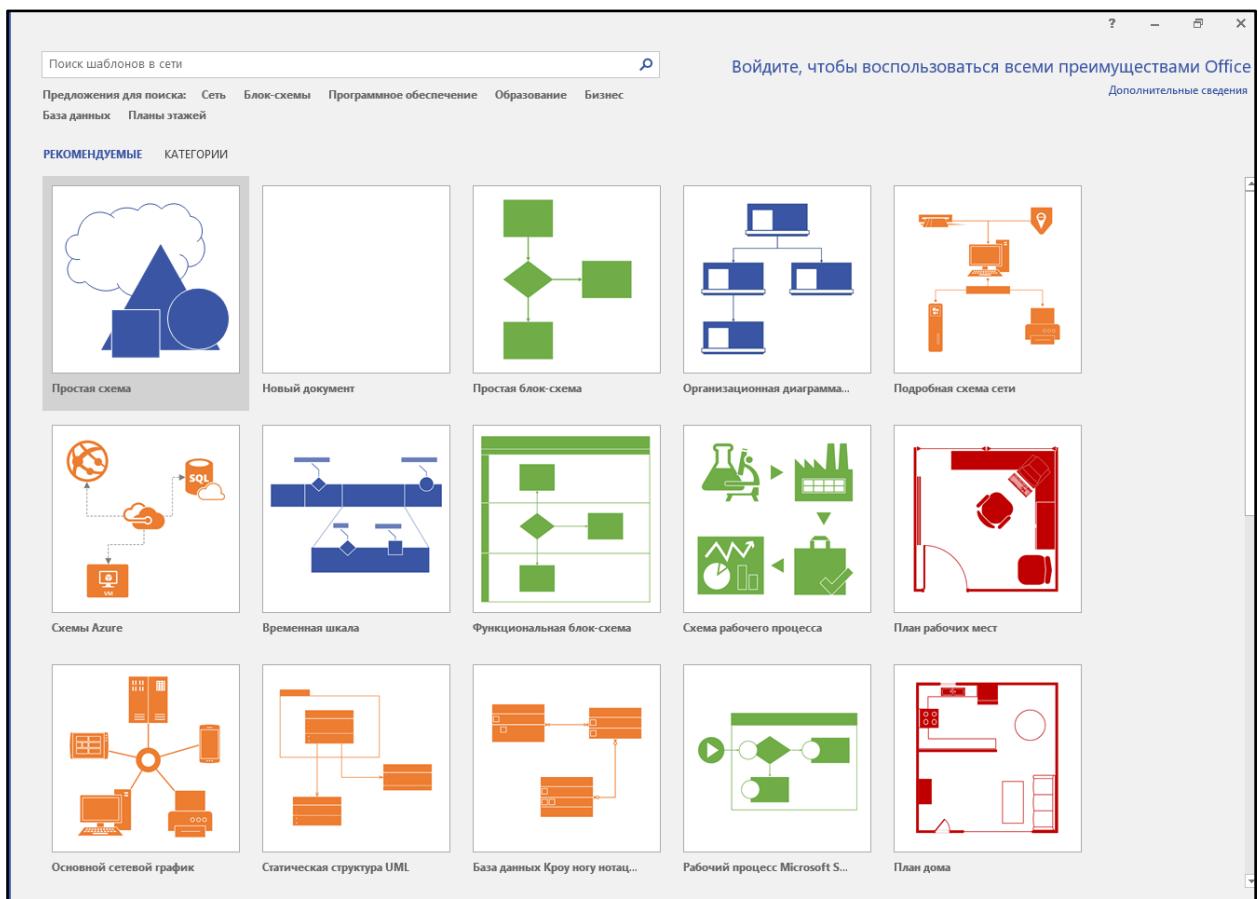


Рисунок П5.1 – Окно *MS Visio* сразу после запуска

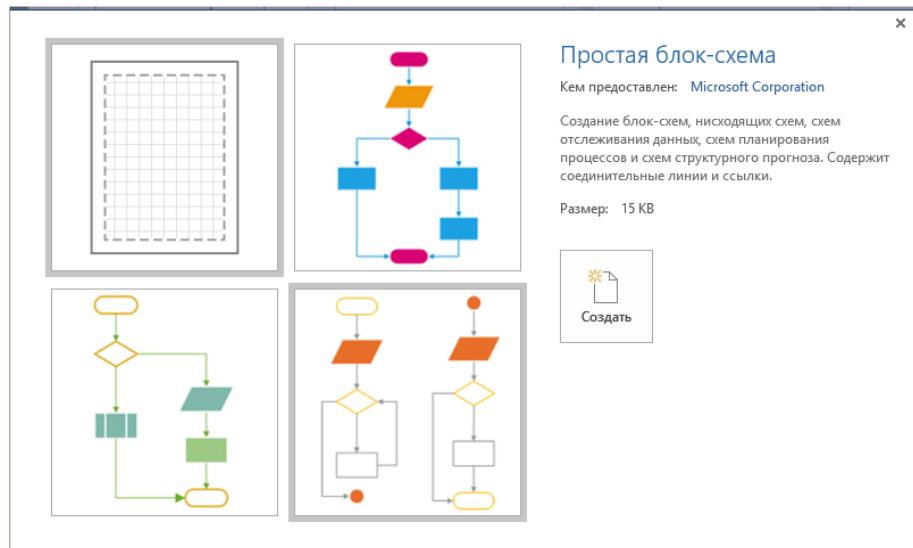


Рисунок П5.2 – Окно шаблона *Простая блок-схема*

После выбора категории шаблонов *Visio* подготовит к использованию тот или иной набор **базовых элементов (фигур, блоков)** для сборки схемы, чертежа или рисунка.

В данном кратком описании *Visio* будут рассмотрены только средства и приемы создания **Простых блок-схем** (схем алгоритмов), поэтому для создания нового документа этого типа нужно открыть в списке **Фигуры**:

наборы элементов Фигуры простой блок-схемы, дважды щелкнув по этой строке (рисунок П5.3).

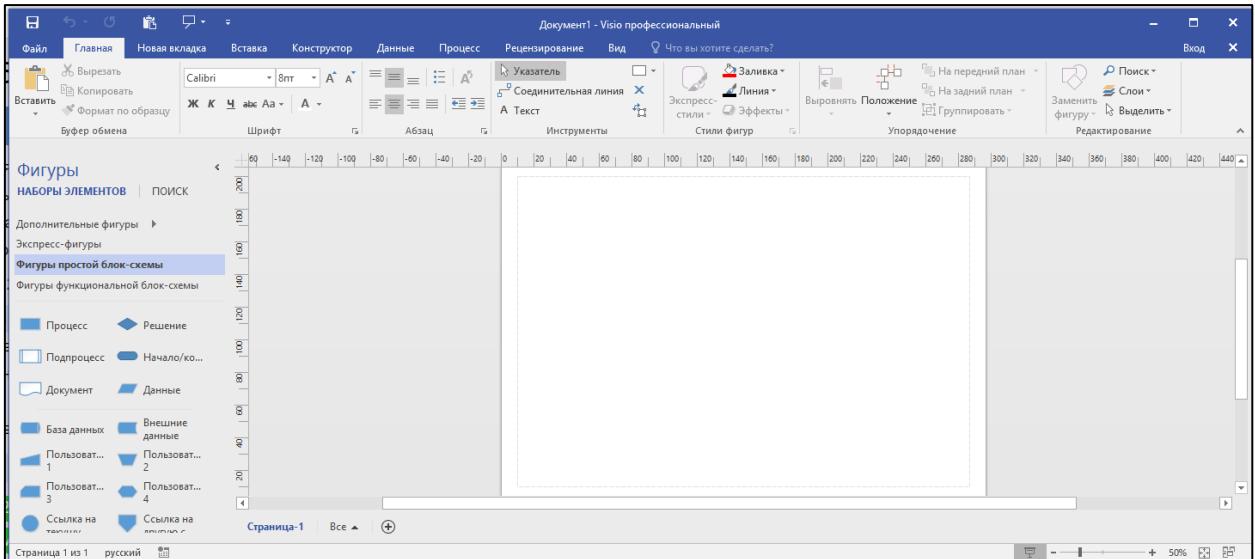


Рисунок П5.3 – Окно *Visio* после создания нового документа

Так как в стандартном наборе элементов могут отсутствовать необходимые фигуры, то можно создать свой список необходимых фигур, сформированный из различных наборов элементов.

Для создания новых страниц можно воспользоваться кнопкой +, находящейся под рабочим полем документа.

Действия, связанные с *Сохранением документа*, *Открытием файла с документом*, *Закрытием документа* и *Завершением работы*, *Отменой* и *повтором отмененных действий* и др., аналогичны соответствующим действиям в *MS Word* соответствующей версии, поэтому их рассматривать не будем.

3) Создание простых схем

Начнем работу над созданием схемы алгоритма с запуска *Visio* и выбора категории создаваемого документа – **Простая блок-схема**. В результате на экране появится окно документа (рисунок П5.3), в центральной части которого будет располагаться пустой лист документа, на котором предстоит разместить блоки схемы алгоритма.

В верхней части окна располагаются панели инструментов, а слева можно видеть панель **Фигуры** (изображение базовых элементов). Именно из них мы будем собирать схему алгоритма.

Блоки и элементы схем разбиты на тематические группы. При выборе категории шаблона *Visio* подбирает подходящие для данной задачи группы элементов автоматически. На рисунке П5.3 показана группа **Фигуры простой блок-схемы**.

Масштаб изображения. Изображение в окне *Visio* может быть слишком мелким или слишком крупным. Нужную величину изображения можно отрегулировать стандартными средствами *MS Office*.

Быстро увеличить или уменьшить изображение можно при помощи колеса мыши – вращая его при нажатой клавише *<Ctrl>*.

Направляющие линии и сетка. В *Visio* предусмотрены функциональные средства для размещения элементов на листе.

Прежде всего, это **направляющая сетка**, которая отображается в окне документа по умолчанию. При перемещении или изменении размеров элементов, их углы или стороны «прилипают» к сетке, что позволяет быстро придавать элементам одинаковые размеры и пропорции. По умолчанию линии сетки проводятся через 5 мм.

Кроме сетки, фигуры рисунков могут **автоматически выравниваться** относительно делений линеек и соседних элементов.

Скрыть сетку или вернуть ее на экран можно воспользоваться командой **Сетка** вкладки **Вид** панели инструментов (рисунок П5.4).

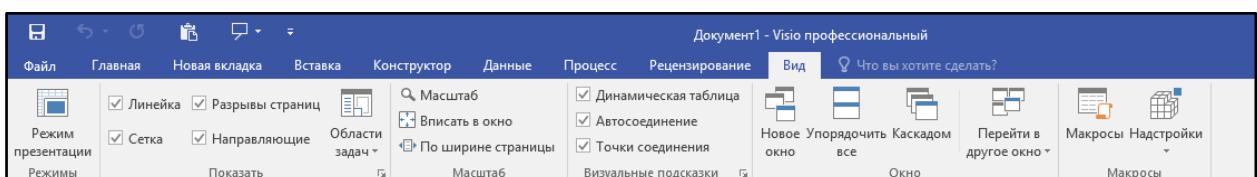


Рисунок П5.4 – Инструменты вкладки **Вид**

Еще одним средством точного позиционирования элементов являются **направляющие линии** (рисунок П5.4), которые могут быть горизонтальными или вертикальными. К ним, как и к линиям сетки, «прилипают» элементы изображения.

К направляющим «прилипают» не только углы и стороны, но и центры элементов, что позволяет быстро и ровно установить несколько блоков по вертикали или по горизонтали.

Для того чтобы создать новую направляющую линию, установите указатель мыши над **линейкой**, расположенной вдоль верхнего или левого края листа. Нажмите левую кнопку мыши и, не отпуская ее, переместите указатель. На листе появится направляющая линия, которую вы сможете поместить в интересующей вас части листа. При необходимости линию можно легко переместить, также с помощью мыши. Выделив направляющую щелчком мыши и нажав клавишу **Delete**, вы сможете удалить линию, которая больше не нужна. Лишние направляющие могут мешать точной расстановке элементов.

Сборка схемы алгоритма из блоков. При помощи мыши необходимо «захватить» нужную фигуру и перенести ее в поле документа. Для этого необходимо установить курсор на интересующую фигуру, нажать левую кнопку мыши и, не отпуская ее, переместить курсор на лист, расположенный в центральной части окна. Можно сразу поместить фигуру в ту часть листа, где он должен находиться. При необходимости можно переместить его

позднее. Подобным образом можно по мере необходимости добавлять к схеме новые блоки любых доступных категорий.

Сразу после того, как элемент помещен на лист рисунка, можно ввести в него текстовую подпись или название. Для этого дважды щелкните на элементе левой кнопкой мыши – *Visio* перейдет в режим редактирования текста, расположенного внутри блока.

Создание связей между фигурами. Неотъемлемой частью схем алгоритмов являются обозначения связей между отдельными элементами. Следует соединить блоки между собой, добавляя по мере необходимости стрелки и, возможно, используя линии разных типов. В *Visio* это действие практически полностью автоматизировано.

Для того чтобы соединить между собой два блока, нужно на панели инструментов выбрать инструмент **Соединительная линия**. Если обратить внимание на изображения блоков, то на их сторонах можно заметить точки привязки. Эти точки служат для «подключения» к блокам соединительных линий. Если подвести указатель мыши к точке привязки, то выбранная точка будет выделена цветным прямоугольником (рисунок П5.5). Теперь нажмите левую кнопку мыши и, не отпуская ее, переместите указатель к точке привязки элемента, с которым вы хотите установить связь. Пунктирная линия отобразит предполагаемую форму соединительной линии. При необходимости позже можно изменить ее траекторию. Убедившись в том, что линия соединяет нужные вам точки, отпустите кнопку мыши. Между блоками появится соединительная линия, которая по умолчанию имеет стрелку на конце.

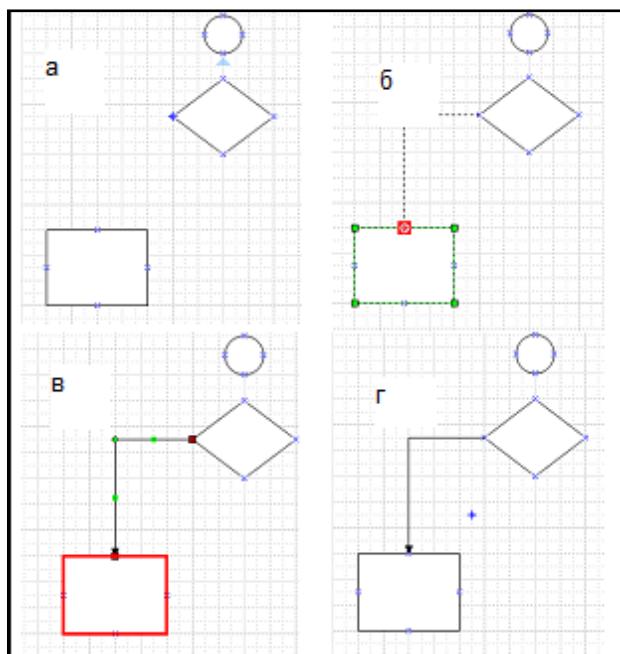


Рисунок П5.5 – Процесс создания соединительных линий:
а – выбор начальной точки линии; б – выбор конечной точки;
в – соединительная линия с маркерами; г – готовый соединитель

Обратите внимание на квадратные маркеры, наложенные на соединительную линию. Перемещая при помощи мыши маркеры,

расположенные на концах соединителя, можно соединить между собой два других блока или перенести конец соединительной линии на другую точку привязки.

После того, как работа с соединительной линией будет закончена, снимите с нее выделение. Для этого щелкните мышью по свободному полю листа.

4) Настройка внешнего вида блоков схемы алгоритма

После того, как основные блоки документа помещены на лист и связи между ними установлены, можно приступить к настройке внешнего вида элементов. Можно изменять цветной фон элементов и параметры их границ, изменить способ начертания соединительных линий или задать форму стрелок. Но перед тем, как приступить к настройке того или иного объекта, необходимо указать, что будут работать именно с ним. Для этого интересующий вас объект следует выделить.

Выделение и группировка элементов. Для выделения объектов предназначен инструмент **Указатель**. Для этого надо нажать кнопку с символом в виде стрелки курсора, расположенную на панели инструментов. После этого подвести указатель мыши к блоку, с которым будут выполнять дальнейшие действия, и щелкнуть на нем левой кнопкой мыши. Вокруг выделенного элемента появляется рамка с маркерами, позволяющими изменять его размер и угол наклона (рисунок П5.6).

Достигнув нужной позиции, кнопку мыши следует отпустить. Если нужно переместить сразу несколько элементов, то надо выделить сразу несколько блоков.

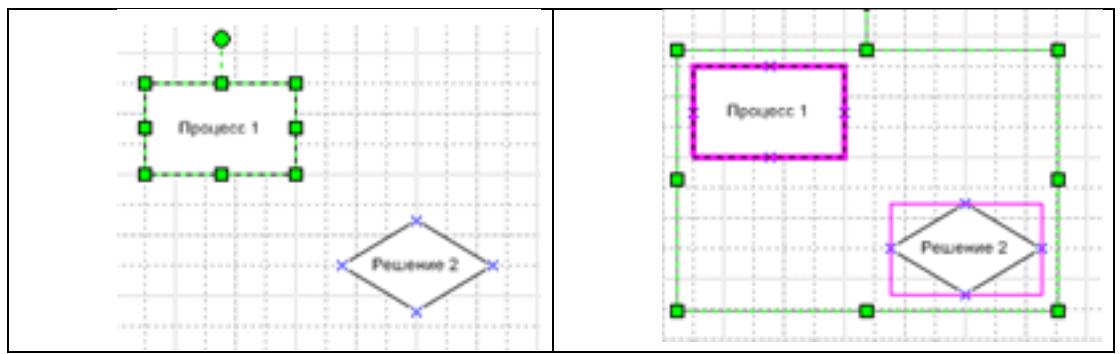


Рисунок П5.6 – Выделение одного или нескольких элементов:
а – вокруг выделенного элемента отображаются маркеры;
в – можно выделить сразу несколько элементов.

Существует несколько способов ускорения процесса выделения:

- Команда **Выделить все** функции **Выделить** категории **Редактирование** инструментальной панели, а также связанная с

ней комбинация клавиш **<Ctrl+A>** позволяют быстро выделить все элементы документа;

- используя инструмент **Указатель**, начертите на листе рамку – для этого перемещайте указатель мыши, удерживая нажатой ее левую кнопку. После того, как вы отпустите кнопку мыши, будут выделены все элементы, полностью оказавшиеся внутри рамки. Если в итоге выделенными оказались несколько «лишних» блоков, снимите с них выделение. Для этого щелкните на них левой кнопкой мыши, удерживая нажатой клавишу **<Ctrl>**;
- можно переключить инструмент **Указатель** в режим выбора нескольких элементов. Для этого щелкните левой кнопкой мыши на небольшой стрелке, расположенной справа от кнопки этого инструмента. Из появившегося на экране меню выберите команду **Выбор нескольких объектов** (рисунок П5.7). После этого изображение на кнопке инструмента дополнится значком +, а каждый щелчок левой кнопки мыши на элементе будет выделять его или снимать выделение.

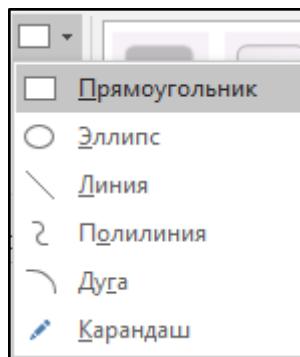


Рисунок П5.7 – Включение режима выделения нескольких объектов

Снятие выделения с элементов. Если вы хотите снять выделение со всех выделенных объектов, просто щелкните левой кнопкой мыши на пустом поле схемы. Если снять выделение нужно только с некоторых объектов, щелкните на них левой кнопкой мыши, удерживая при этом нажатой клавишу **<Shift>**. Эта же клавиша позволит добавить к выделению новые блоки.

Копирование объектов. Так как схема алгоритма может содержать множество однотипных элементов, то подготовив один образец и скопировав его нужное число раз, можно существенно сократить временные затраты.

Для того, чтобы скопировать один или несколько элементов, надо их выделить, затем переместить при помощи мыши, удерживая нажатой клавишу **<Ctrl>** – вместо обычного перемещения они будут скопированы в выбранную вами область листа. При нажатой клавише **<Ctrl>** указатель мыши будет дополнен символом + – это говорит о том, что режим копирования включен.

С помощью буфера обмена можно поместить в документ **Visio** и элементы, помещенные в буфер обмена при помощи других программ. Вкладка панели инструментов **Вставка** главного меню позволяет не только вставить

фрагмент из буфера, но и выбрать формат, в котором этот фрагмент будет использован (рисунок П5.8).

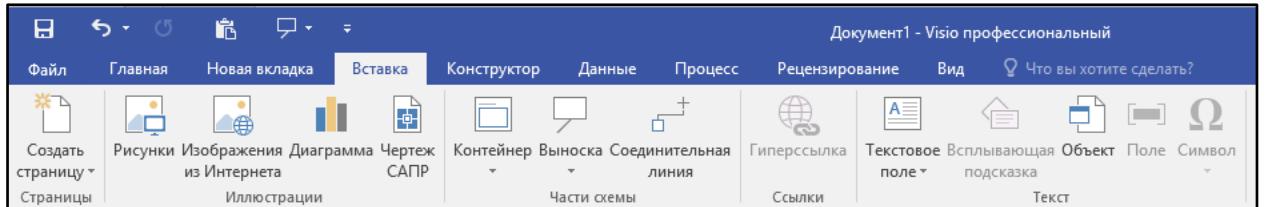


Рисунок П5.8 – Команды вкладки Вставка

Удаление блоков. Чтобы удалить один или несколько блоков из документа, их просто надо выделить и нажать клавишу *Delete*.

Если необходимо не просто удалить, а переместить объект в другой документ или приложение, можно воспользоваться комбинацией клавиш *<Ctrl+X>* или командой *Вырезать* элемента *Правка* главного меню или контекстного меню объекта. Эти команды и комбинация клавиш позволяют удалить выделенные объекты из документа, сохранив их в буфере обмена.

Изменение размера, перемещение. Как уже говорилось, при выделении одного или нескольких элементов вокруг них появляется рамка с маркерами. Перемещая эти маркеры мышью, можно изменять размеры и пропорции выбранных блоков (рисунок П5.9).

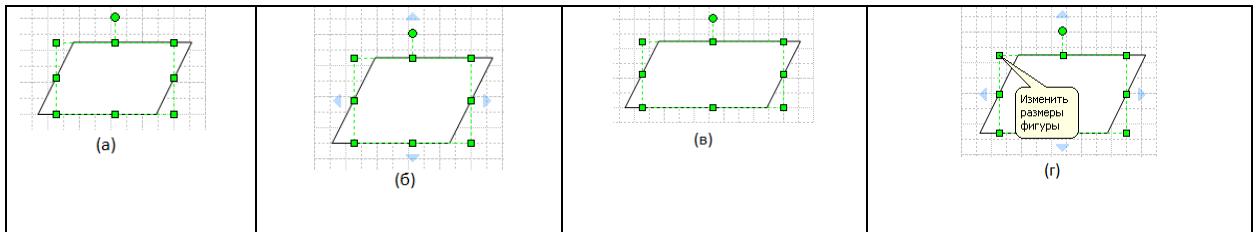


Рисунок П5.9 – Маркеры рамки выделения позволяют изменять размеры объекта:

- а – объект в исходном состоянии;
- б – изменение размера по вертикали;
- в – изменение размера по горизонтали;
- г – пропорциональное изменение размера элемента

Подведя к маркеру указатель мыши, можно увидеть, что он (указатель мыши) принял вид двунаправленной стрелки. Это говорит о том, что программа готова к настройке объекта.

Маркеры, расположенные на середине сторон рамки выделения, позволяют изменять размер элемента по вертикали или горизонтали (рисунок П2.9б, в), а использование угловых маркеров позволяет изменить размер фигуры пропорционально в обоих направлениях (рисунок П5.9г).

Пропорции фигур изменять можно только в крайних случаях, так пропорции основных фигур определены в ГОСТе.

Настройка толщины, типы штриха и окончаний линий. Настроить внешний вид образующих документ линий можно с использованием команд инструмента **Линия** (рисунок П5.10).

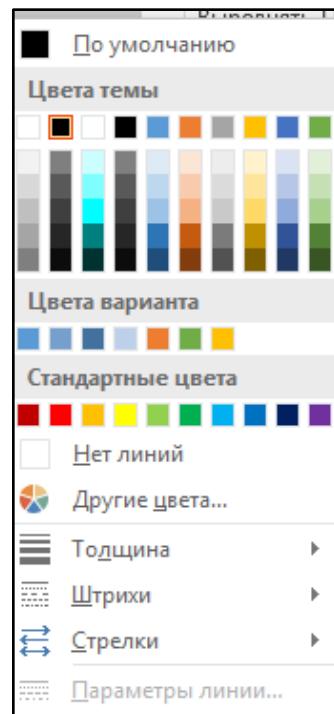


Рисунок П5.10 – Команды инструмента **Линия**

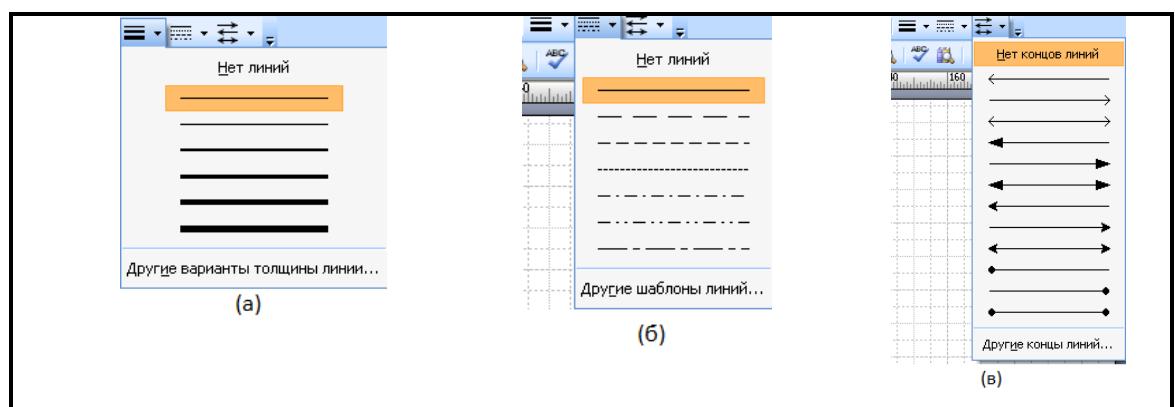


Рисунок П5.11 Элементы настройки внешнего вида границ элементов и соединительных линий

При помощи команды **Толщина** можно выбрать толщину границы блока или соединительной линии. При простом нажатии на эту команду для выбранного элемента устанавливается толщина линии, используемая по умолчанию, или выбранная при настройке предыдущего объекта. При нажатии на небольшую стрелку, расположенную справа от кнопки, на экран выводится меню, позволяющее выбрать другое значение толщины (рисунок П5.11а).

Следующая команда, имеющая название **Штрихи**, позволяет выбрать штриховку линии, например, сплошную или пунктирную. Небольшая стрелка,

расположенная справа от этой кнопки, позволяет вызвать на экран меню выбора типа штриха (рисунок П5.11б).

Третья кнопка, называющаяся *Стрелки*, позволяет снабдить концы соединительной линии стрелками. Нажав на нее, можно, по аналогии с другими кнопками настройки, применить к объекту параметры линии, выбранные при изменении предыдущего блока. Небольшая стрелка, расположенная справа от кнопки, позволяет вызвать на экран меню выбора оформления концов линии, которое представлено на рисунок П5.11в.

Настроенный формат можно перенести на другие элементы документа. Для этого надо выделить объект с копируемым форматом и нажать кнопку *Формат по образцу*, находящуюся на панели инструментов или воспользоваться комбинацией клавиш *<Ctrl+Shift+P>*. Указатель мыши будет дополнен значком в виде кисти. После этого достаточно щелкнуть левой кнопкой мыши по элементу, которому необходимо придать такой же формат, что и исходный объект.

5) Работа с текстом

Практически все документы содержат текстовые элементы. Все текстовые элементы, используемые в *Visio*, можно разделить на три группы:

- **текстовые подписи к элементам** – вводятся в процессе создания или редактирования отдельных блоков и являются их составляющими частями;
- **самостоятельные надписи** (текстовые блоки) – существуют в виде отдельных элементов и могут быть перемещены или настроены независимо от остальных частей изображения;
- **выносные элементы и комментарии** – объекты этого типа специально предназначены для размещения подписей и «выносок», поясняющих значение других элементов.

Текстовые подписи к элементам. Для ввода подписи к одному из блоков достаточно дважды щелкнуть по нему правой или левой кнопкой мыши. При этом программа автоматически перейдет в режим ввода текста. Если размеры элемента или вводимого текста слишком малы для комфортной работы, масштаб изображения будет автоматически увеличен (не рекомендуется).

Закончив редактирование надписи, необходимо щелкнуть левой кнопкой мыши за ее пределами, при этом произойдет выход из режима редактирования текста.

Соединительные линии также могут быть снабжены поясняющими надписями. Чтобы ввести такую надпись, необходимо дважды щелкнуть на самой линии левой кнопкой мыши. Если надпись и линия сливаются друг с другом, можно выделить линию щелчком левой кнопки мыши и перенести надпись в сторону при помощи желтого маркера.

Редактируя поясняющие надписи, можно использовать различные средства оформления, например, сделать текст жирным, изменить его размер или выделить часть надписи цветом.

Самостоятельные надписи. Поскольку самостоятельные надписи или текстовые блоки являются отдельными элементами документа, то их можно перемещать и настраивать независимо от других объектов. Такие надписи удобно использовать для оформления заголовков, подписей или выносных комментариев в тех случаях, когда их размещение внутри блоков невозможно или нежелательно.

Для создания самостоятельного текстового элемента воспользуйтесь инструментом **Текст**. Выбрать его можно при помощи кнопки, расположенной на панели инструментов.

Щелкнув один раз левой кнопкой мыши на свободном участке листа, можно быстро создать элемент-надпись и сразу же приступить к вводу текста (рисунок П5.12). Закончив набирать текст, щелкните левой кнопкой мыши за пределами надписи.

Следует отметить, что при изменении размера элементов в *Visio* размер шрифта надписей не изменяется. Для того чтобы уменьшить или увеличить размер символов, можно воспользоваться средствами настройки форматирования.

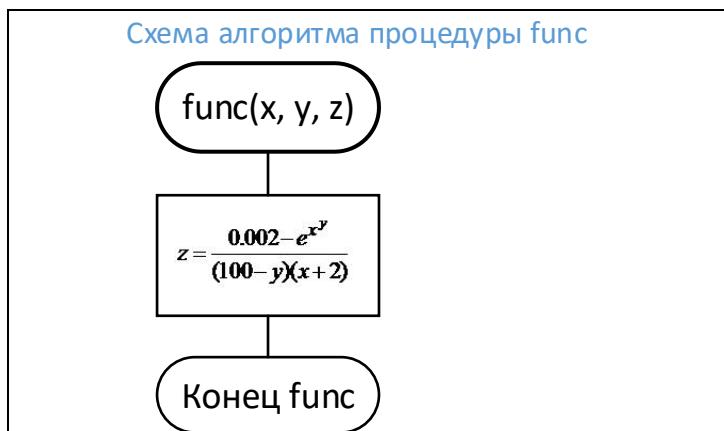


Рисунок П5.12 – Создание самостоятельной надписи с использованием инструмента **Текст**

Выносные элементы и комментарии. Эти элементы находятся в наборах фигур *Visio* и добавляются к иллюстрации точно так же, как любые другие стандартные блоки, однако, элементы комментариев имеют некоторые особенности, делающие их удобными для работы с большими фрагментами текста.

Самым распространенным видом выносного текстового элемента является так добываемое **Примечание**. Этот блок включается во многие наборы блоков.

Для того чтобы создать примечание, необходимо перенести соответствующий элемент в панели **Фигуры** на лист документа. После этого расположить его в нужной точке листа и ввести текст.

При помощи маркеров, появляющихся при выделении элемента, можно изменить размер блока, отведенного для размещения текста, расположение выносной линии или, например, длину боковых линий, охватывающих текст (рисунок П5.13).

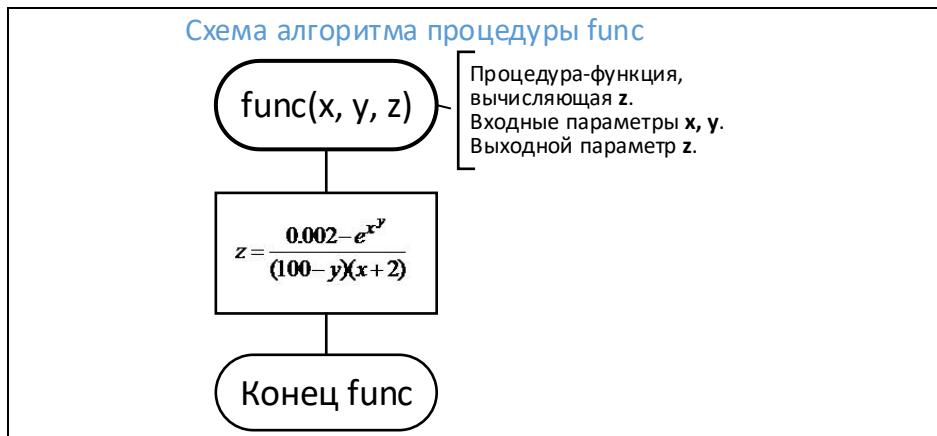


Рисунок П 5.13 – Создание примечаний к блокам

Если, после завершения ввода текста, необходимо вернуться к его редактированию, надо еще раз дважды щелкнуть на нужном элементе левой кнопкой мыши.

Вставка специальных символов. В некоторых случаях, особенно при подготовке технических или научных материалов, возникает необходимость использовать в тексте специальные символы, например, обозначение градуса («°») или знак «плюс-минус» («±»)

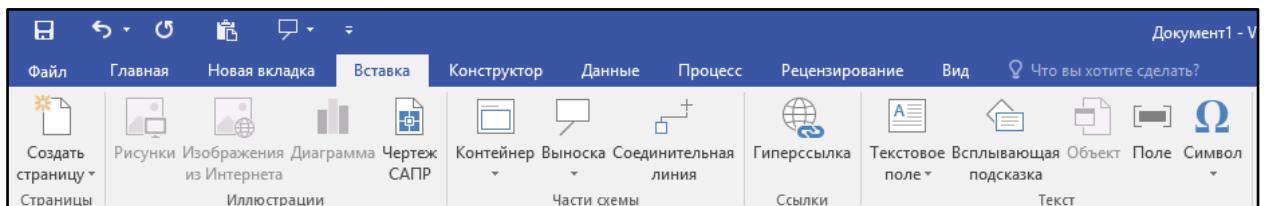


Рисунок П5.14 – Вкладка *Вставка*

Для того чтобы поместить в текст символ, можно воспользоваться командой **Символ** вкладки *Вставка* (рисунок П5.14). При вызове этой команды на экране появится окно, представленное на рисунке П5.15.

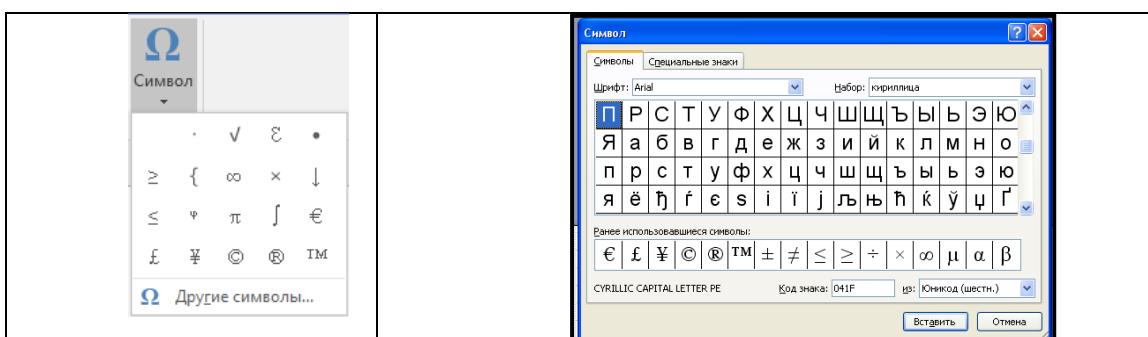


Рисунок П5.15 – Окно команды *Символ*

При помощи раскрывающегося списка ***Шрифт*** здесь можно выбрать начертание символов, с которым вы хотите работать, и щелкнуть по кнопке ***Вставить***.

Работая с текстом, можно свободно пользоваться командами инструмента ***Буфер обмена*** для копирования фрагментов надписей или отдельных символов. Или, работая с несколькими документами ***Visio*** одновременно, вы можете копировать между ними не только фрагменты текста, но и блоки целиком. При этом будут сохранены все параметры их внешнего вида.