

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ

Заведующий методическим кабинетом

_____ Е.В.Фалей

« ____ » _____ 2018 года

Специальность: 2-40 01 01 «Программное обеспечение информационных технологий»	Дисциплина: «Основы алгоритмизации и программирование»
---	--

Лабораторная работа № 32
Инструкционно-технологическая карта

Тема: Разработка и оформление алгоритмических схем на программы и функции

Цель: Научиться создавать алгоритмические схемы для программ, функций, методов

Время выполнения: 2 часа

1. Краткие теоретические сведения

Алгоритм – это строго определенная последовательность действий, приводящая к решению задачи или демонстрации сообщения о невозможности найти решение данной задачи.

Алгоритм может описываться тремя способами:

1) человеческим языком (метаязыком, псевдокодом): например, «если А больше Б, то увеличить А в Б раз, иначе уменьшить А на величину Б»;

2) графическим способом (в виде логических схем, которые раньше чертили на бумаге вручную, а сейчас используют специализированные программы, например, Microsoft Office Visio);

3) собственно кодом программы на низкоуровневом (например, ассемблер) или высокоуровневом (C, C++, C#, Java, Object C, SWIFT, Python) языке программирования.

Далее мы рассмотрим основные элементы, используемые в логических схемах программ и функций. Описание логических схем формализовано государственными и международными стандартами, например, ГОСТ 19.701-90 и ISO 5807-85.

Использование термина «блок-схемы» нежелательно, поскольку данное понятие было изъято из международной документации стандартов к концу 1980-х годов. Лучше заменить его другими, например, «логическая схема программы (функции, метода, алгоритма)».

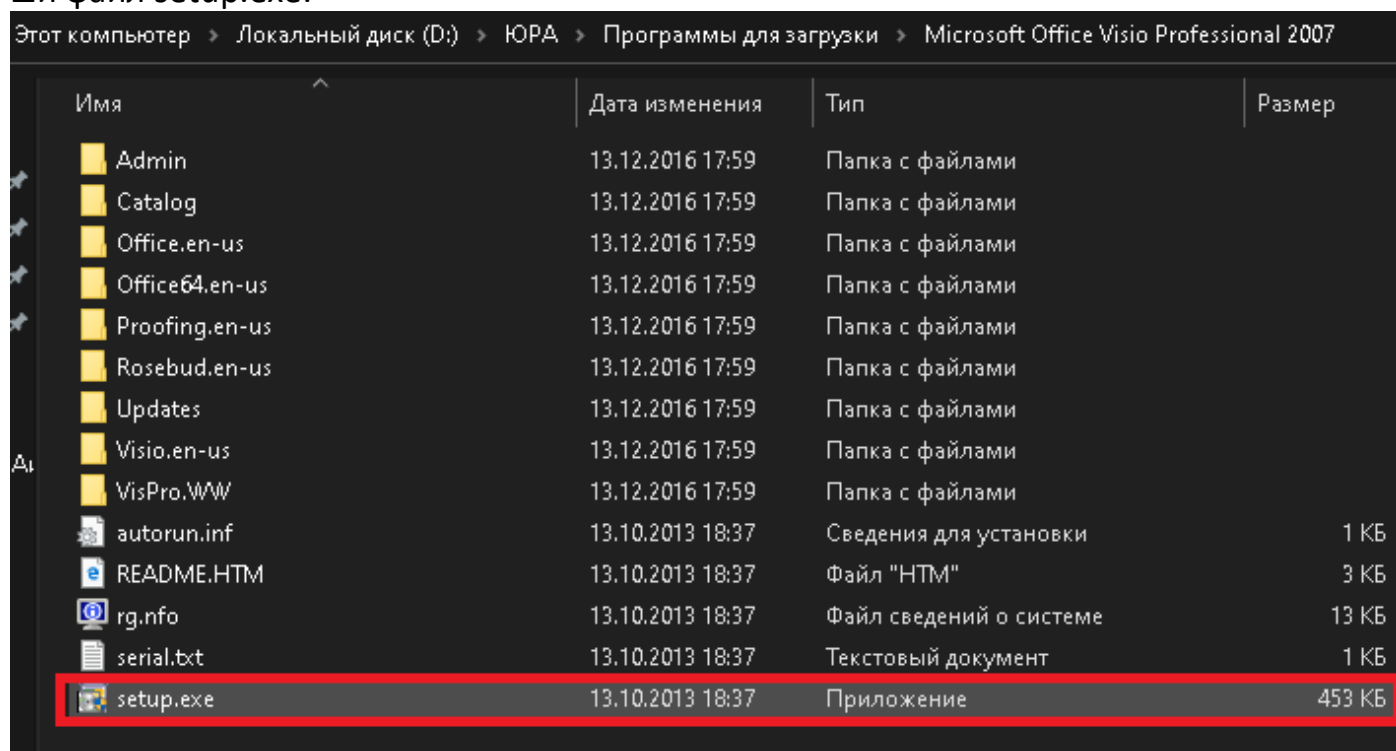
Задание 1.

Установить на свой компьютер программу MS Visio, создать в ней логические схемы, реализовав все типы элементов кода (ветвления, циклы и т.д.).

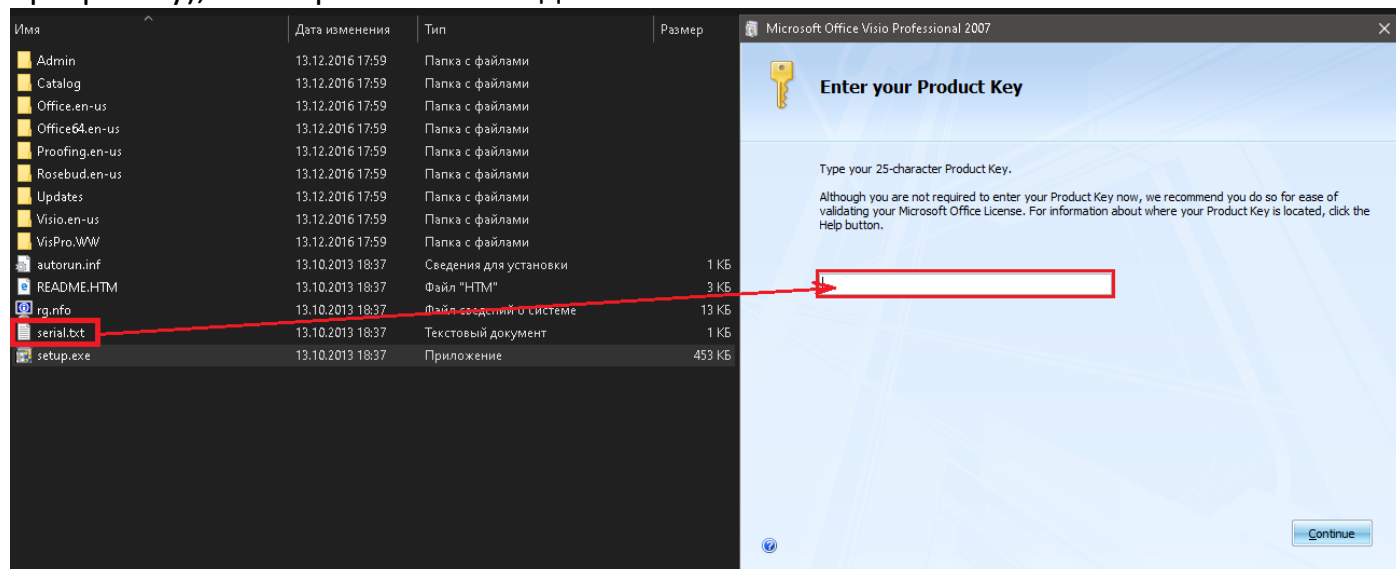
Для создания логических схем используется программа Microsoft Office Visio, которая входит в пакет прикладных программ Microsoft Office. Особенностью этой про-

граммы является то, что ее «старые» версии имеют весь необходимый набор элементов для отображения логических схем, а новые версии имеют интерфейсные нововведения, но необходимые для разработчика графические элементы нужно отыскивать в различных наборах, ввиду чего работа в старой версии программы проще и быстрее. Из-за этого рассмотрим установку Microsoft Office Visio 2007 с англоязычным интерфейсом, который также является ее плюсом, поскольку во многих аутсорсных ИТ-компаниях практикуется максимальное использование английского языка. Эта версия программы протестирована на многих версиях операционной системы Microsoft Windows, например, Microsoft Windows 7 разрядности 32 бита, Microsoft Windows 10 разрядности 64 бит.

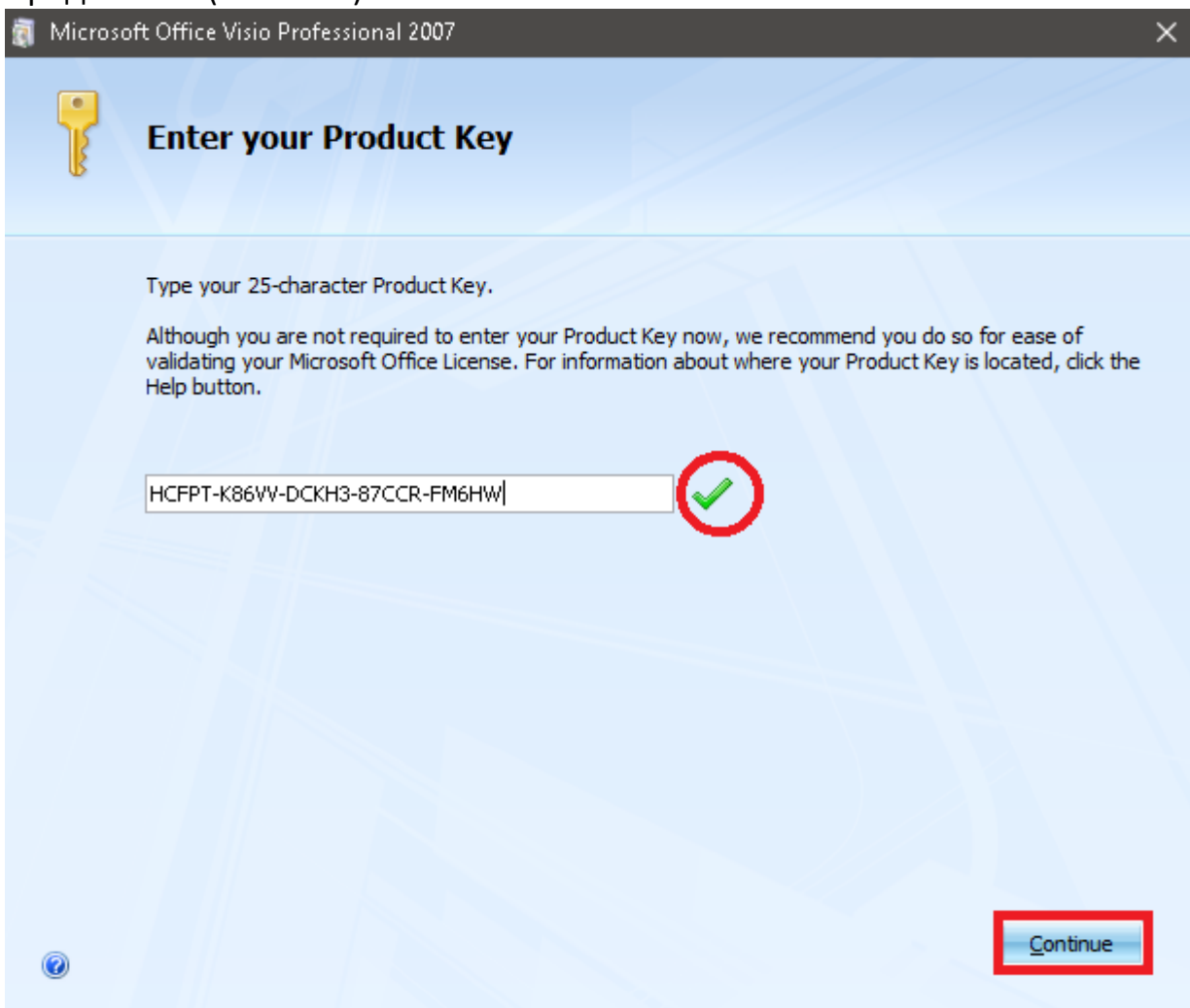
1. Откроем папку программы-установочника и запустим двойным щелчком мыши файл setup.exe:



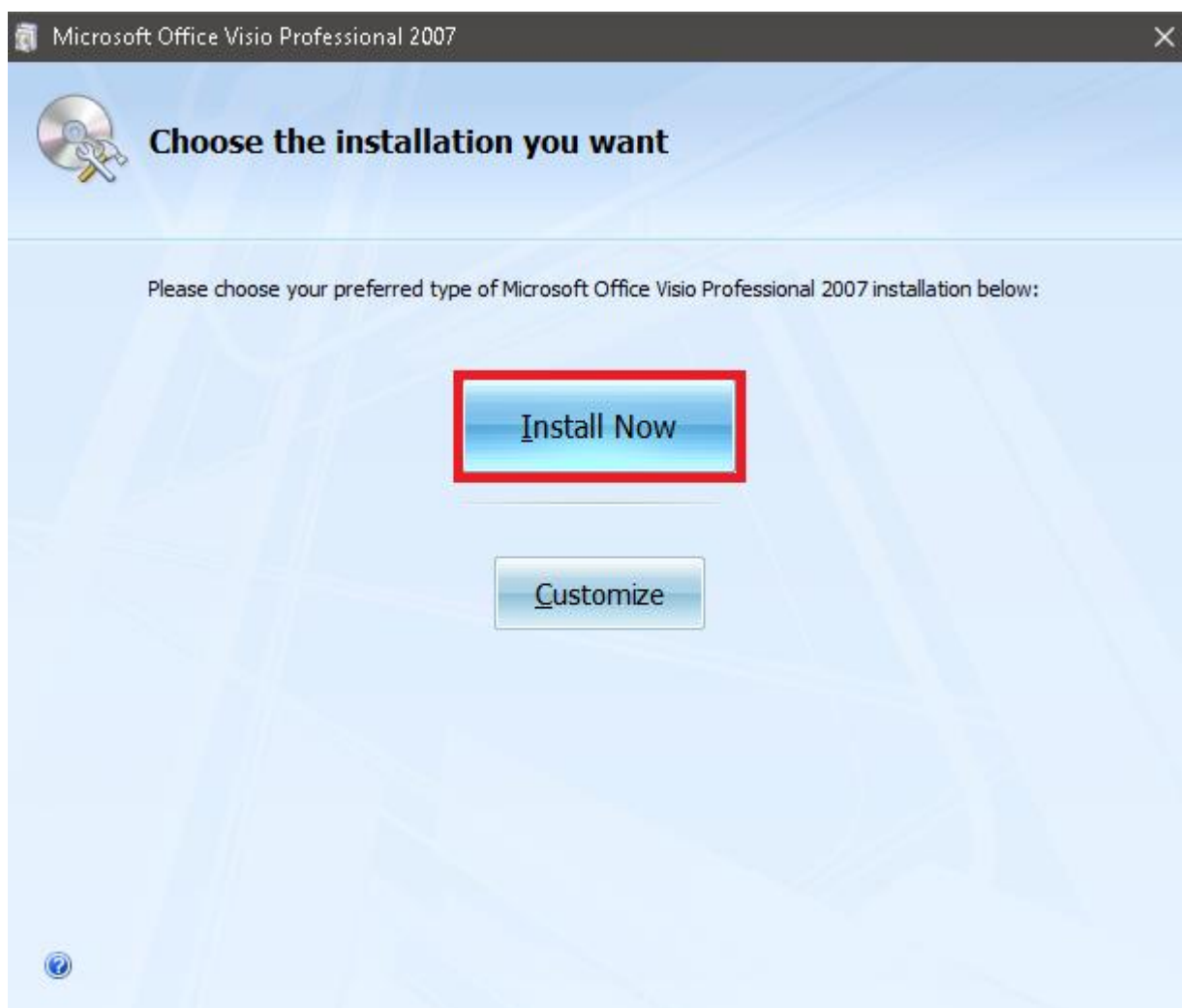
2. Разрешите ОС MS Windows внести изменения в систему (установить новую программу), на запрос ключа введите в поле значение:



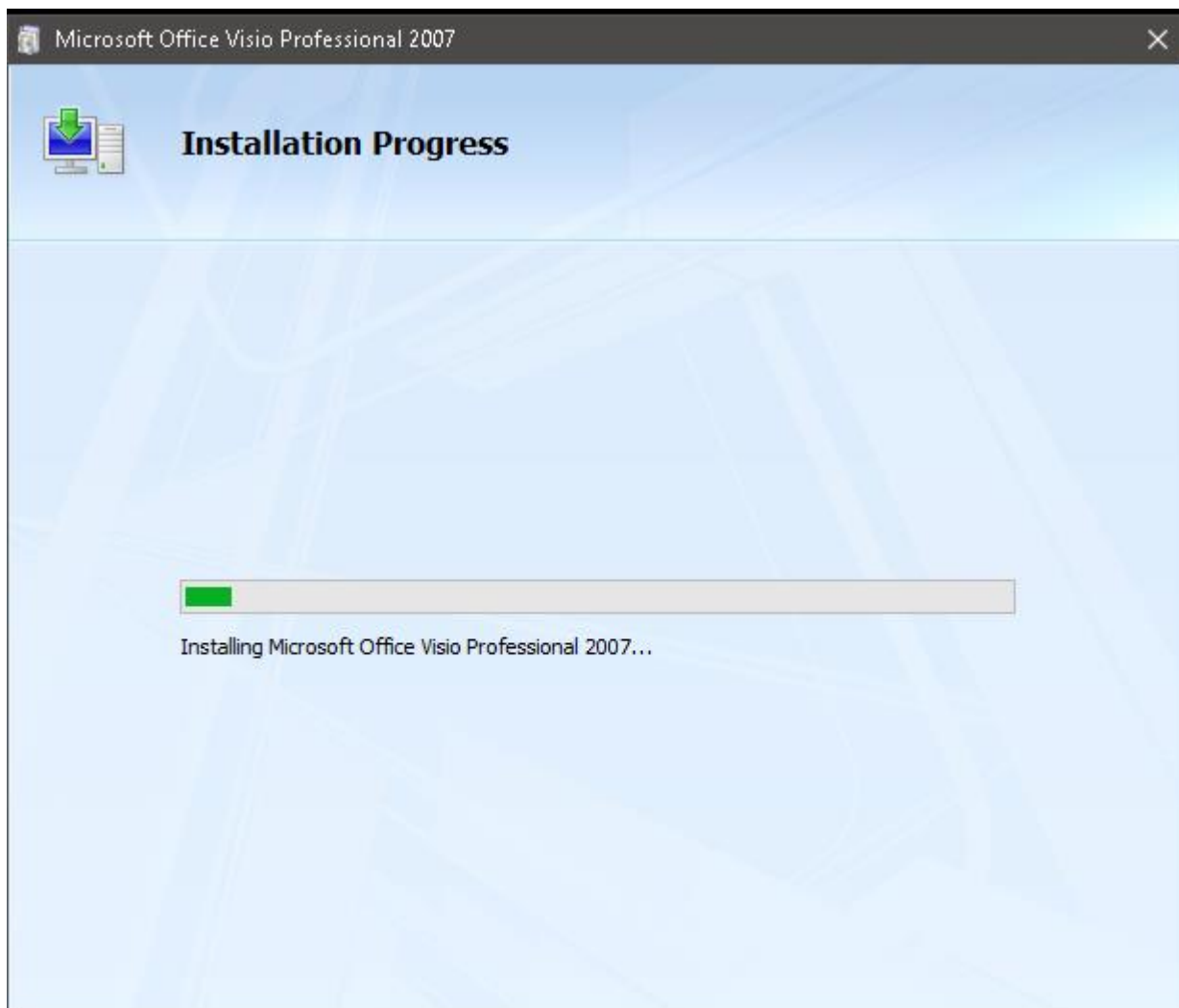
3. Если значение введено правильно и появилась «зеленая галочка», нажмите Продолжить (Continue):

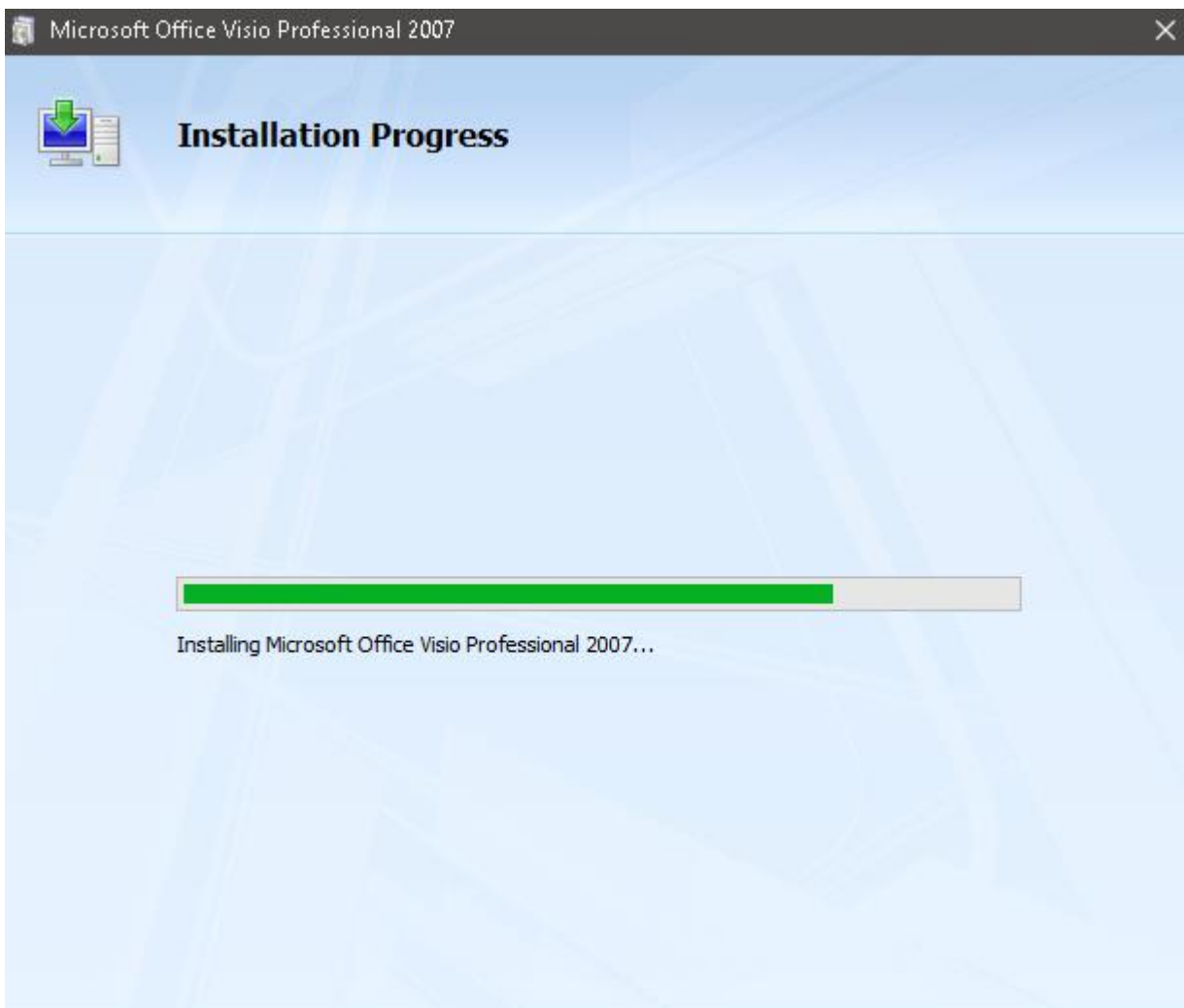


4. Выберите Установить сейчас (Install Now):

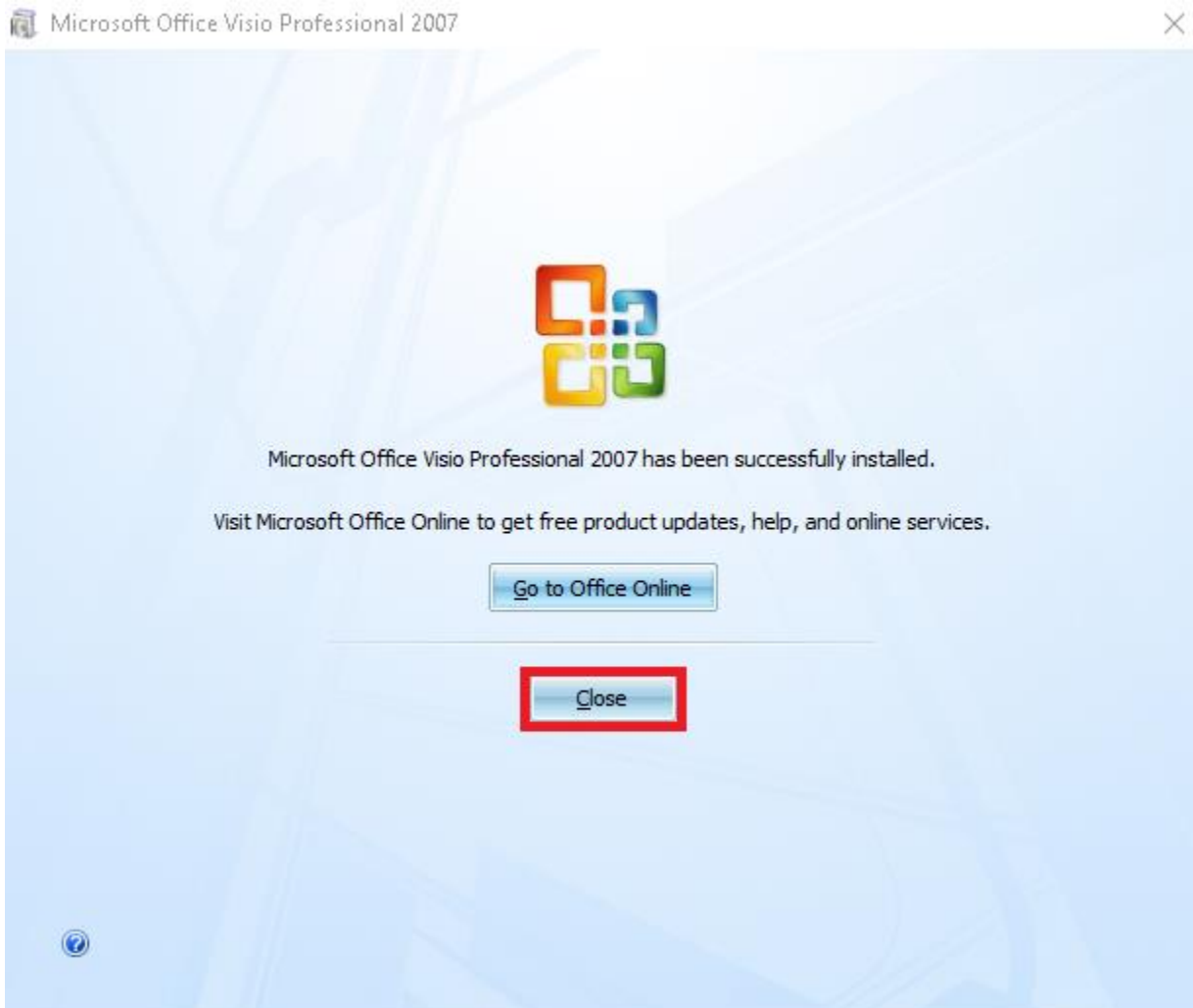


5. Дождитесь окончания установки:

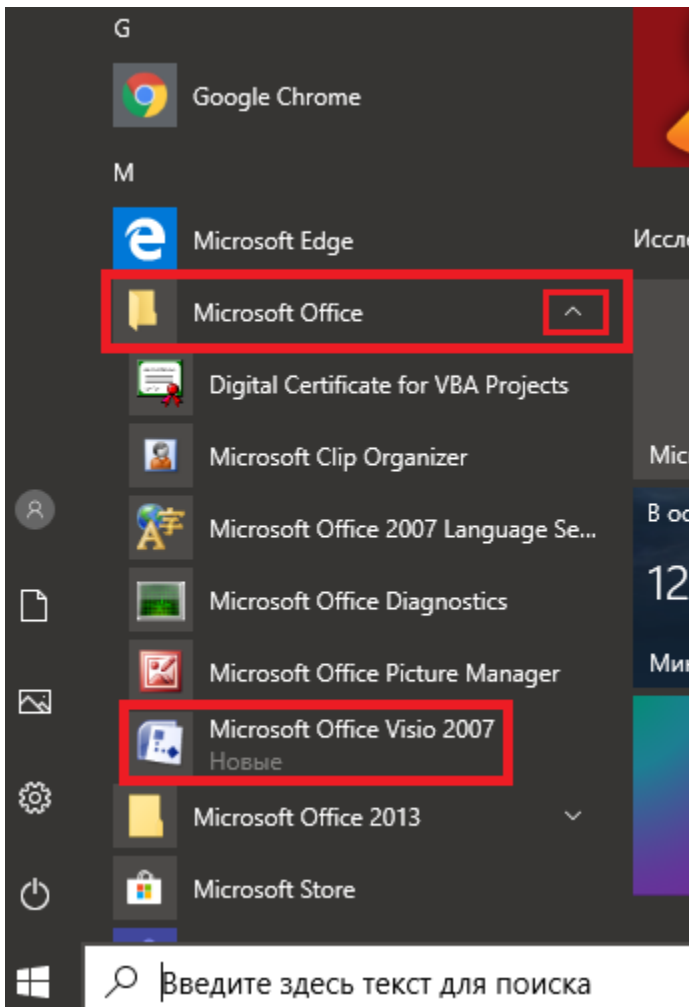




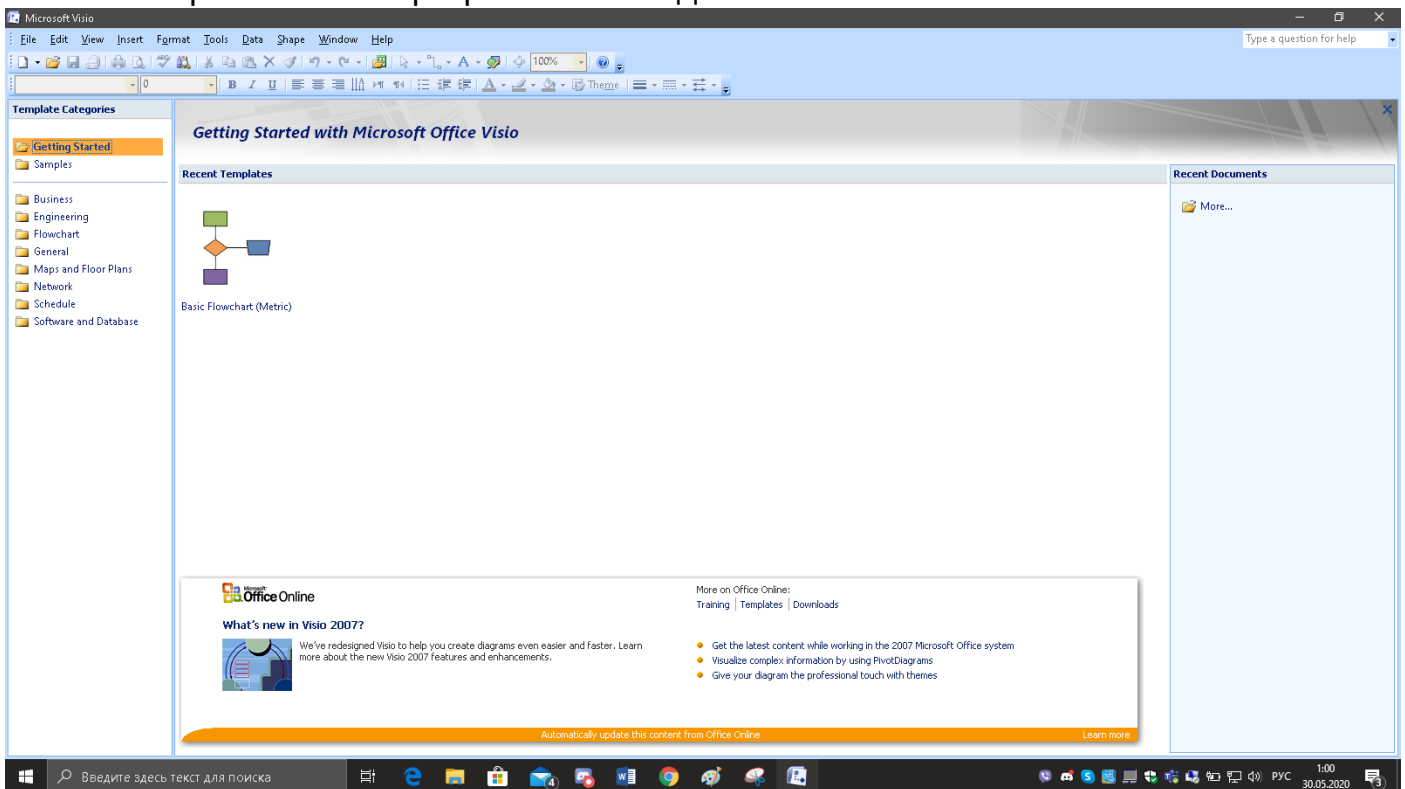
6. По окончании установки выберите Закреть (Close):



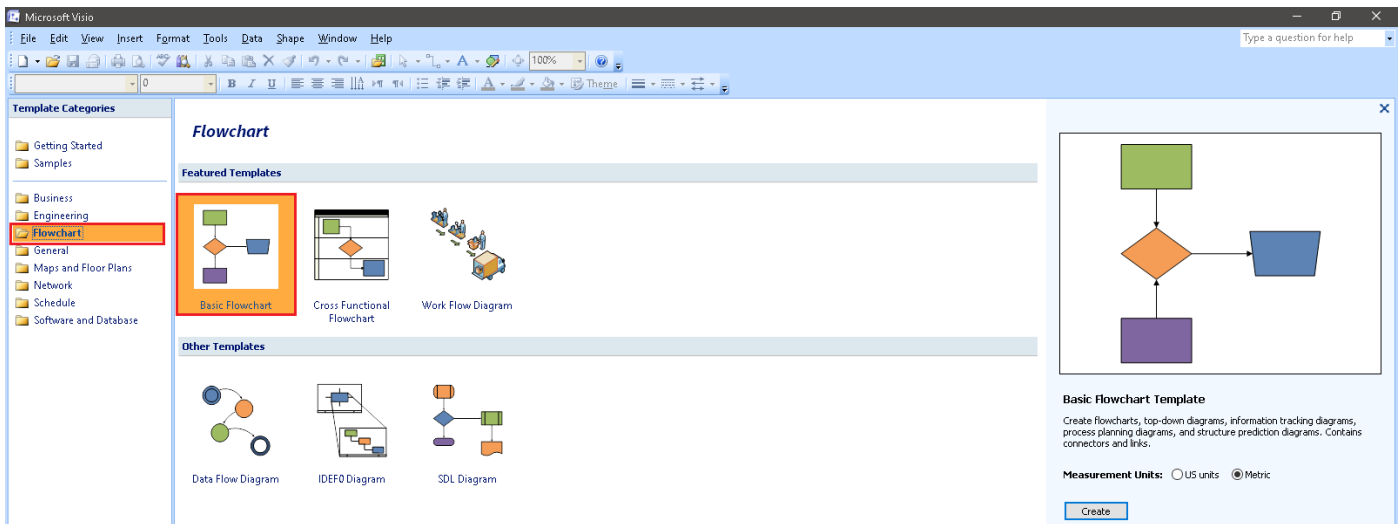
7. Проверяем появление новоустановленной программы в списке программ на вашем компьютере и запускаем ее двойным кликом мыши. Обратите внимание, если на вашем компьютере уже установлен MS Office другого года выпуска, то будет создана новая папка для данной программы данной версии.



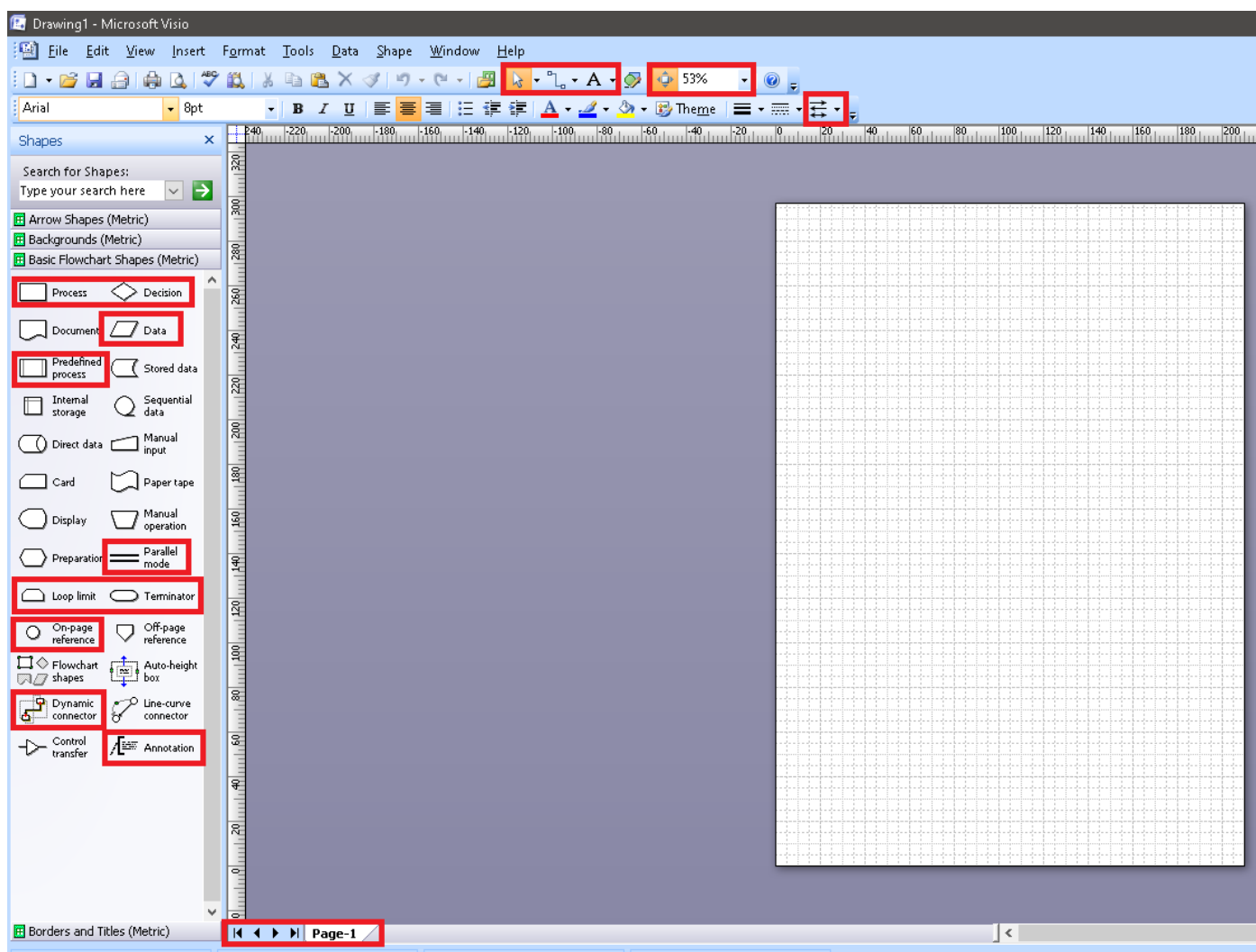
8. Стартовое окно программы выглядит так:



9. Выберите пункт меню слева Flowchart (Диаграммы) и в нем подпункт Basic Flowchart (Базовая диаграмма):



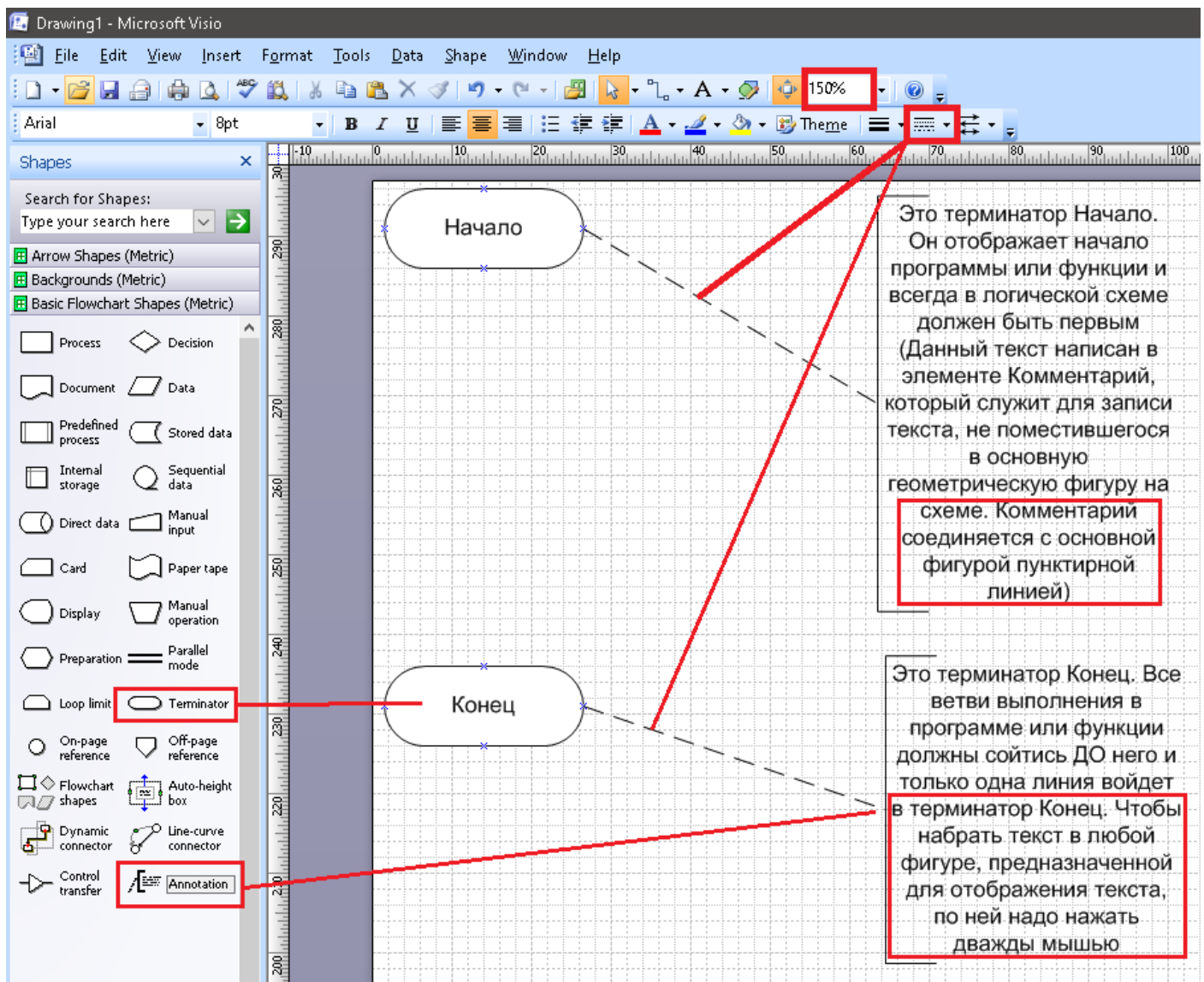
10. В наборе доступных элементов слева проверяем наличие основных графических фигур, нужных нам. Проверкой набора элементов на достаточность для создания логических схем можно считать наличие фигуры Loop limit (Ограничитель цикла). Переключаться между режимами рисования стрелок и редактирования линий и фигур можно с помощью кнопок «Стрелка» и «Ломаная линия» вверху. Правее находится листбокс изменения масштаба, который в начале работы следует задать на большую величину, чтобы можно было экономно располагать элементы начиная от края листа и точно располагать стрелки (сам лист при этом в действительности остается прежнего размера, например, A4, хотя формат листа можно менять). Кнопка «А» служит для расположения текста вне фигур (обычно это надписи «Да, Нет» над стрелками ветвлений). Внизу можно создавать новые страницы логических схем и переключаться между ними.



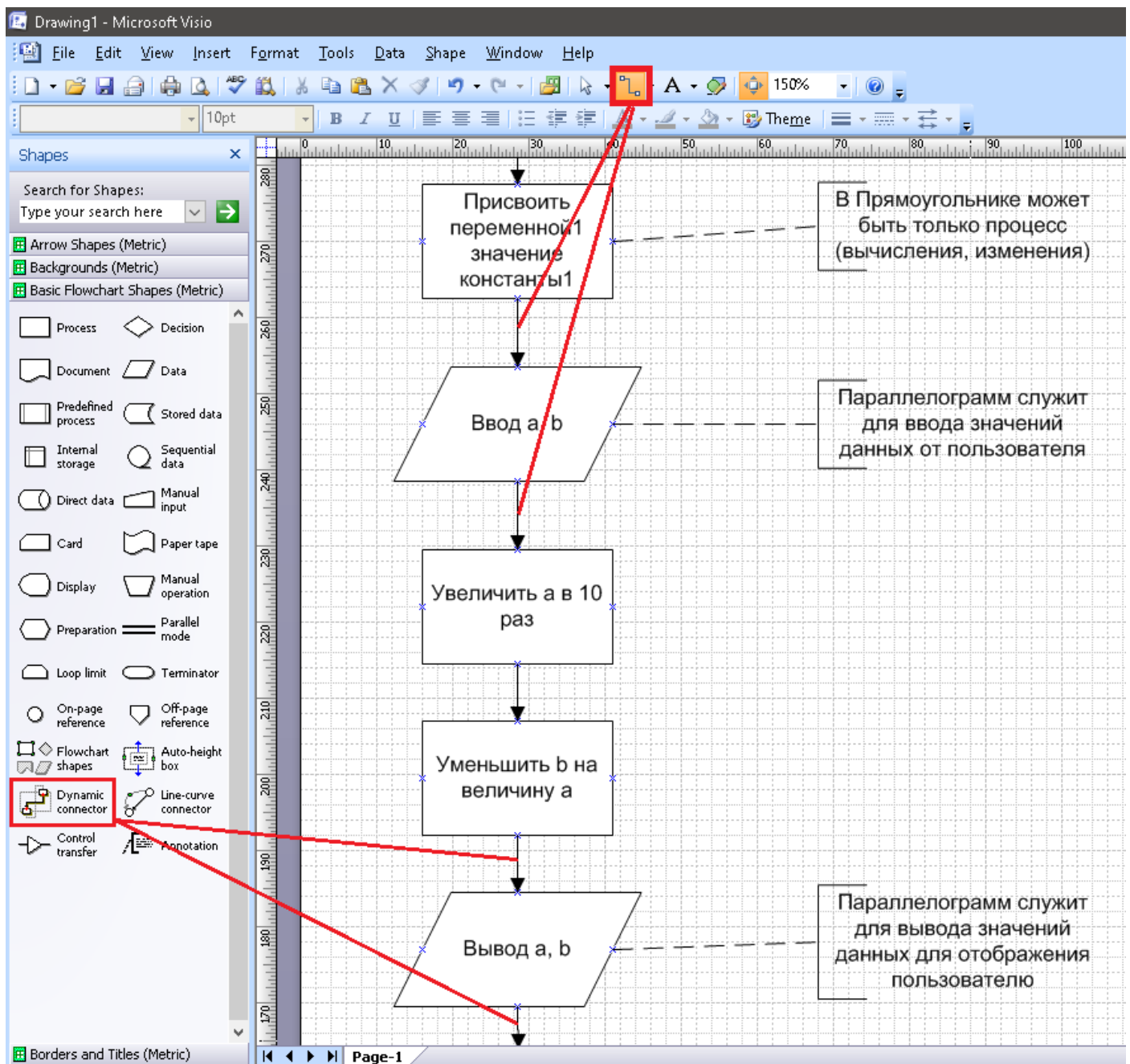
11. Фигура Терминатор, начинающая и заканчивающая любую логическую схему любой программы или функции. Терминаторы дополнены вспомогательными элементами Комментариями, которые соединяются с комментируемой фигурой пунктирной линией, но сама квадратная скобка комментария при этом должна остаться сплошной линией.

Весь текст в фигурах должен быть языконезависимым, то есть не содержать готового кода, который автоматически привязывает данную «схему» к конкретному языку программирования, делая саму схему узкоспециализированной и ненужной. Код во всех фигурах должен быть на естественном человеческом языке (русский язык), но при этом кратким, однозначным. Допускается использование латинского алфавита для именования переменных и иных объектов, хотя русские буквы, в том числе заглавные («большие»), тоже подойдут для этого.

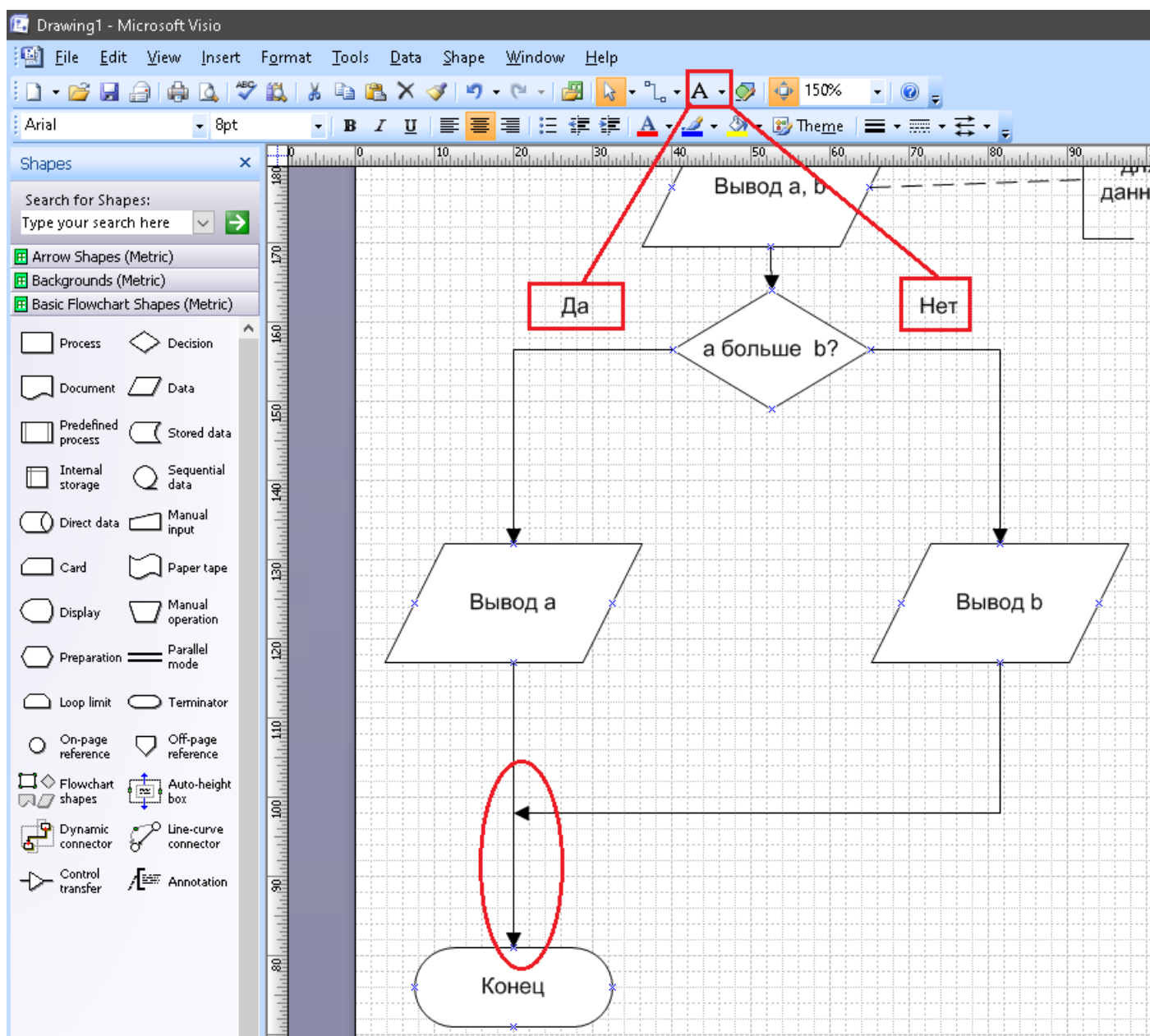
Учитывайте временной порядок выполнения выражений в коде. Например, фигура ветвления Ромб будет выше, чем дальнейшие действия в двух ветках за ней, но действия в этих ветках будут на одном уровне между собой (по крайней мере в начале веток), поскольку каждая из веток имеет шанс на выполнение вслед за проверкой ветвления.



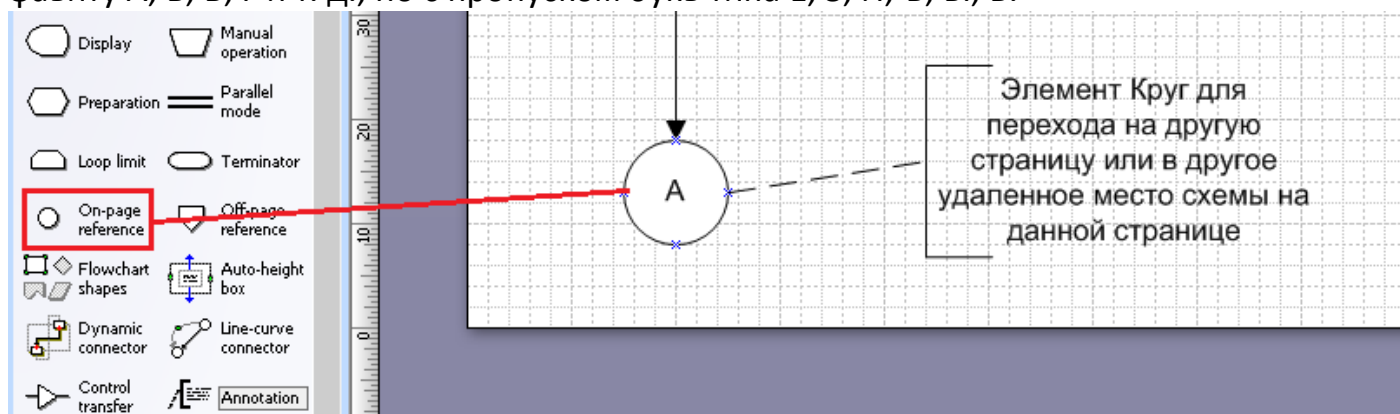
12. Элемент Прямоугольник для процессов (действий), Параллелограмм – для отображения ввода значений от пользователя или для отображения вывода значений из программы пользователю. Фигуры соединяются автоматическими «тянущимися» линиями, причем соседние фигуры можно даже автоматически соединять, наведя курсор мыши на соединяющие точки, расположенные по центрам сторон фигур.

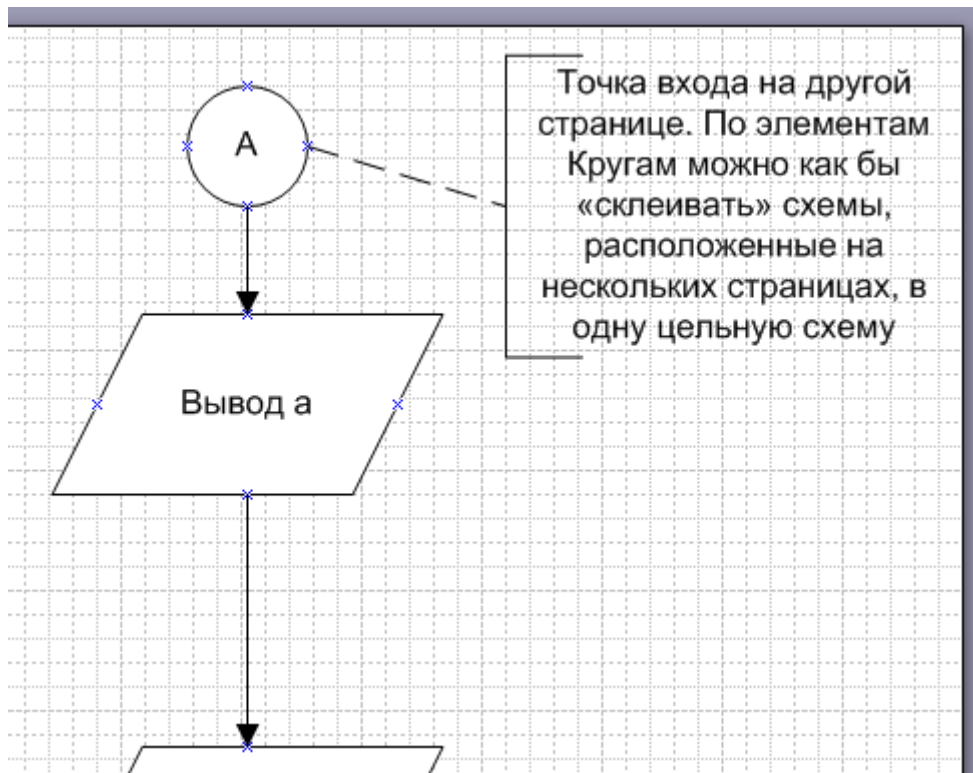


13. Ромб (ветвление) имеет только один вход (как и любая другая фигура), но, в отличие от остальных фигур, имеет два выхода, причем левый должен быть связан с положительным результатом проверки («Да»), а правый – с отрицательным результатом проверки («Нет»). «Да» всегда должен быть левее «Нет».



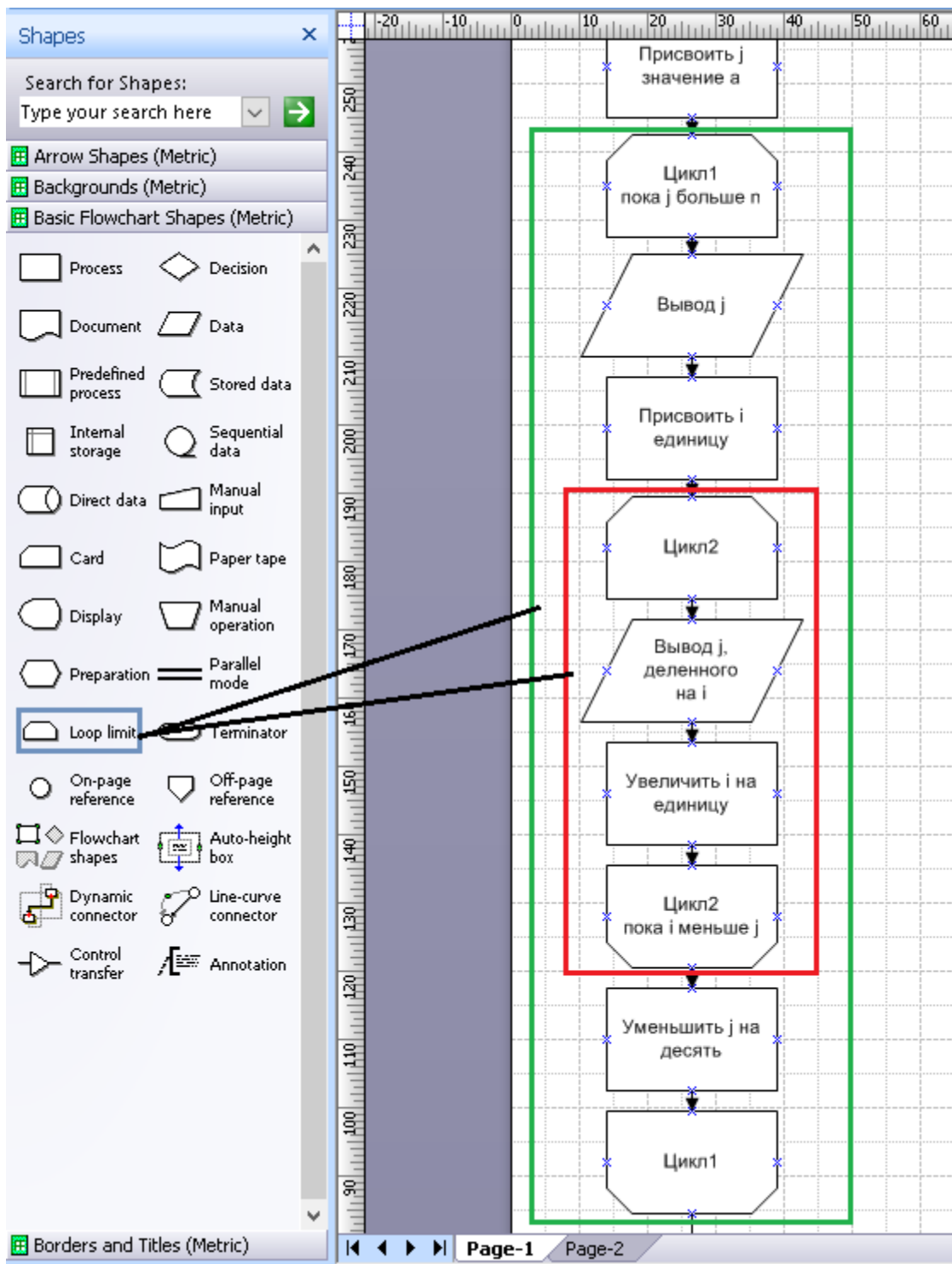
14. Элемент Круг (специальный символ Соединитель) служит для соединения схем, расположенных на нескольких страницах, в одну цельную схему, а также соединения разрывов схем на одной странице. Круги именуются большими буквами по алфавиту А, Б, В, Г и т. д., но с пропуском букв типа Ё, З, Й, Ъ, Ы, Ь.



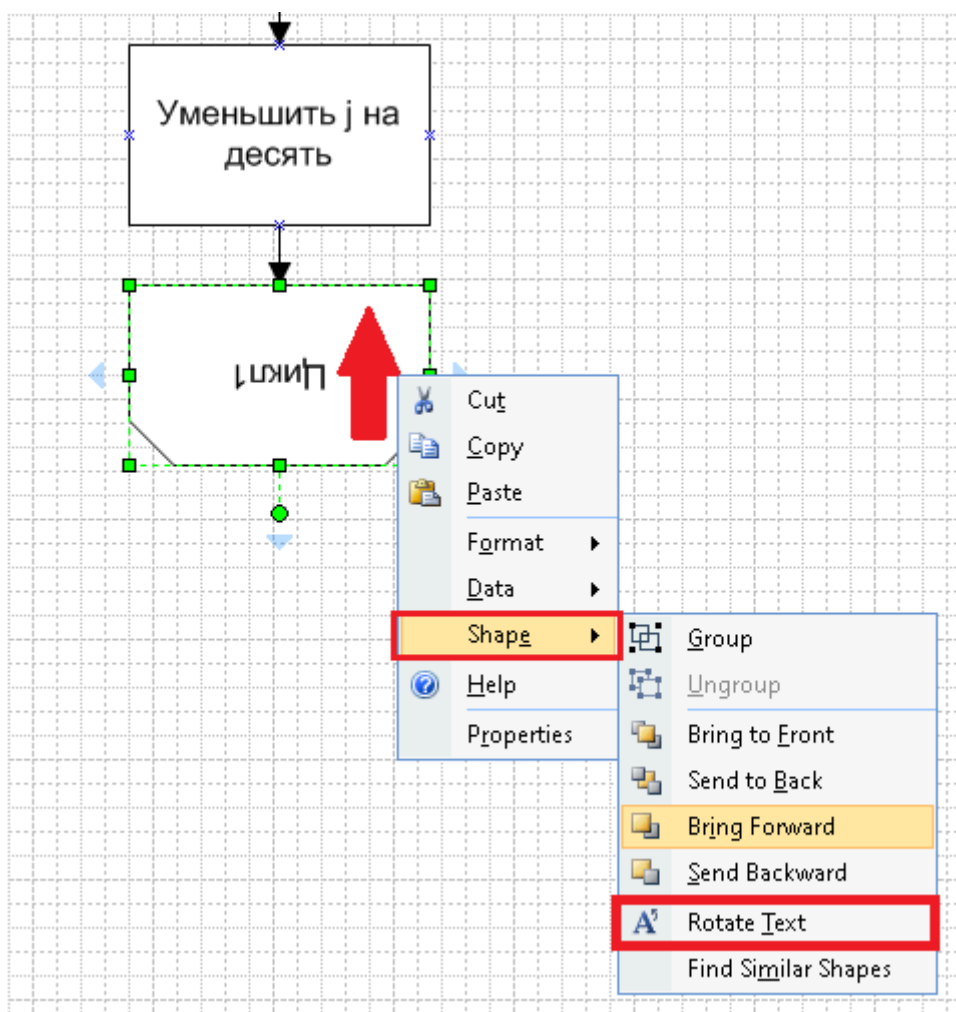


15. Хотя циклы можно собирать из ветвлений (ромбов), процессов (прямоугольников) и линий (стрелки направлений хода действий в итерациях и выхода из цикла), но если в алгоритме много циклов, то выгоднее применить специальную фигуру – Прямоугольник с двумя скошенными углами. В начале цикла ставится такая фигура и в конце цикла ставится снова такая фигура, которая переворачивается мышью на 180 градусов. Они обе содержат имя цикла по типу Цикл1, Цикл2, Цикл3 и т. д. Циклы нумеруются сквозным образом во всем данном алгоритме. Циклы могут вкладываться друг в друга. Не забывайте декларировать, инициализировать и изменять итерационную переменную, что нужно делать в отдельных графических элементах (прямоугольниках). Любой цикл начинается с имени, но если цикл с предусловием, то проверка условия его выполнения пишется в открывающей фигуре цикла, а если цикл с постусловием, то проверка условия продолжения цикла пишется в закрывающей фигуре, а в открывающей – только название такого цикла.

В примере ниже Цикл1 с предусловием, а Цикл2 – с постусловием. Цикл2 вложен в Цикл1.



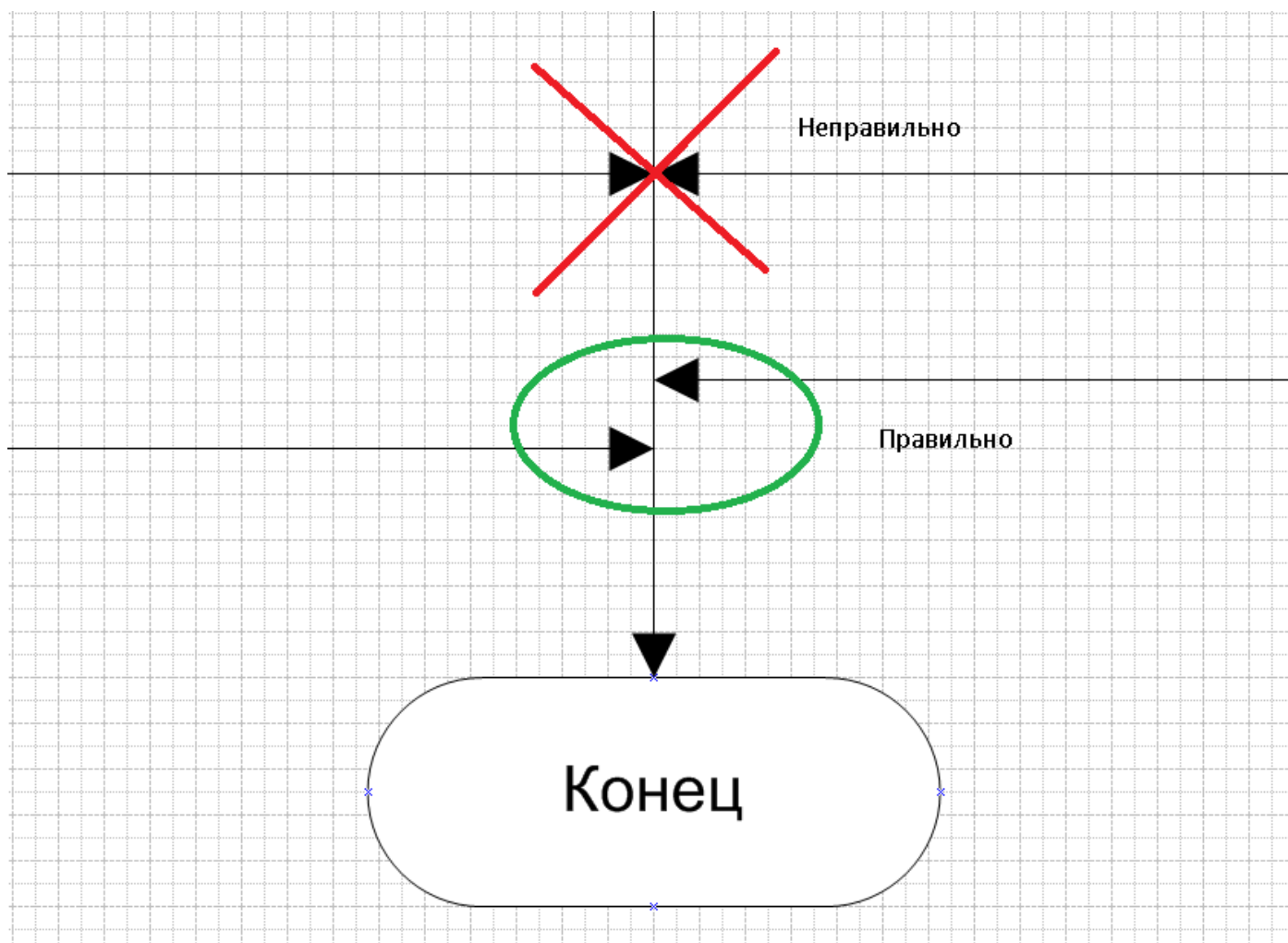
16. Чтобы перевернуть текст в завершающей фигуре цикла, которая переворачивается мышью «вверх ногами», в результате чего ее текст тоже оказывается перевернутым, надо правой кнопкой мыши нажать на участок фигуры, НЕ занятой текстом (иначе будет контекстное меню для символов текста), и в контекстном меню самой фигуры выбрать Shape (Форма) и Rotate Text (Перевернуть текст). Текст повернется на 90 градусов, поэтому надо эти действия повторить, и текст в фигуре перевернется суммарно на 180 градусов, что и требовалось.



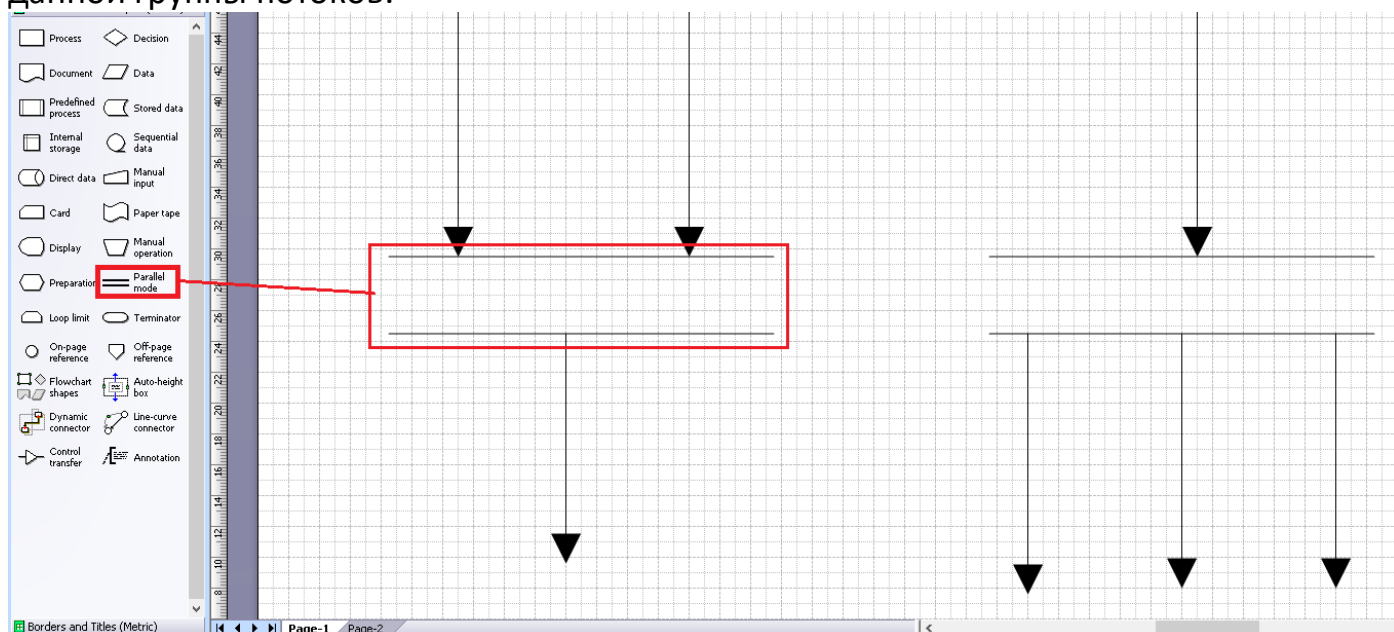
17. Линии и Линии с наконечником также являются элементом логических схем и изображают направленность выполнения выражений в программе (какие действия следуют за какими предыдущими действиями). Если линии сходятся в одну, то не должно быть визуального перекрестка – линии должны сходиться не друг в друга, а на некотором расстоянии друг от друга, хотя бы миллиметра 2 – 3. Линии должны идти только вертикально или горизонтально, могут иметь только углы 90 градусов на сгибах и не должны иметь закруглений, плавностей. Линия может быть ломаной, но следует минимизировать количество углов в ней, и эти сгибы могут быть только под углом 90 градусов.

Все линии лучше делать с наконечником, в этом случае не надо каждый раз вспоминать правило, согласно которому линии, идущие сверху вниз и слева направо могут не иметь наконечника, а линии, идущие снизу вверх и справа налево **обязательно** должны иметь наконечник.

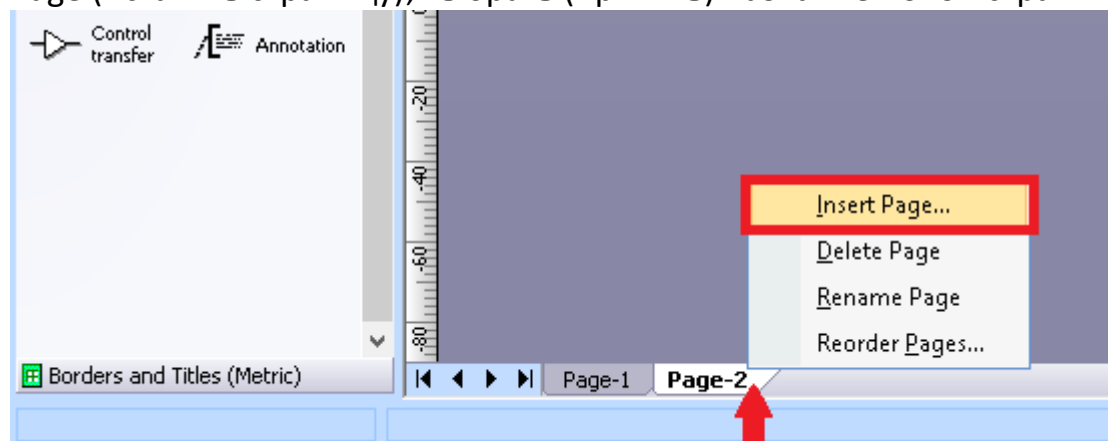
Линии в схемах должны подходить к фигуре либо слева, либо сверху, а выходить из фигуры либо справа, либо снизу. Линии должны быть направлены к центру фигуры, то есть присоединяться к Фигуре посередине любой ее стороны. Две линии не могут входить в фигуру, они должны объединиться заранее и войти в фигуру одной линией.



18. Для отображения синхронизации потоков в многопоточном алгоритме используется элемент Двух параллельных линий, означающие синхронизацию нескольких потоков или, наоборот, распараллеливание вычислений на несколько потоков. Синхронизация означает, что выполнение дальше не пойдет, пока все потоки не выполнятся. Распараллеливание должно завершаться синхронизацией всех потоков из данной группы потоков.



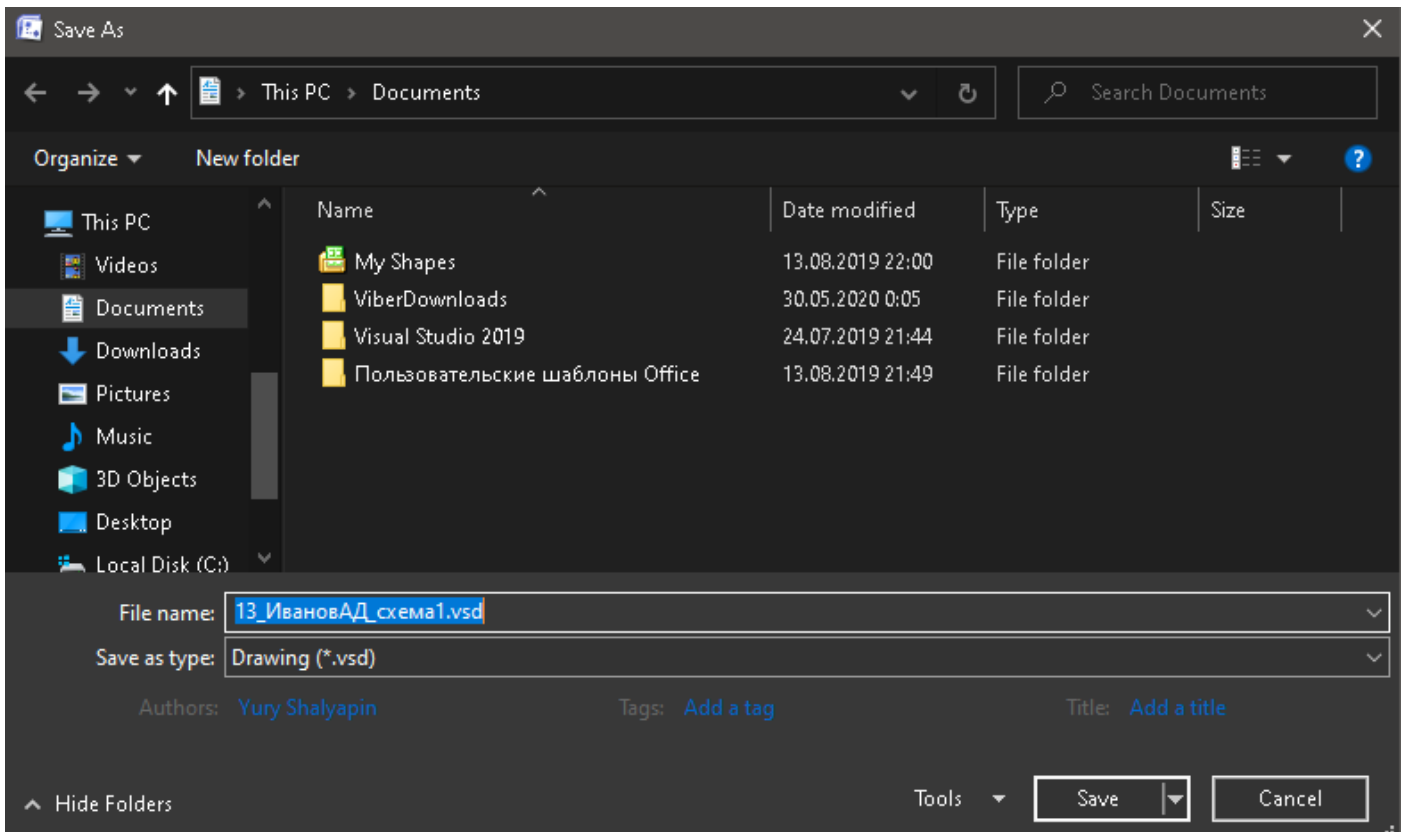
19. Чтобы добавить новую страницу в документ, надо нажать правой кнопкой мыши по вкладке страницы внизу слева и вызвать пункт контекстного меню Insert Page (Вставить страницу), выбрать (принять) название новой страницы и нажать ОК



20. Текст в элементах схемы должен быть понятным, кратким, однозначным. Шрифт и размер текста должны быть одинаковыми для всех элементов одной схемы, а поэтому, при необходимости изменения шрифта надо выделить все элементы схемы Ctrl+A и для всей схемы изменить шрифт (обычно размер шрифта уменьшают). Если текст не помещается внутри фигуры, следует дописать «непомещающуюся» его часть в элемент Комментарий, соединенный с данной фигурой пунктирной линией.

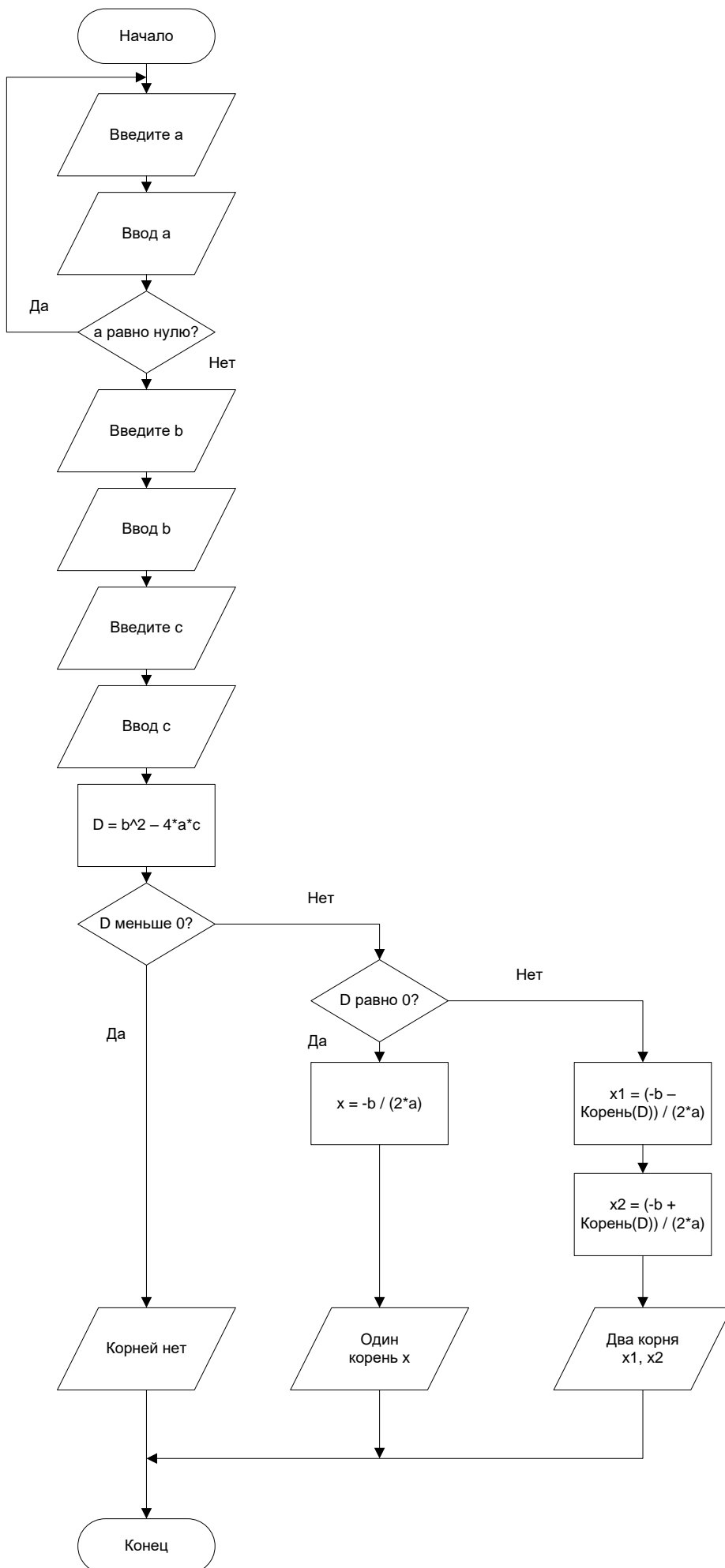
Все элементы из Набора элементов (Shapes) имеют изначальный стандартный размер и пропорциональны друг другу, поэтому нельзя отдельно изменять размер одного или нескольких элементов на схеме, менять форму или углы наклона линий элемента.

21. MS Visio позволяет сохранять вашу работу в формате *.vsd и потом ее снова открыть в программе MS Visio и продолжить «дорисовывать» диаграммы, редактировать ранее сделанные диаграммы. Формат *.vsd открывается в браузерах Microsoft Edge и Internet Explorer, но только в режиме просмотра, для внесения правок в логические схемы вам потребуется сама программа MS Visio.

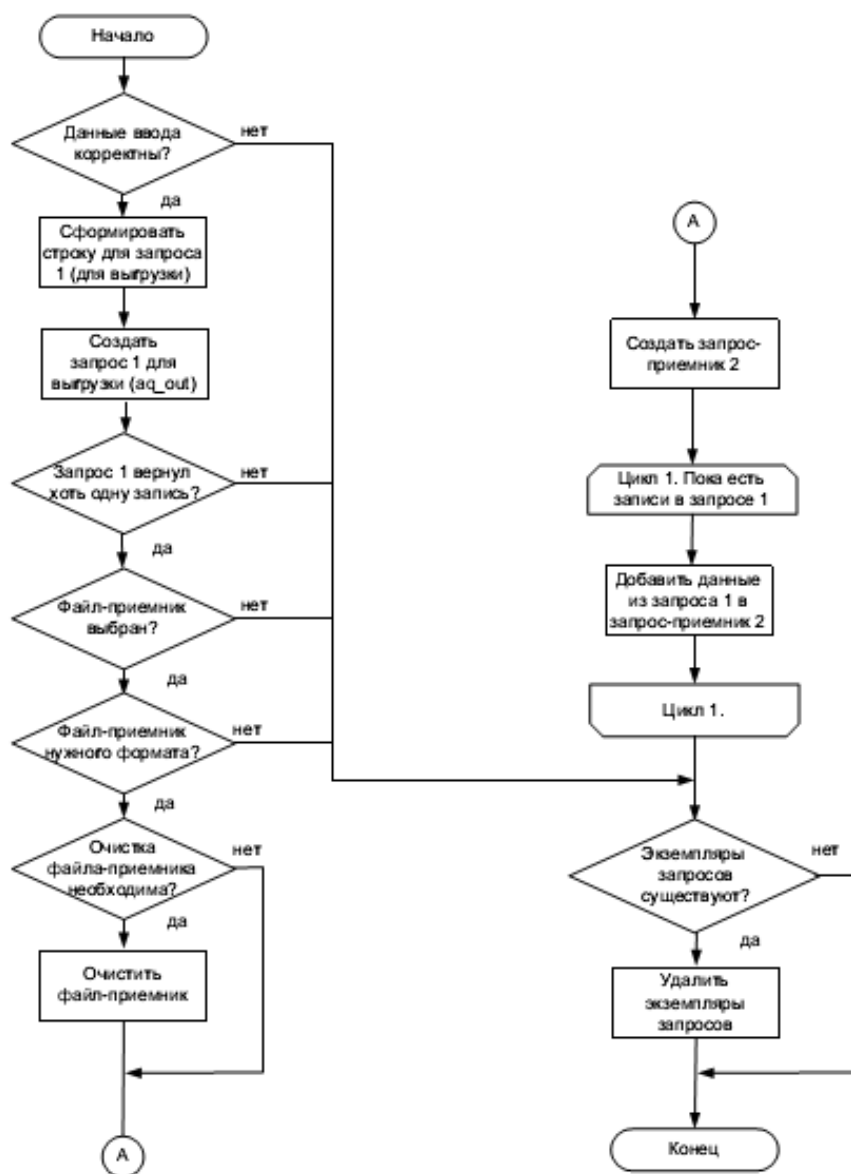


22. Меню switch – case отображается в логических схемах в виде ветвления (проверка Ромб) из которого в ЭТОМ случае выходит только одна стрелка и она входит в линию, из которой много выходов – по количеству кейсов. Каждый кейс должен быть назван или хотя бы пронумерован. Линия от Ромба **не** должна продолжиться визуально ни в одной линии ветвления, поскольку это нарушит «правило перекрестка» (см. п. 17).

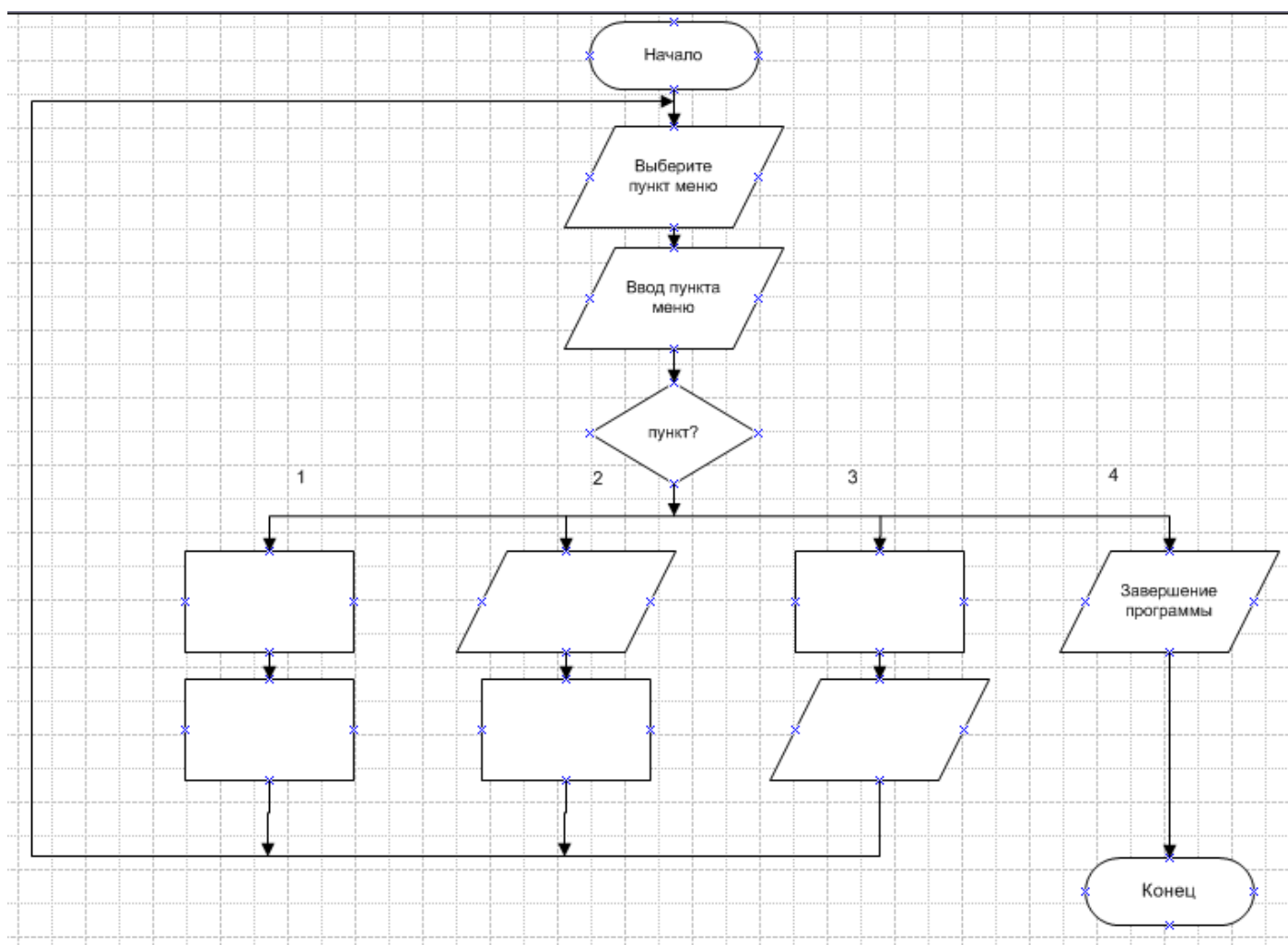
23. Ниже пример алгоритма нахождения корней уравнения $a \cdot x^2 + b \cdot x + c = 0$, где a не должно быть равным 0. Обратите внимание, что последнее третье ветвление Ромб располагается ниже второго ветвления, поскольку по времени выполнения на процессоре сначала выполнится второе ветвление, а только после него может настать черед выполнения третьего ветвления, но не раньше и не одновременно. В то же время внизу параллелограммы показа пользователю результатов работы программы располагаются в одной линии, хотя в случае выполнения программы, если корней не будет, то сразу снизу за вторым ветвлением Ромб будет отображаться результат, то есть этот параллелограмм «Корней нет» должен быть по одному уровню с Ромбом третьего ветвления. Аналогично, параллелограмм «Один корень x » должен располагаться на одном уровне с Прямоугольником вычисления x_2 , но для симметрии отображения результатов работы программы параллелограммы расположены на схеме в одну линию.



24. Пример логической схемы отправки запросов к базе данных и сохранения полученных ответов (возвращенных базой данных записей) в файл-приемник. Обратите внимание, что данная «длинная» схема разбита на два участка, расположенные на одной странице, и соединенные элементом Круг.



25. Ниже пример отображения меню switch – case в вертикальном расположении (есть еще горизонтальное). Обратите внимание, что все кейсы имеют обозначения (названия, номера) перед ветками.



Задание 2.

Сделать логическую схему на свою консольную программу, выполненную вами по индивидуальному варианту в одной из предыдущих лабораторных работ.

Задание 3.

Сделать логическую схему на свою оконную Windows Forms программу, выполненную вами по индивидуальному варианту в одной из предыдущих лабораторных работ.

Литература

Зиборов, В. MS Visual C++ 2010 в среде .NET (Библиотека программиста) / В. Зиборов. — СПб. : Питер , 2012.

Преподаватель

Белокопыцкая Ю.А.

Рассмотрено на заседании цикловой
комиссии ПОИТ №10

Протокол № ____ от « ____ » _____ 2018 года

Председатель ЦК _____