

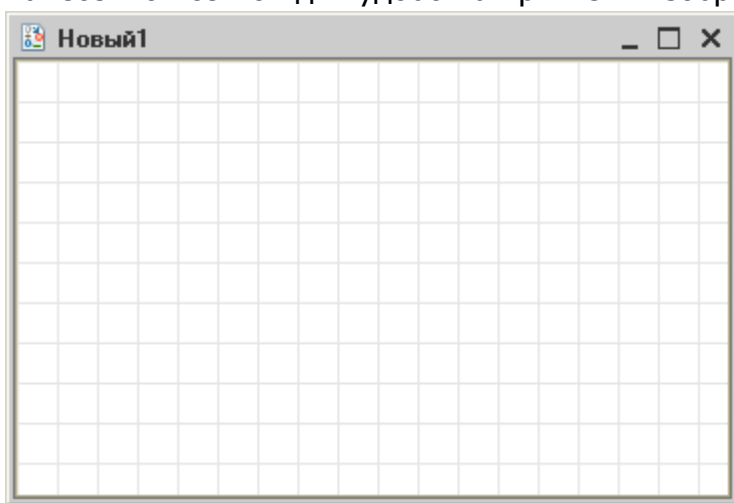
Встроенный в платформу «1С:Предприятие» векторный графический редактор *Графическая схема* предназначен для разработки *Карты маршрута бизнес-процесса*, хранящейся в конфигурации, или других файлов, хранящихся на диске.

Создание новой схемы, которая в дальнейшем будет храниться в отдельном файле, можно выполнить в режиме *Конфигуратор* или *1С:Предприятие* одним из способов:

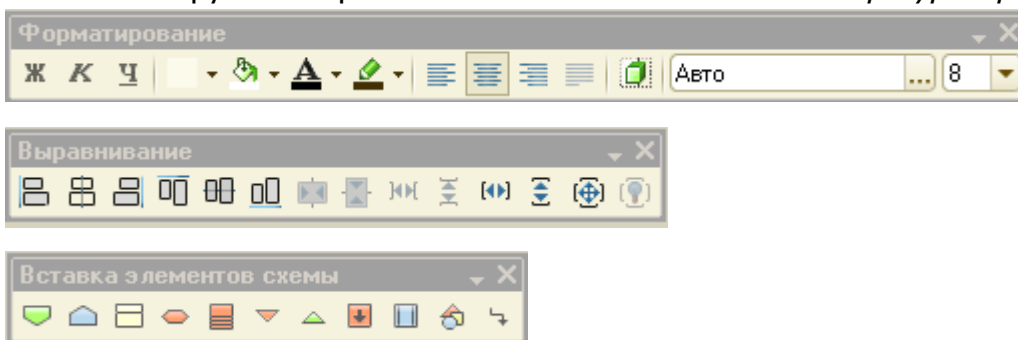
- *Файл – Новый – Графическая схема – ОК;*
- сочетание клавиш *Ctrl+N – Графическая схема – ОК .*

Для создания схемы, хранящейся в конфигурации, используют кнопку *Карта маршрута* на закладке *Прочее* в окне редактирования бизнес-процесса.

Окно графического редактора в режиме редактирования содержит рабочее поле с нанесенной сеткой для удобства привязки изображения:

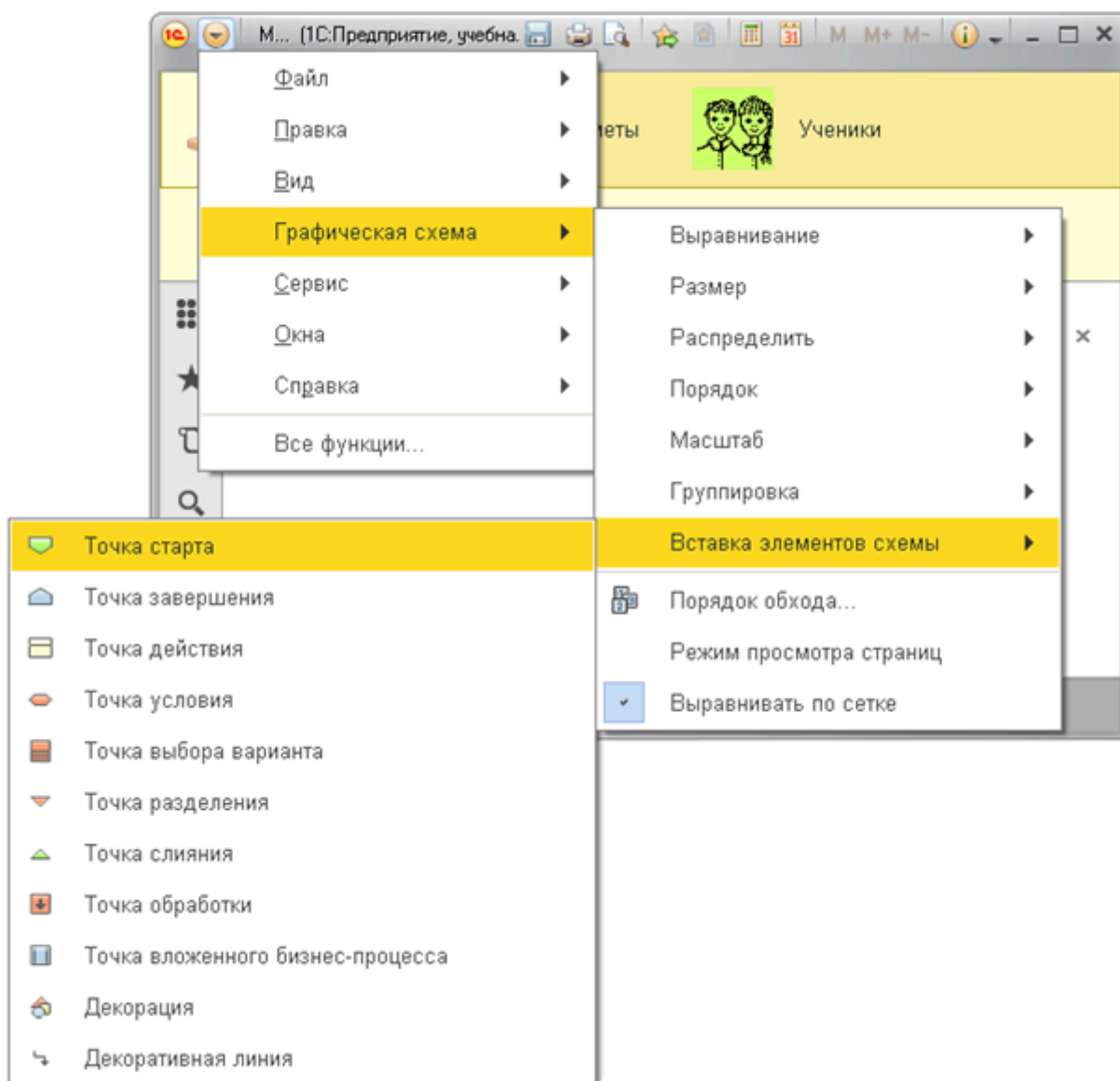


Панели инструментов расположены в основном окне *Конфигуратора* :




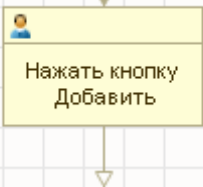
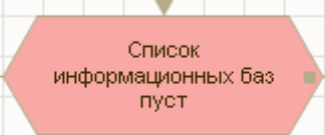
Состав команд интуитивно понятен.

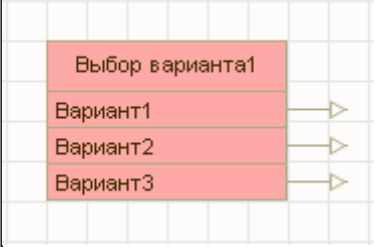
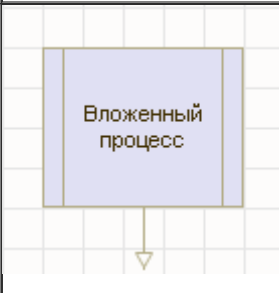
При создании графической схемы в режиме *1С:Предприятие* команды для разработки схемы вызываются из главного меню:



Панель инструментов содержит команды для создания карты маршрута бизнес-процесса.


Рассмотрим назначение некоторых блоков панели инструментов *Вставка элементов*:

Старт – начало выполнения алгоритма или какого-либо процесса	
Точка действия. Выполнение одной или нескольких операций	
Точка условного перехода – используется для ветвления процесса, в зависимости от условия Может применяться для циклов с предусловием, постусловием и условной развилки	

Выбор варианта – используется для выбора одного из нескольких возможных путей движения алгоритма	
Вложенный процесс – используется для упрощения схем сложных алгоритмов, а также для выделения одинаковых частей разных алгоритмов (предполагается, что какая-то часть алгоритма описывается отдельно, а в общую блок-схему весь вынесенный фрагмент включается как один блок)	
Завершение – последний этап в жизненном цикле процесса	

Декорация – может принимать различные формы, используется только для оформления.

Будем использовать редактор графических схем для разработки блок-схем известных алгоритмов. Для создания традиционных элементов блок-схемы можно применить команду

 **Декорация**. В свойствах этой команды указывается фигура и вводится текст заголовка. Например:



Задание 1

Разработайте блок-схему в редакторе *Графическая схема* в системе «1С:Предприятие» для алгоритма подсчета суммы цифр числа, введенного с клавиатуры.

Введем обозначения:

N – число, которое вводит пользователь с клавиатуры;

S – сумма цифр числа.

Подсчитать сумму цифр в натуральном числе n можно следующим образом: с помощью функции *СтрДлинаКоличество* узнать количество символов в записи числа и использовать цикл с заданным числом итераций. Алгоритм решения задачи выглядит так:

1. Определим количество цифр в записи числа (найдем K).
2. Положим сумму цифр в числе равной нулю ($S = 0$).
3. Выполним следующие (4, 5, 6) операции K раз:
4. Выделяем младшую цифру числа n .
5. Прибавляем выделенную цифру к сумме S .

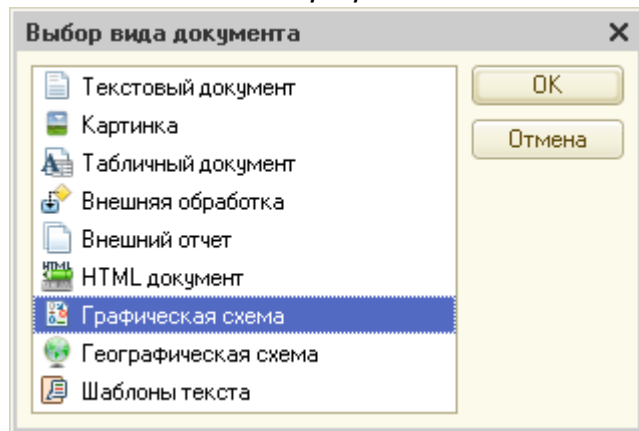
6. Из записи числа n удаляем младшую цифру.

7. Распечатаем значение S , конец алгоритма.

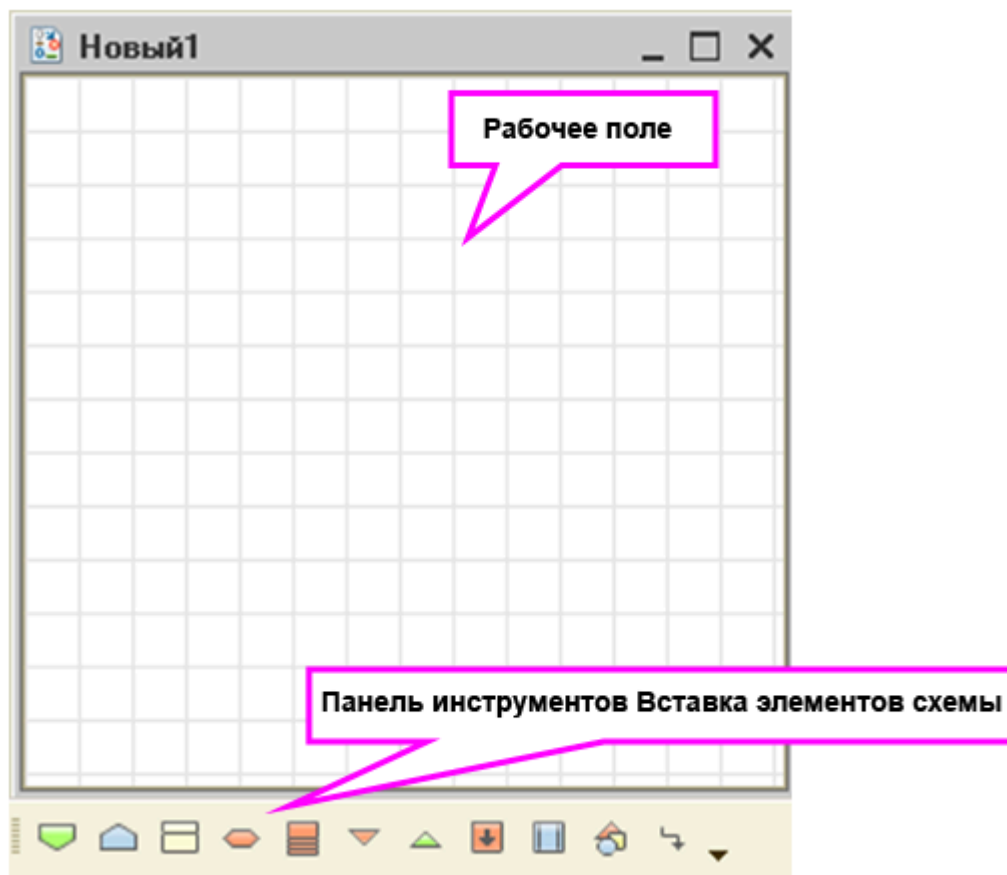
Разработаем блок-схему:

1. Откроем информационную базу в режиме *Конфигуратор*.

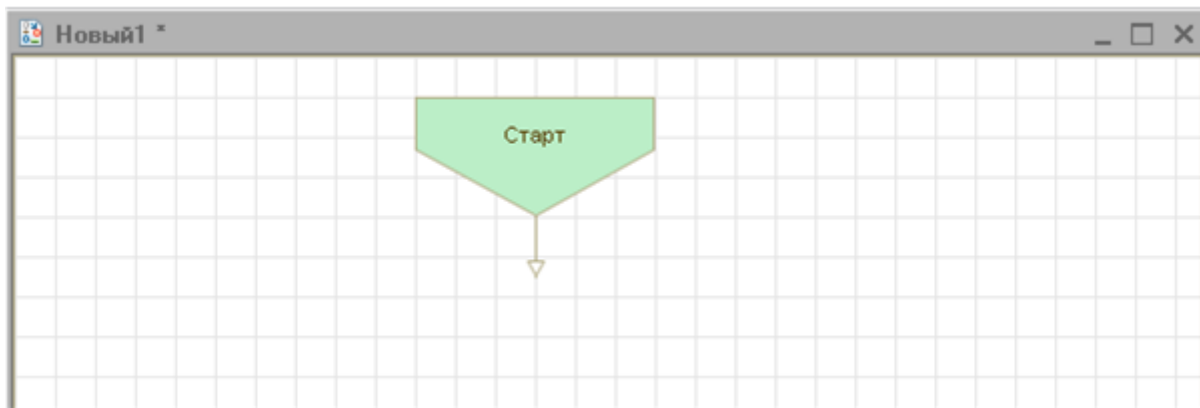
2. *Файл – Новый – Графическая схема – ОК*:




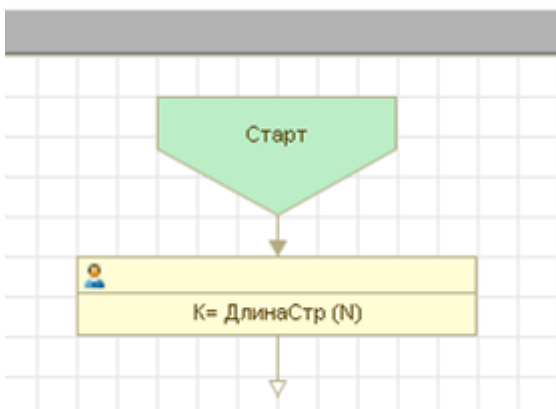
Откроется окно для разработки графической схемы:



Выделим *Старт* и, удерживая левую клавишу мыши, укажем на рабочем поле место для него:



4. Разместим блок  *Действие*, в котором вычисляется количество цифр числа, в свойстве заголовок напишем первый шаг алгоритма:



В окне свойств укажем заголовок, по желанию пользователя можно настроить оформление:

Свойства: Точка действия

Основные:

- Имя: Действие1
- Заголовок: K = ДлинаСтр(N)
- Гиперссылка: ☐

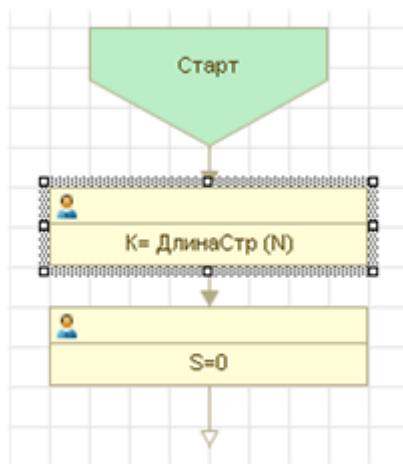
Характеристики:

- ГоризонтальноеПоложение: Центрировать
- ВертикальноеПоложение: Центрировать
- Подсказка:

Оформление:

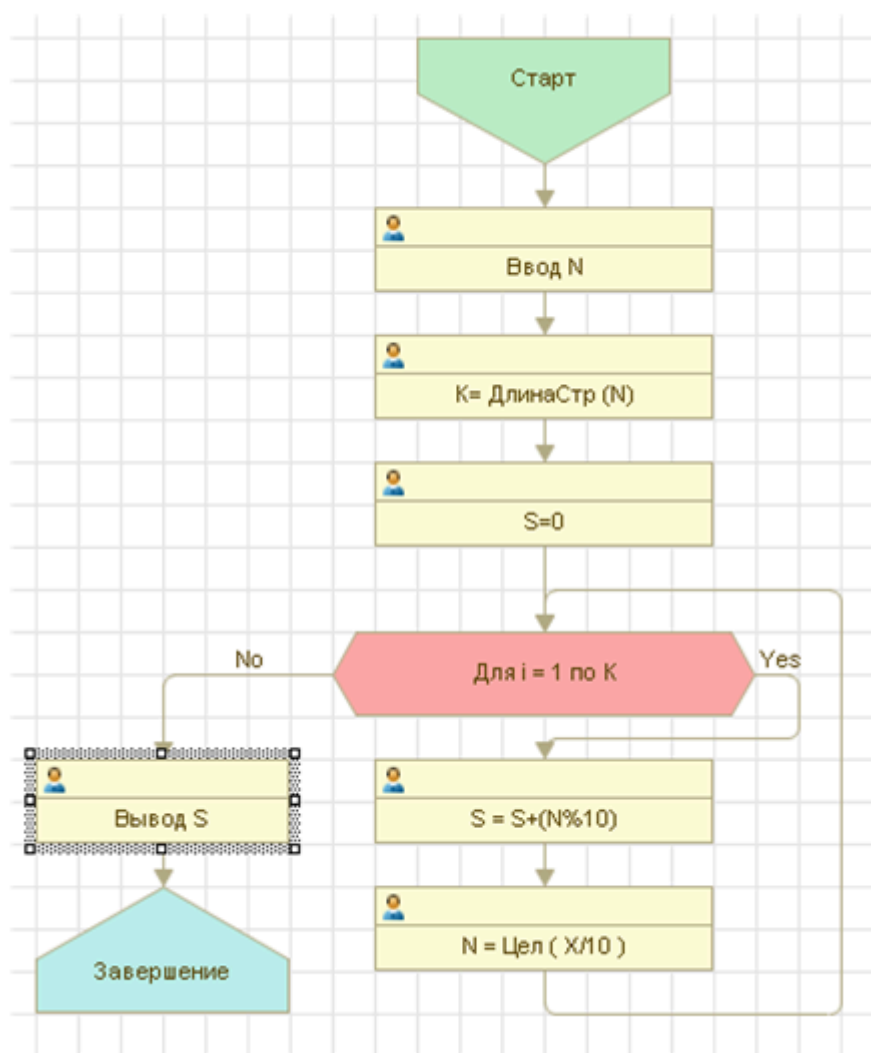
- ЦветФона: Авто
- ЦветЛинии: стиль: Рамка
- ЦветТекста: стиль: Текст формы
- Шрифт: Авто
- ПоложениеКартинки: Лево
- ПрозрачныйФон: ☐
- Картинка:
- РазмерКартинки: АвтоРазмер
- Рамка: — Сплошная

5. Аналогично создадим блок, в котором сумма цифр числа «обнуляется»:



6. Добавим блок для цикла с параметром  и запишем ограничения.

7. В результате получится следующая блок-схема:



Задание 2

Измените блок-схему алгоритма в задаче 1 так, чтобы она удовлетворяла следующему варианту алгоритма:

1. Положим сумму цифр в числе равной нулю ($S = 0$).
2. Если в записи числа n остались цифры, то выполняем п. 3–6, иначе переход на п. 7.
3. Выделяем младшую цифру числа n .
4. Прибавляем выделенную цифру к сумме S .

5. Из записи числа n удаляем младшую цифру.
6. Переходим на п. 2.
7. Распечатываем значение S , конец алгоритма.

Задание 3

Разработайте блок-схему для алгоритма Евклида.

Алгоритм Евклида может быть представлен в двух вариантах.

Вариант I

Алгоритм нахождения НОД (делением):

1. Если первое число не делится на второе число без остатка, идем дальше, иначе второе число и есть НОД (следует выйти из цикла – к п. 4).
2. Если есть остаток, то первому числу присваиваем значение делителя, а второму – остатка от деления.
3. Переходим к п. 1.
4. Вывод НОД.

Блок-схема

Вариант II

Алгоритм нахождения НОД (вычитанием):

1. Если числа не равны друг другу, то переходим к следующему пункту, иначе любое из них является НОД (следует выйти из цикла – к п. 4).
2. Если числа не равны, то из большего вычитаем меньшее и большее число заменяем на результат вычитания.
3. Переходим к п. 1.
4. Вывод НОД.

Блок-схема

Задание 4

Разработайте блок-схему для алгоритма проверки, является ли число простым.

Рассмотрим два варианта словесного описания алгоритма.

Вариант I

Для определения, является ли число простым, необходимо проверить, есть ли у него делители, кроме единицы и самого себя. Алгоритм может выглядеть так:

1. Если число равно единице, то оно не простое, и переходим к п. 7.
2. Признаем число простым, пока не доказано обратное.
3. Если шаг $\leq \sqrt{n}$ и число не является составным, переходим к следующим пунктам, иначе заканчиваем выполнение цикла и переходим к п. 7.
4. Находим остаток от деления числа n на i .
5. Если остаток равен 0, признаем число составным.
6. Увеличиваем i на 1 и возвращаемся к п. 3.

7. Сообщаем, является ли число простым.

Вариант II

Можно сократить число проверок в программе, учитывая, что все четные числа составные, исключим их сразу и будем проводить поиск с шагом 2. Алгоритм немного изменится:

1. Если число равно единице или делится на 2, то оно не простое и переходим к п. 7.
2. Признаем число простым, пока не доказано обратное.
3. Если шаг $\leq \sqrt{n}$ (начинаем с $i = 3$) и число не является составным, переходим к следующим пунктам, иначе заканчиваем выполнение цикла и переходим к п. 7.
4. Находим остаток от деления числа n на i .
5. Если остаток равен 0, признаем число составным.
6. Увеличиваем i на 2 и возвращаемся к п. 3.
7. Сообщаем, является ли число простым.

Составьте для каждого варианта блок-схему.

Если у вас по каким-либо причинам не получились блок-схемы, то познакомиться с результатом работы можно так: сохраните следующие файлы и откройте в 1С:Предприятие.

Задание 1

Задание 3 - Вариант 1

Задание 3 - Вариант 2