

Занятие 3.

1. Линейные алгоритмы.

ЛИНЕЙНЫЙ АЛГОРИТМ – алгоритм, в котором команды (действия) выполняются друг за другом.

Пример линейного алгоритма: «Рецепт выпечки торта», «Расписание уроков».

А вот «План действий» или «План решения задачи» может быть зависимым от каких – то условий. Но об этом мы узнаем в следующей теме.

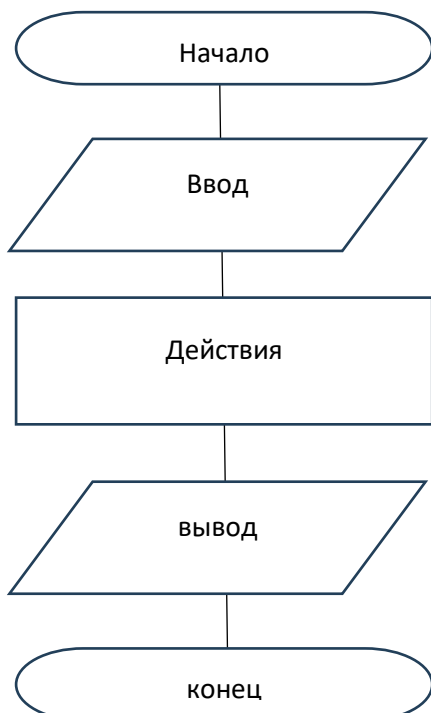
Познакомимся с понятием блок – схемы. БЛОК – СХЕМА – это способ описания алгоритма. Для ее составления используют блоки.

	Блок начала / конца алгоритма
	Блок ввода / вывода
	Блок выполнения команды (действия)

Существуют правила составления блок-схем. Начнем с главных

1. Блоки должны быть одинаковыми по ширине
2. Блоки нанизываются сверху вниз. Линия на которую они нанизываются называется «царская дорога», она показывает направление чтения алгоритма. Для указания направления можно указывать стрелочку, направление сверху – вниз считается основным, стрелочку можно не ставить.

Для линейных алгоритмов чаще всего блок- схема выглядит так:



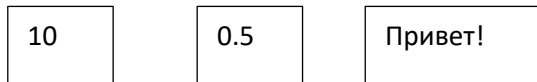
2. Виды величин.

Константы – величины, которые хранятся в оперативной памяти, но не могут изменяться во время выполнения программы. Используются в операторах присваивания, вывода и пр.

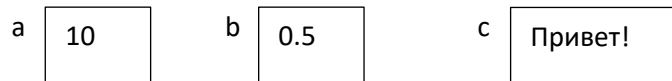
a = 10 10 – числовая константа целого типа

b = 0.5 0.5 – числовая константа вещественного типа

c = 'Привет!' 'Привет' – строковая константа



Переменные – ячейки оперативной памяти, которые имеют имя (адрес на ячейку), значение и тип. Переменные могут менять свое значение в процессе выполнения программы. Тип переменных определяется по значению



a = 10 a – переменная числовая целого типа, ее значение 10, имя a.

b = 0.5 b – переменная числовая вещественного типа, ее значение 0.5, имя b.

c = 'Привет!' c – переменная строковая, ее значение 'привет!', имя c.

3. Типы величин. Типы величин определяются по их значениям автоматически!

Str – строки ('Привет!')

Int – целые числа (5, -10)

Float – вещественные числа (0.5, -0.001)

Bool – логический тип (True - истина, False - ложь)

Оператор type() – определяет тип величин
print(type(a))

На языке Питон допустимо в программе использовать одну и ту же переменную смешивая типы значений. **Но так лучше не делать!!!!**

```
a = 10
print (a)           # 10
print (type(a))     # int
a= 'Привет!'
print(a)            # Привет!
print (type(a))     # str
```

4. Оператор ввода

Для того чтобы ввести данные с клавиатуры и присвоить эти значения переменным используется оператор **input()**

Он используется в других операторах, например в присваивании:

a = input()

В скобочках можно добавлять комментарии, которые будут выводиться перед вводом.

Например: a = input('Введите сторону квадрата')

При выполнении данной команды python останавливает выполнение программы и ждет ввода данных и нажатия клавиши Enter. Сигналом необходимости ввода является отсутствия знака готовности ввода команд (>>>)

ВНИМАНИЕ! Input() вводит с клавиатуры СТОРКУ! А значит действия, которые могут быть выполнены – это действия со строками. Если необходимо вводить числа,

то нужно делать преобразования строки в число. Например `a = int(input())` – введет строку '5' и конвертирует в целое число 5.

Например:

<code>a = input('введите значение')</code>	(str)	#5	
<code>b = input('Введите значение')</code>	(str)	#10	
<code>s = a + b</code>	(str)	#510	произошла операция склеивания строк
<code>print(s)</code>		510	
<code>print(type(s))</code>		str	

В данной задаче все переменные строковые (str)

Для того, чтобы выполнялось арифметическое сложение нужно сделать преобразование строк в числа (функции **int** или **float**).

```
a = int(input('введите число')) (int)
a = float(input('введите число')) (float)
```

<code>a = int(input('введите значение'))</code>	(int)	#5	
<code>b = int(input('Введите значение'))</code>	(int)	#10	
<code>s = a + b</code>	(int)	#15	сложение
<code>print(s)</code>		15	
<code>print(type(s))</code>		int	

Ввод данных осуществляется с клавиатуры (или других устройств ввода) в оперативную память.

Вывод констант и значений переменных осуществляется из оперативной памяти на монитор.

Задача 1. Необходимо организовать диалог между программой и пользователем:

(программа) - Привет, как тебя зовут?

(пользователь) - Петя

(программа) – Здорово, Петя. А сколько тебе лет?

(пользователь) - 12

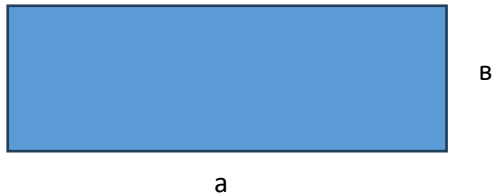
(программа) - Петя, ты молодец, в возрасте 12 лет настало время учиться программировать!

Программа выглядит приблизительно так:

```
name = input('Привет, как тебя зовут?')
print('Здорово,', name)
age = input('А сколько тебе лет?')
print(name, ' ты молодец, в возрасте ', age, ' лет настало время учиться
программировать')
...
```

Задача 2. Кот Матроскин решил посадить на своем огороде картошку. Ему нужно его огородить забором и купить картошку для посадки. Сколько нужно Матроскину денег, чтобы реализовать свою мечту, если огород прямоугольной формы длиной *a* и шириной *b*, стоимость одного метра забора 200р, для посадки картошки нужно учесть, что на 1 м.кв нужно 0.5 кг, а 1 кг картошки стоит 70р.

Рассмотрим эту задачу математически (математическая постановка):



Чтобы найти длину всего забора, нужно знать периметр огорода (прямоугольника):

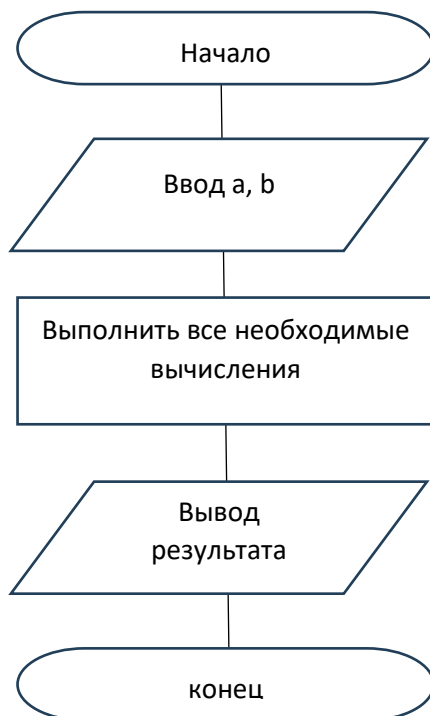
$$P = 2 * (a + b)$$

Денег на покупку забора нужно $P * 200$.

Чтобы посчитать сколько нужно картошки, нужно знать площадь огорода $S = a * b$

Тогда картошки нужно купить $S * 0.5$, а заплатить $S * 0.5 * 70$ (р). Для окончательного результата нужно сложить деньги, необходимые для покупки забора и картошки.

Составим блок – схему:



```
a = int(input('Введите 1 сторону огорода'))
```

int понадобился для
преобразования строки в число

```
b = int(input('Введите 2 сторону огорода'))
```

```
p = 2 * (a + b)
```

```
d1 = p * 200
```

```
s = a * b
```

```
d2 = s * 0.5 * 70
```

```
rezult = d1 + d2
```

```
print('Матроскину понадобится ', rezult, 'руб')
```

Домашнее задание.

1. Продолжить программу «Диалог»
2. Изменить программу «Ваня», так чтобы данные вводились с клавиатуры
3. Написать программу, которая вычисляет периметр и площадь квадрата.