**Сценарий-конспект онлайн урока**

Выполнила студентка Московского Государственного Областного Университета, физико-математического факультета, по направлению: Педагогическое образование, Информатика Киричевская Юлия Владимировна

Тема урока: «Программирование разветвляющихся алгоритмов. Условный оператор».

Учебник: Л.Л. Босова Информатика 8 класс.

Цели урока:

* *образовательная:*
* формирование у учащихся представлений о разветвляющемся алгоритме и условном операторе; усвоение правильности описания этого вида ветвлений в блок-схеме и программе на языке Паскаль; формирование практических навыков по изменению программы, отладке и просмотру полученных результатов (связь с прошлыми уроками.
* *воспитательная:*
* выработка терпения и аккуратности при исправлении программы и, особенно, при поиске и исправлении ошибок); выработка последовательности действий по алгоритму.
* *развивающая:*
* развитие наблюдательности при работе с алгоритмом, записанном в виде блок-схемы и программы; осознание полученного результата.

Результаты обучения:

* *предметные:*
* умение записывать на языке программирования короткие алгоритмы, содержащие алгоритмическую конструкцию «ветвление»;
* *метапредметные:*
* умение самостоятельно планировать пути достижения целей;
* умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией;
* умение оценивать правильность выполнения учебной задачи;
* *личностные:*
* алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе;
* представление о программировании как сфере возможной профессиональной деятельности.

Решаемые учебные задачи:

* рассмотреть правила записи условного оператора;
* рассмотреть формат и назначение составного оператора;
* подвести учащихся к выводу, что для решения одной и той же задачи можно разработать разные алгоритмы;
* сформировать умение записывать на языке программирования короткие алгоритмы с ветвлениями.

Основные понятия, рассматриваемые на уроках:

* условный оператор;
* неполная форма условного оператора;
* составной оператор;
* вложенные ветвления.

Оборудование

* персональный компьютер (ПК) учителя, подключение к Интернету;
* ПК (смартфон) учащихся, подключение к Интернету;

Программное обеспечение: сервис для онлайн общения Discord, сервис разработки графических схем Lucidchart (данное приложение есть как приложение Google Диска, так и его можно скачать на телефон), сервис онлайн компилятор языка программирования Паскаль <https://www.onlinegdb.com/online_pascal_compiler>, также можно использовать приложение в смартфоне. Тип урока: онлайн урок

Подготовка к уроку:

1. Оповестить обучающихся о форме проведения урока. Разослать инструкции по скачиванию и установке Discord, настройке аккаунта. (Приложение 1);
2. Разослать инструкции по подключению и использованию сервиса Lucidchart. (Приложение 2);
3. Провести тестовую онлайн конференцию, проверить верить подключение, показать возможности стрима.

Ход урока

|  |  |  |
| --- | --- | --- |
| Этап урока | Деятельность учителя | Деятельность обучающихся |
| Организационный | Проверка подключения к серверу  Тестирование звука и камеры  Учитель приветствует обучающихся в голосовом (видео) чате.  Приветствие дублируется текстовом чате,  Помогает правильно включиться в работу.  Напоминает о подготовке к уроку учебника и тетради | Учащиеся приветствуют, в чат пишут свои фамилии  Учащиеся пишут в текстовом чате о качестве подключения сообщения: все хорошо или есть технические проблемы (ставят «+»).  Готовят учебник и тетрадь |
| Мотивация | У каждой программы два назначения: что она должна делать и чего не должна. Если согласны то пришлите + или позитивный смайлик | Ученики присылают отклик в чат |
| Актуализация | Слайд 1 коллаж из алгоритмических конструкций  Вопросы учителя  Какие виды алгоритмических конструкций мы изучили?  Какой вид алгоритмической конструкции мы уже умеем программировать, т. е. записывать на языке Паскаль? | Ответы пишут в чате, учащиеся также используют голосовой чат для ответов |
| Определение темы урока и плана | Слайд 2.  Учитель создает стрим в чате, и демонстрирует свой экран. Сначала показывается презентация.  Сегодня мы изучим, как записать на языке программирования Паскаль разветвляющийся алгоритм, т.е. алгоритм содержащий условие  Тема урока: «Программирование разветвляющихся алгоритмов. Условный оператор».  План работы:  - знакомство с синтаксисом условного оператора, запись в тетрадь (5 мин)  - решение задачи 1: запись условия, блок-схемы, ввод отладка и тестирование программы;  - решение задачи 2: запись условия, блок-схемы, ввод отладка и тестирование программы;  - подведение итогов урока;  - домашнее зание. | Отклик в чате о готовности к работе и хорошем обзоре презентации «+»  Записывают тему урока в тетрадь |
| Объяснение | Слайд 3  Слова *if-then-else* переводятся с английского языка на русский как *если-то-иначе*, что полностью соответствует записи ветвления на алгоритмическом языке.  **Перед *else*  знак «;» не ставится.**  Общий вид условного оператора.  Полная форма условного оператора:  *if* <условие> *then* <оператор\_1> *else*<оператор 2>  Неполная форма условного оператора:  *if* <условие> *then* <оператор>  Условия – логические выражения. Бывают простые (операции отношений) и сложные (логические операции). | Записывают в тетрадь  ***if***<условие> ***then***  <оператор\_1>  ***else***<оператор 2> ;  ***if***<условие> ***then***  <оператор\_1> ; |
| Инструктаж по оформлению результатов работы на уроке. | Слайд 4.  Отчетность о проделанной работе.  Отчет состоит из документа, в котором представлены 4 скрина: 2 скрина блок-схем и 2 скрина выполненных программ.  Этапы решения задач:  1. Записать условие задачи.  2. Составить блок-схему (в сервисе Lucidchart).  3. Записать программу на языке Паскаль. Отладить, запустить для тестовых данных | Принимают план, задают вопросы в голосовом чате, если возникают вопросы по отчетности |
| Решение задачи 1 | Слайд 5.  На слайде представлена задача и блок-схема к ней.  Пример 1. Запишите на языке Паскаль алгоритм определения принадлежности точки *x* ~~к~~ отрезку [*a,b*].  Для того чтобы решить задачу нужно определить.  Переменные: точка x, концы отрезка a, b. Тип переменных – рациональные числа, Real  Чтобы определить принадлежит ли точка отрезку, нужно использовать условную развилку, в котором проверяем условие *a<x<b*. Двойное неравенство запишем с помощью конъюнкции (логического умножения ) – операнда *and* *(a<x) and (x<b).* Если условие выполняется, то на экран выводится *ДА*, если условие не выполнено, то *НЕТ*.  Учитель после того, как разобрал блок-схему на слайде, присылает скриншот блок-схемы в специальный чат, чтобы учащимся было удобнее выполнять задание.  Учитель переходит в сервис Lucidchart и на экране демонстрирует данный сервис и процесс создания блок-схемы. Также учитель напоминает, что для инициализации учащихся они должны добавить специальный блок, в катором укажут свою фамилию. | Учащиеся во время объяснения блок-схемы отвечают на вопросы по выполнению блок-схемы, также задают свои вопросы.  Данную блок-схему учащиеся должны выполнить в Lucidchart. Учащиеся параллельно объяснениям выполняют блок-схему и после ее выполнения присылают скриншот экрана о выполненной работе. |
|  | Слайд 6.  *Учитель комментирует программу. Привлекает учащихся для комментариев известных им операторов.*  На слайде приведена программа для решения задачи.  С самого начала программы записывается слово program и после него название программы, после обязательно «;».  Раздел переменных **var**.Так как координаты точки и концов могут быть представлены в виде нецелого числа, будем использовать вещеcтвенный тип переменной – real.  Тело программы записывается между **begin** (после не ставится «;») и **end.** В теле программы:   1. Вывести на экран условие задачи, посредством оператора **writeln** (напомнить, что пишется в одинарных кавычках); 2. Вывести на экран, что требуется ввести концы отрезков, посредством оператора **write** (напомнить, что пишется в одинарных кавычках); 3. Ввести в память значения концов отрезков, посредством оператора **readln**; 4. Вывести на экран, что требуется ввести координату точки, посредством оператора **write** (напомнить, что пишется в одинарных кавычках); 5. Ввести в память значение координаты точки, посредством оператора **readln**; 6. Если координата точки, больше начальной точки отрезка и меньше конечной точки отрезка (используя условны оператор), то вывести на экран сведения о принадлежности точки. Если наоборот, то точка не принадлежит отрезку. Условие «ДА» записывается после **then**, условие «НЕТ» записывается после **else:**; 7. Вывод результата, посредством оператора **writeln**.   После разбора программы учитель скидывает готовую программу в специальный чат и переходит в онлайн-компилятор, который также размещает в чат. Учитель напоминает, что для свое инициализации учащиеся вместо n\_9 пишут свою фамилию. | Учащиеся участвуют в голосовом чате в комментировании программы.  Учащиеся переходят в онлайн-компилятор и копируют программу, изменяя ее. После того, как учащиеся сделали программу, они присылают скриншот в чат. |
|  | Слайд 7  Тестирование работы программы:  Запишите в тетради данные и ответ   |  |  |  |  | | --- | --- | --- | --- | | Ввод значений переменных | | | Ответ программы | | a | b | x | | 0 | 20 | 10 |  | | -5 | 5 | 10 |  | | Ученики проводят тестирование работы программы, записывают результаты в таблицу. После этого присылают ответ в чат в виде фото/текста. |
| Решение задачи 2. | Слайд 8  Пример неполного условного оператора. На примере задачи: присвоить переменной *y* значений наибольшей из трех переменных *a, b, c*.  Для сравнения трех величин нужно ввести переменную y. Затем присвоить данной переменной первое число. В блоке условие сравнить переменную у со вторым числом, если второе число больше у, присвоить у значение второго числа и перейти к следующему сравнению, если нет, то перейти к следующему сравнению. В блоке условие сравнить переменную у с третьим числом, если третье число больше у, присвоить у значение третьего числа, если нет, то у остается неизменным. Затем вывести у – это и будет наибольшая величина.  Учитель после того, как разобрал блок-схему на слайде, присылает скриншот блок-схемы в специальный чат, чтобы учащимся было удобнее выполнять задание.  Учитель переходит в сервис Lucidchart и на экране демонстрирует данный сервис и процесс создания блок-схемы. Также учитель напоминает, что для инициализации учащихся они должны добавить специальный блок, в катором укажут свою фамилию. | Учащиеся во время объяснения блок-схемы отвечают на вопросы по выполнению блок-схемы, также задают свои вопросы.  Данную блок-схему учащиеся должны выполнить в Lucidchart. Учащиеся параллельно объяснениям выполняют блок-схему и после ее выполнения присылают скриншот экрана о выполненной работе. |
|  | Слайд 9.  *Учитель комментирует программу. Привлекает учащихся для комментариев известных им операторов.*  На слайде приведена программа для решения задачи.  С самого начала программы записывается слово program и после него название программы, после обязательно «;».  Раздел переменных **var**. В данной задаче будут использоваться цедые числа, поэтому используется целый тип переменной – integer.  Тело программы записывается между **begin** (после не ставится «;») и **end.** В теле программы:   1. Вывести на экран условие задачи, посредством оператора **writeln** (напомнить, что пишется в одинарных кавычках); 2. Вывести на экран, что требуется ввести три числа, посредством оператора **write** (напомнить, что пишется в одинарных кавычках); 3. Ввести в память значения чисел, посредством оператора **readln**; 4. Так как переменной y присваивается большее значение, сначала переменной приравнивается первое число (**:=**); 5. Затем происходит сравнение переменной y и второго числа (**if**). Если второе число больше, то переменной y присваивается (**:=**) значении второго числа и переходят к следующему этапу, если нет то программа переходит к следующему сравнению. 6. Второе сравнение: переменную у сравнивают с третьим числом, если оно больше, то его значение присваивают (**:=**) у, если нет, переходят к следующему этапу; 7. После всех сравнений выводят переменную у; 8. Вывод результата, посредством оператора **writeln**.   После разбора программы учитель скидывает готовую программу в специальный чат и переходит в онлайн-компилятор, который также размещает в чат. Учитель напоминает, что для свое инициализации учащиеся вместо n\_9 пишут свою фамилию. | Учащиеся участвуют в голосовом чате в комментировании программы  Учащиеся переходят в онлайн-компилятор и копируют программу, изменяя ее. После того, как учащиеся сделали программу, они присылают скриншот в чат. |
|  | Слайд 10.  Тестирование работы программы:  Запишите в тетради данные и ответ   |  |  |  |  | | --- | --- | --- | --- | | Ввод значений переменных | | | Ответ программы | | A | b | с | | 2 | -2 | 0 |  | | 20 | -3 | -1 |  | | Ученики проводят тестирование работы программы, записывают результаты в таблицу. После присылают результаты в Discord (ответ, фото, скрин) |
|  | Слайд 11.  Спасибо за внимание! | Поочередно каждый учащийся говорит в голосовом чате фамилию и выходит из стрима |

Домашнее задание представлено в виде теста. Тест также присылается в чат Dicscord.

<https://forms.gle/ZRYzVJUo6sh34Bfj6>

ПРИЛОЖЕНИЕ 1

Для регистрации в Discord нужно зайти на сайт <https://discordapp.com/> . Затем нажать на Открыть Discord (рис. 1).

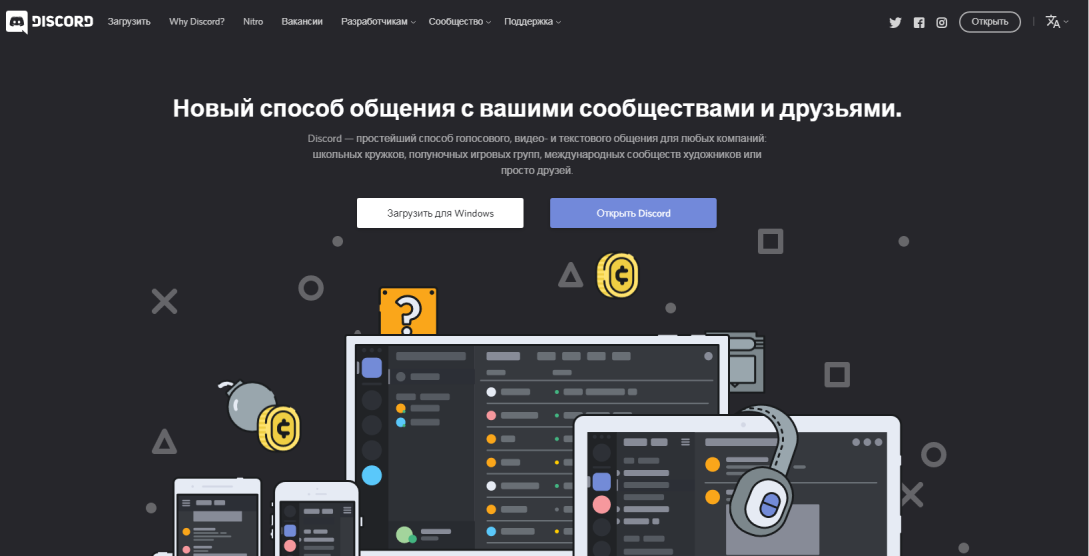


Рис.1. Главная страница Discord

Нажмите на Зарегистрироваться, после заполните все необходимые поля. Нажмите Продолжить (рис. 2).

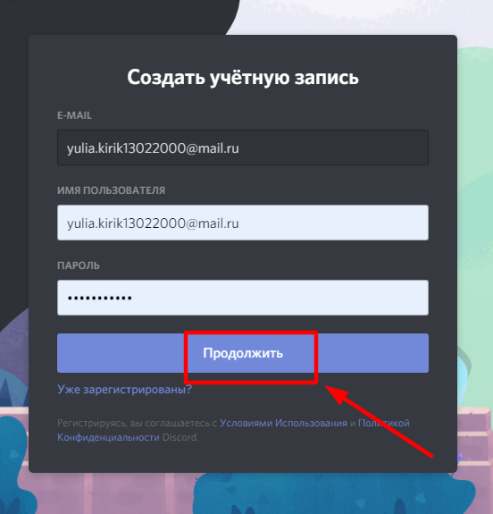
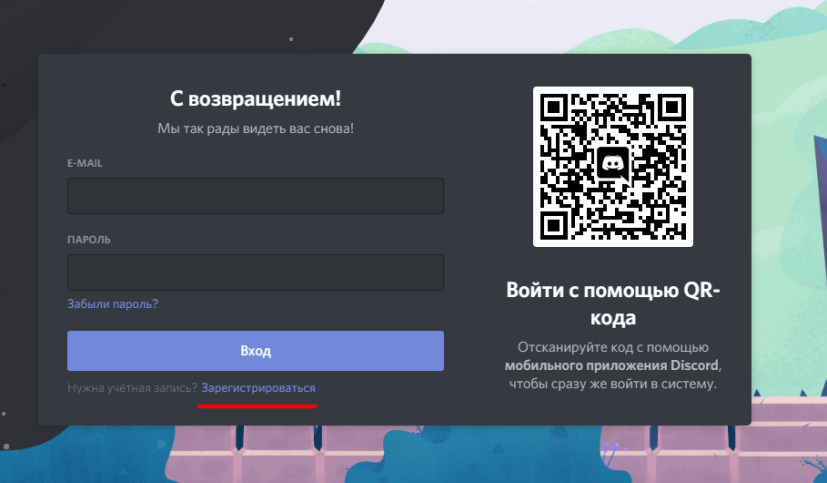


Рис. 2. Регистрация в Discord

ПРИЛОЖЕНИЕ 2

Методические рекомендации по выполнению практической работы

Для того, чтобы подключить сервис Lucidchart, вам нужно зайти в свой Google Диск, нажать на кнопку Создать, затем Еще и выберите пункт Подключить другие приложения (рис. 3).

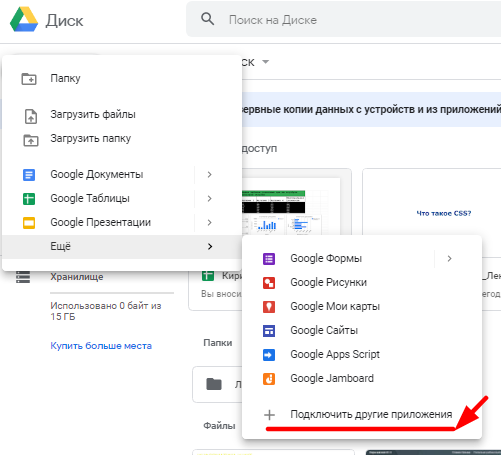
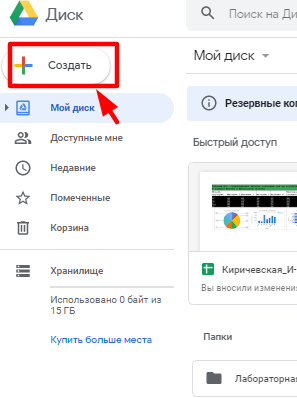


Рис. 3. Рекомендации к подключению сервиса часть 1

Найдите Lucidchart Diagrams, выберите данный сервис и нажмите Установить/Подключить (рис. 4).

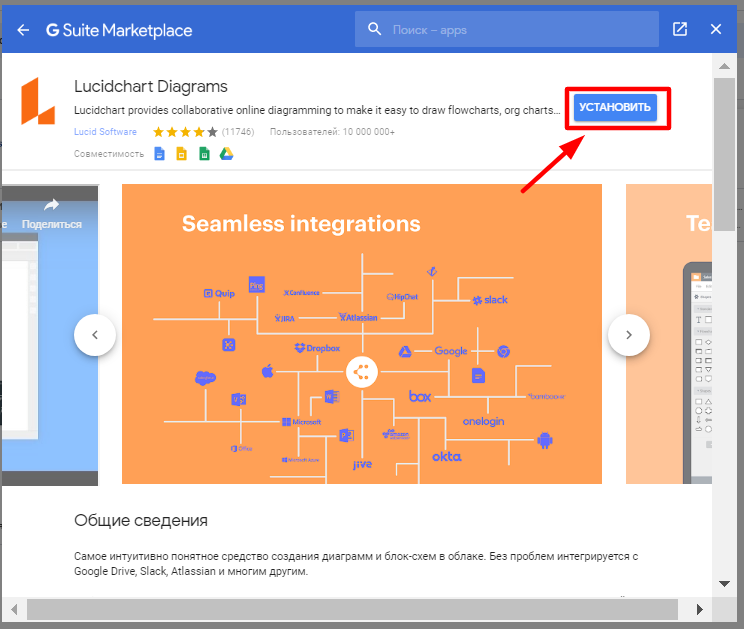


Рис. 4. Рекомендации к подключению сервиса часть 2

В списке шаблонов нужно выбрать Flowcharts. Затем выбрать Blank Diagram (рис. 5). Перед вами окажется пустой лист.

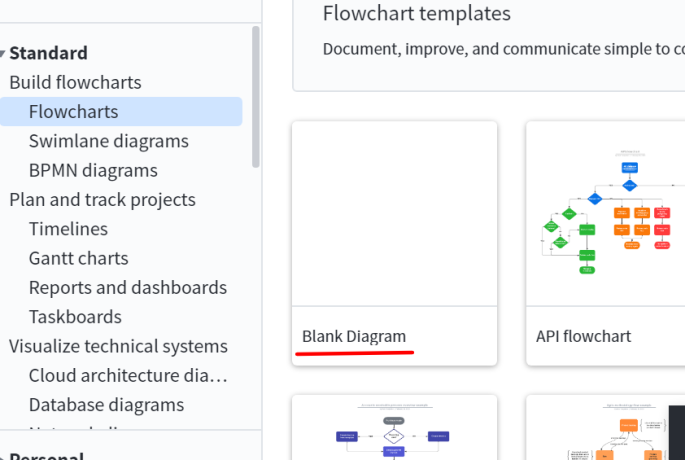
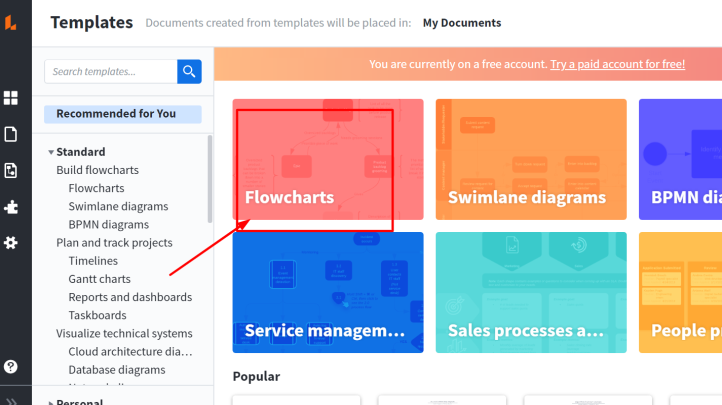


Рис. 5. Создание пустого документа

Чтобы изменить название данного документа в левом верхнем углу нажмите на Пустая диаграмма и измените название (рис. 6).

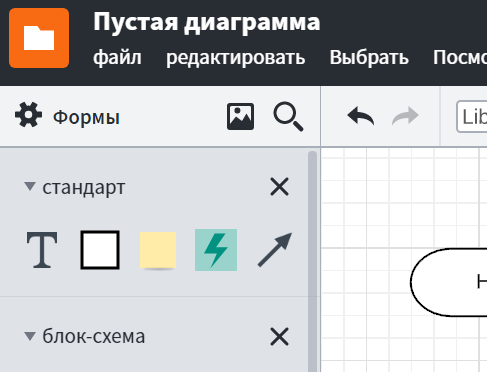


Рис. 6. Изменение названия документа

Для того чтобы создать блок-схему, нужно из левой колонки перетащить нужные блоки. Блоки можно изменять по ширине, длине, записывать в них текст (двойное нажатие правой кнопкой мыши по блоку). Затем соединить их стрелками (рис. 7).

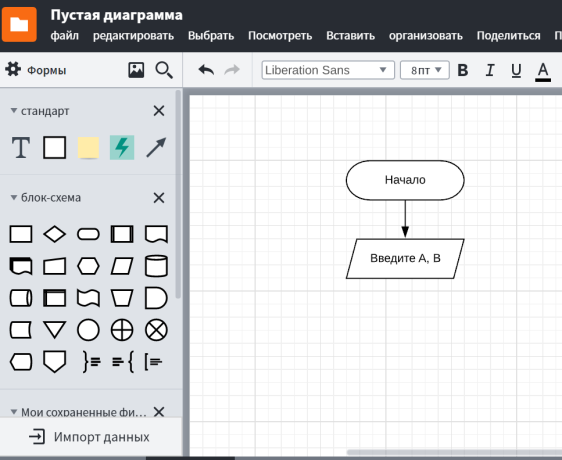


Рис. 7. Пример части блок-схемы

Для удобства выравнивания блок-схемы при перемещении блоков представлены значения расстояний между ними. Также сохранение данного документа происходит автоматически, поэтому после завершения работы, просто закройте документ.

Онлайн компилятор <https://www.onlinegdb.com/online_pascal_compiler>

На данном сайте вводится программа как в обычном Паскале. Для выполнения нажать на RUN. Данные вводятся также, как и в Паскале на компьютере.

Если у учащихся нет возможности работать за компьютером, то они могут воспользоваться онлайн-компилятором языка Паскаль в своем смартфоне, также приложение Lucidchart представлено в маркете приложений.