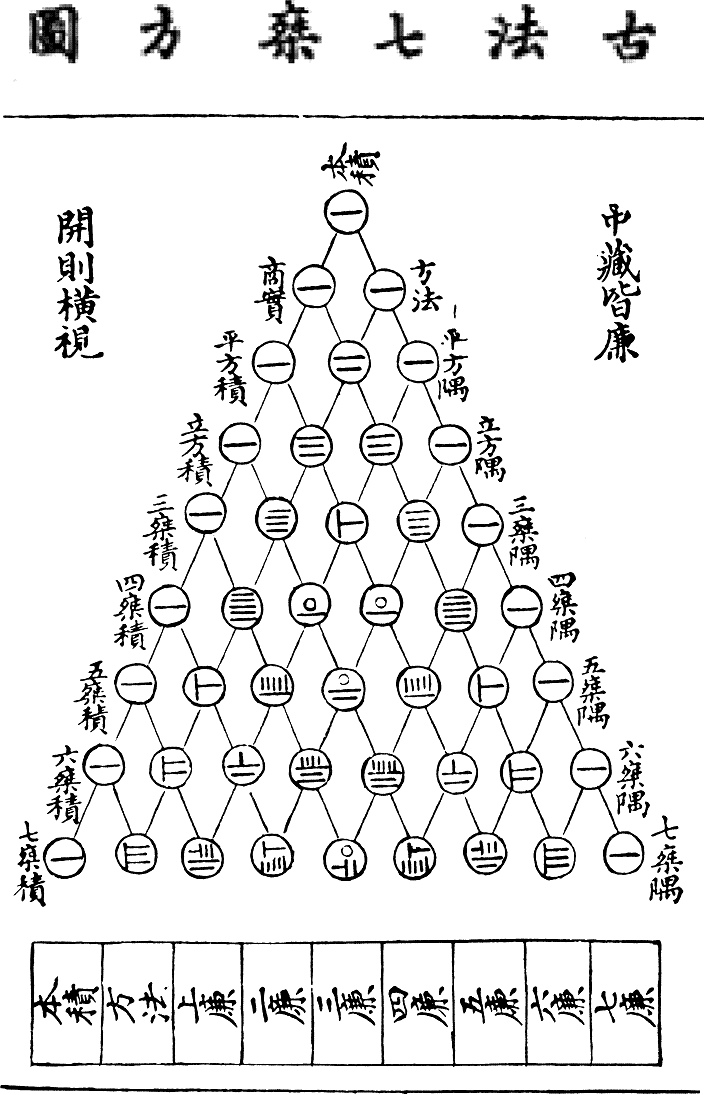
**История вычислительной техники и численные задачи**

Сегодня мы поговорим об истории вычислительной техники и о тех задачах, для которых потребовалось создавать вычислительные машины.

*Если материал кажется скучным или простым, можно промотать ниже — там начинается чистое программирование.*

**Развитие науки до эпохи счетных машин**



[Треугольник Паскаля](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA_%D0%9F%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D1%8F). Китай, 1303 год. Показан счет палочками.

Простейшим приспособлением для счета являются [Счётные палочки](https://ru.wikipedia.org/wiki/%D0%A1%D1%87%D1%91%D1%82%D0%BD%D1%8B%D0%B5_%D0%BF%D0%B0%D0%BB%D0%BE%D1%87%D0%BA%D0%B8).Из практики их использования родился символьный язык математики, который используется в настоящее время.

Для арифметических расчетов до недавнего времени массово использовались различные механические устройства. **Еще в 90-х годах XX века даже в магазинах Москвы можно было встретить счёты.** Но они быстро пропали благодаря удешевлению калькуляторов.



Картина [Б.М. Кустодиева](https://ru.wikipedia.org/wiki/%D0%9A%D1%83%D1%81%D1%82%D0%BE%D0%B4%D0%B8%D0%B5%D0%B2,_%D0%91%D0%BE%D1%80%D0%B8%D1%81_%D0%9C%D0%B8%D1%85%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B8%D1%87) «Купец»,1918 год. На картине — [русские счёты](https://ru.wikipedia.org/wiki/%D0%A1%D1%87%D1%91%D1%82%D1%8B).

Сегодня сложно представить науку без сложных вычислений, выполняемых на компьютерах. Однако еще задолго до появления электронных вычислительных машин (ЭВМ) были созданы достаточно сложные механические устройства.

**История счетных машин**

Для умножения и деления многозначных чисел были созданы логарифмические линейки и механические арифмометры. Изобретение и использование таких устройств относится к времени научной революции начала XVII века. Тогда вычислительные устройства благодаря точным расчетам помогли сделать важнейшие открытия в астрономии. Так, [Иоганн Кеплер](https://ru.wikipedia.org/wiki/%D0%9A%D0%B5%D0%BF%D0%BB%D0%B5%D1%80,_%D0%98%D0%BE%D0%B3%D0%B0%D0%BD%D0%BD) сформулировал законы движения планет Солнечной системы, носящие его имя.

**Блез Паскаль и его суммирующая машина**

К середине столетия французским математиком Блезом Паскалем была изобретена [суммирующая машина](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%BC%D0%BC%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%9F%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D1%8F). Машина Паскаля представляла собой механическое устройство в виде ящичка с многочисленными связанными одна с другой шестерёнками. Складываемые числа вводились в машину при помощи соответствующего поворота наборных колёсиков. На каждое из этих колёсиков, соответствовавших одному десятичному разряду числа, были нанесены деления от 0 до 9. При вводе числа колесики прокручивались до соответствующей [ц](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%B0)ифры. Совершив полный оборот, избыток над цифрой 9 колёсико переносило на соседний разряд, сдвигая соседнее колесо на 1 позицию.

Из предыдущего рассказа мы знаем, что компьютеры не используют десятичную систему счисления из-за ее неэффективности и сложности реализации. **Но именно с десятичной системы, используемой человеком, и началась история счетных машин.**

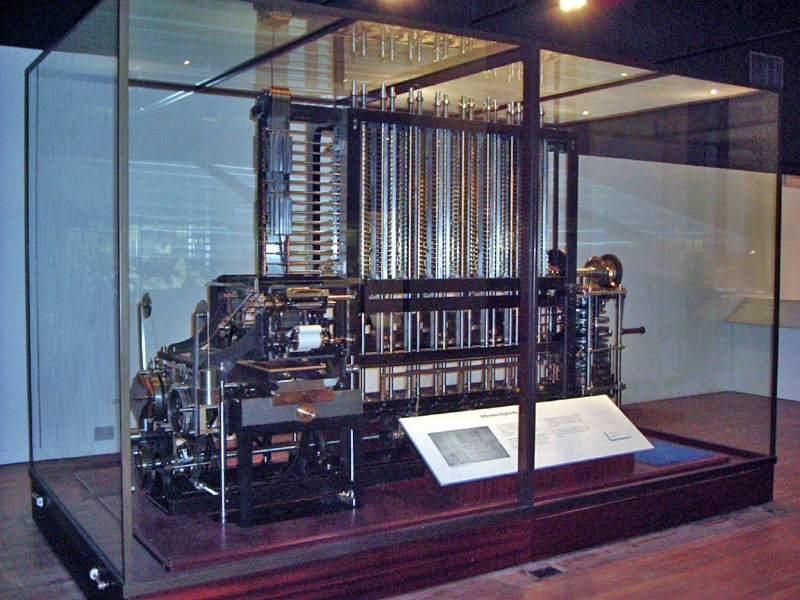
**Ткацкий станок Жаккара**



Перфокарты ткацкого станка Жаккара

В 1804 году [Жозеф Мари Жаккар](https://ru.wikipedia.org/wiki/%D0%96%D0%B0%D0%BA%D0%BA%D0%B0%D1%80,_%D0%96%D0%BE%D0%B7%D0%B5%D1%84_%D0%9C%D0%B0%D1%80%D0%B8) разработал ткацкий станок, в котором вышиваемый узор определялся [перфокартами](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%84%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0). Серия карт могла быть заменена, и смена узора не требовала изменений в механике станка. Так появился один из первых в истории информатики [носитель информации](https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%81%D0%B8%D1%82%D0%B5%D0%BB%D1%8C_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8). **И он был двоичным**: информация кодировалась последовательностями нулей (нет отверстия) и единиц (отверстие в карте).

**Чарльз Бэббидж**



Первая полностью построенная разностная машина в [лондонском Музее науки](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%B7%D0%B5%D0%B9_%D0%BD%D0%B0%D1%83%D0%BA%D0%B8_(%D0%9B%D0%BE%D0%BD%D0%B4%D0%BE%D0%BD))

В XIX веке Чарльзом Бэббиджем была изобретена **разностная машина**. Ее идея состояла в том, что вычисления можно производить с конечной точностью, а не абсолютно точно. Благодаря этому, машина становилась универсальным вычислительным прибором, использующим достижения математики того времени. Конечно, машина работала с числами в десятичной системе счисления.

В ходе работы у Бэббиджа возникла идея создания универсальной вычислительной машины, которую он назвал [**аналитической**](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%A7%D0%B0%D1%80%D0%BB%D1%8C%D0%B7%D0%B0_%D0%91%D1%8D%D0%B1%D0%B1%D0%B8%D0%B4%D0%B6%D0%B0)**.**Она стала прообразом современного [цифрового компьютера](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80).

В единую логическую схему Бэббидж увязал **арифметическое устройство**(названное им «мельницей»), **регистры памяти**, объединённые в единое целое («склад»), и **устройство ввода-вывода**, реализованное с помощью перфокарт трёх типов.

**Перфокарты операций**переключали машину между режимами сложения, вычитания, деления и умножения.

**Перфокарты переменных** управляли передачей данных из памяти в арифметическое устройство и обратно.

**Числовые перфокарты**могли быть использованы как для ввода данных в машину, так и для сохранения результатов вычислений, если памяти было недостаточно.

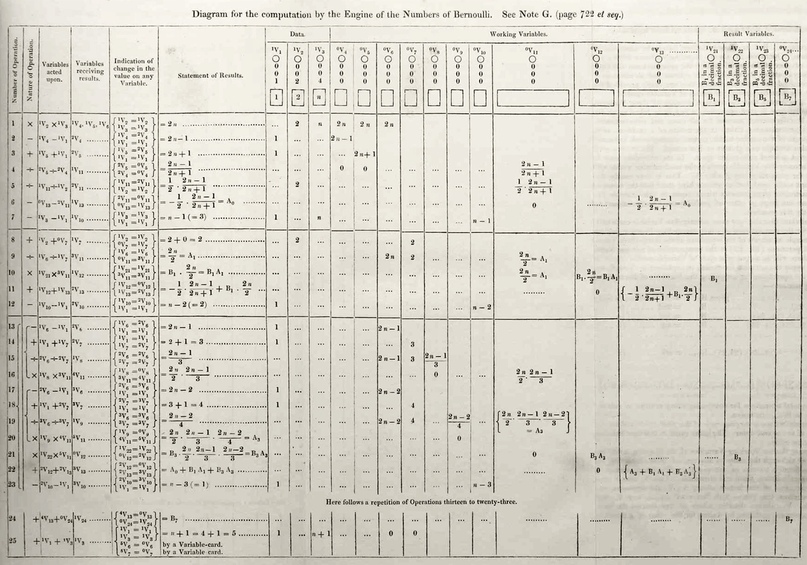
Это очень похоже на архитектуру современной вычислительной машины! О ней мы говорили в прошлый раз. Для создания компьютера в современном понимании оставалось лишь придумать **схему с хранимой программой**, что было сделано 100 лет спустя [Эккертом](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D0%BA%D0%B5%D1%80%D1%82,_%D0%94%D0%B6%D0%BE%D0%BD_%D0%9F%D1%80%D0%B5%D1%81%D0%BF%D0%B5%D1%80), [Мочли](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%87%D0%BB%D0%B8) и [Фон Нейманом](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D0%BD_%D0%9D%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD). В честь последнего названа [архитектура](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%84%D0%BE%D0%BD_%D0%9D%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD%D0%B0) почти всех современных компьютеров.

**Ада Лавлейс**



Бэббидж разрабатывал конструкцию аналитической машины в одиночку. Он часто посещал промышленные выставки, где были представлены различные новинки науки и техники. Именно там состоялось его знакомство с [Адой Августой Лавлейс](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B2%D0%BB%D0%B5%D0%B9%D1%81,_%D0%90%D0%B4%D0%B0) (дочерью Джорджа Байрона), которая стала его очень близким другом, помощником и единственным единомышленником.

В 1840 году Бэббидж ездил по приглашению итальянских математиков в Турин, где читал лекции о своей машине. [Луиджи Менабреа](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D0%B0%D0%B1%D1%80%D0%B5%D0%B0,_%D0%9B%D1%83%D0%B8%D0%B4%D0%B6%D0%B8_%D0%A4%D0%B5%D0%B4%D0%B5%D1%80%D0%B8%D0%BA%D0%BE) создал и опубликовал конспект лекций на французском языке. Позже Ада Лавлейс перевела эти лекции на английский язык, дополнив их комментариями, по объёму превосходящими исходный текст. Он был опубликован в сентябре 1843 года. В комментариях Ада сделала описание вычислительной машины и инструкции по программированию к ней. **Это были первые в мире программы. Именно поэтому Аду Лавлейс называют первым программистом.** Однако, аналитическая машина так и не была закончена.



Создать карусель

Алгоритм вычисления [чисел Бернулли](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%91%D0%B5%D1%80%D0%BD%D1%83%D0%BB%D0%BB%D0%B8) — первая в мире программа, написанная для вычислительной машины.

**Различные численные задачи**

[Алгоритмы](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) для решения [численных задач](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B) были известны с античности. [Компьютерная программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) отличается от алгоритма тем, что она:

1) написана на языке программирования,

2) предназначена для выполнения на вычислительной машине,

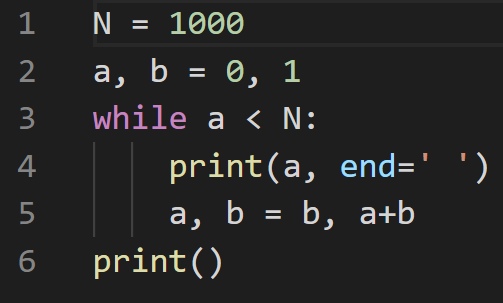
3) может содержать как инструкции, так и данные.

На главной странице [сайта языка программирования Python](https://www.python.org/) показан пример реализации алгоритма — вычисление [чисел Фибоначчи](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8).

На этом примере разберем отличия алгоритма и программы.

Числа Фибоначчи — это элементы числовой последовательности, в которой первые два числа равны либо 1 и 1, либо 0 и 1, а каждое последующее число равно сумме двух предыдущих чисел.

Значит, для вычисления следующего числа из двух предыдущих надо 1) выполнить операцию сложения и 2) запомнить результат, после чего можно приступать в вычислению следующего значения.



Программа на языке Python будет состоять из нескольких операций.

Сначала идут **операции присваивания**, они обозначены знаком «=».

Для работы программы нам понадобятся [**Константы**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D0%B0%D0%BD%D1%82%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))и[**переменные**](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))**.**Это, прежде всего, имя, которому присваивается значение. В отличие от переменной, изменение константы в процессе выполнения программы не предполагается или запрещается.

Строки 1 и 2 — **три**операции присваивания:

1. переменной с именем n присваивается значение 1000. Изменение этой переменной не предполагается, поэтому мы ее можем назвать константой.

*— Как тогда отличить константу от переменной в программе на Python?*

Для этого существует договоренность: названия констант пишутся ЗАГЛАВНЫМИ буквами. Но эсли этого не сделать, ничего не случится, программа будет работать, как и прежде.

2. Переменным a и b присваиваются значения 0 и 1 соответственно. То есть, **в одной строке можно выполнять несколько операций**. Названия переменных и их значения тогда перечисляются через запятую.

3. Далее следует определение цикла **while**. Оно состоит из ключевого слова while (по-русски *пока*) и условия, которое является **булевой, или бинарной операцией**. Об этом говорили в прошлый раз. Мы как бы говорим: «Пока выполняется условие цикла, выполняй команды в следующих строках, отмеченных отступами в 4 пробела». При **проверке условия на истинность**выполняется операция сравнения: если a меньше N, ответ «да», если больше и равно — ответ «нет».

Условие цикла — простое: пока значение переменной a меньше значения константы N. Цикл будет выполняться «по кругу»: когда выполнятся команды в строках 4 и 5, после проверки условия цикла (результат проверки — «да» или «нет») снова выполнятся команды строк 4 и 5.

А в этих строках — 1) команда вывода значения переменной a на экран,

2) **две** операции присваивания: в переменную a запишем значение переменной b, а в переменную b запишем **результат операции сложения** a+b.

Таким образом, значение переменной a будет постоянно расти, пока значение не превысит 1000. Именно тогда условие цикла перестанет выполняться, и произойдет выход из цикла. Только тогда выполнится последняя строка: команда вывода на экран… пустоты. Зачем?

Все просто. Печать на экран выполняется запуском **функции** с названием print(). У функции **всегда** сразу после названия стоят открывающая и закрывающая скобки.

Внутри скобок — данные, которые надо вывести на экран, напечатать. Их можно написать сколько угодно. Главное, чтобы они были перечислены через запятую.

Здесь использован необязательный аргумент функции print() — end. Его применение позволяет печатать числа Фибоначчи через пробел, а не с новой строки, как это происходит обычно. И последняя строка 6 в программе печатает совсем не пустоту (внутри скобок ничего нет), а **невидимый символ перевода на новую строку**. Так что повторный запуск программы напечатает числа Фибоначчи с новой строки.

**Самостоятельная работа**

Мы познакомились с правилами языка программирования Python. Это правила написания **присваивания значений** переменным (знак «=»), **сравнения**(знак «меньше», «<»), а также правила выполнения функций. В нашем случае, команды печати.

Важнейшим для программы структурным элементом стал цикл «пока», while. Именно он заставляет выполняться программу до победного конца.

В качестве самостоятельной работы попробуйте:

1. написать и запустить разобранную программу,
2. изменить значение константы N; Сколько чисел Фибоначчи до N = 10000?
3. изменить и/или убрать необязательный аргумент end функции print,
4. убрать или изменить 6-ю строку.
5. по желанию: напишите программу, которая суммирует **все целые положительные числа**от 1 до заданного в программе константой. Например, N = 1000.
6. по желанию: напишите программу, которая суммирует **все целые нечетные положительные числа** от 1 до заданного в программе константой. Например, N = 1000.