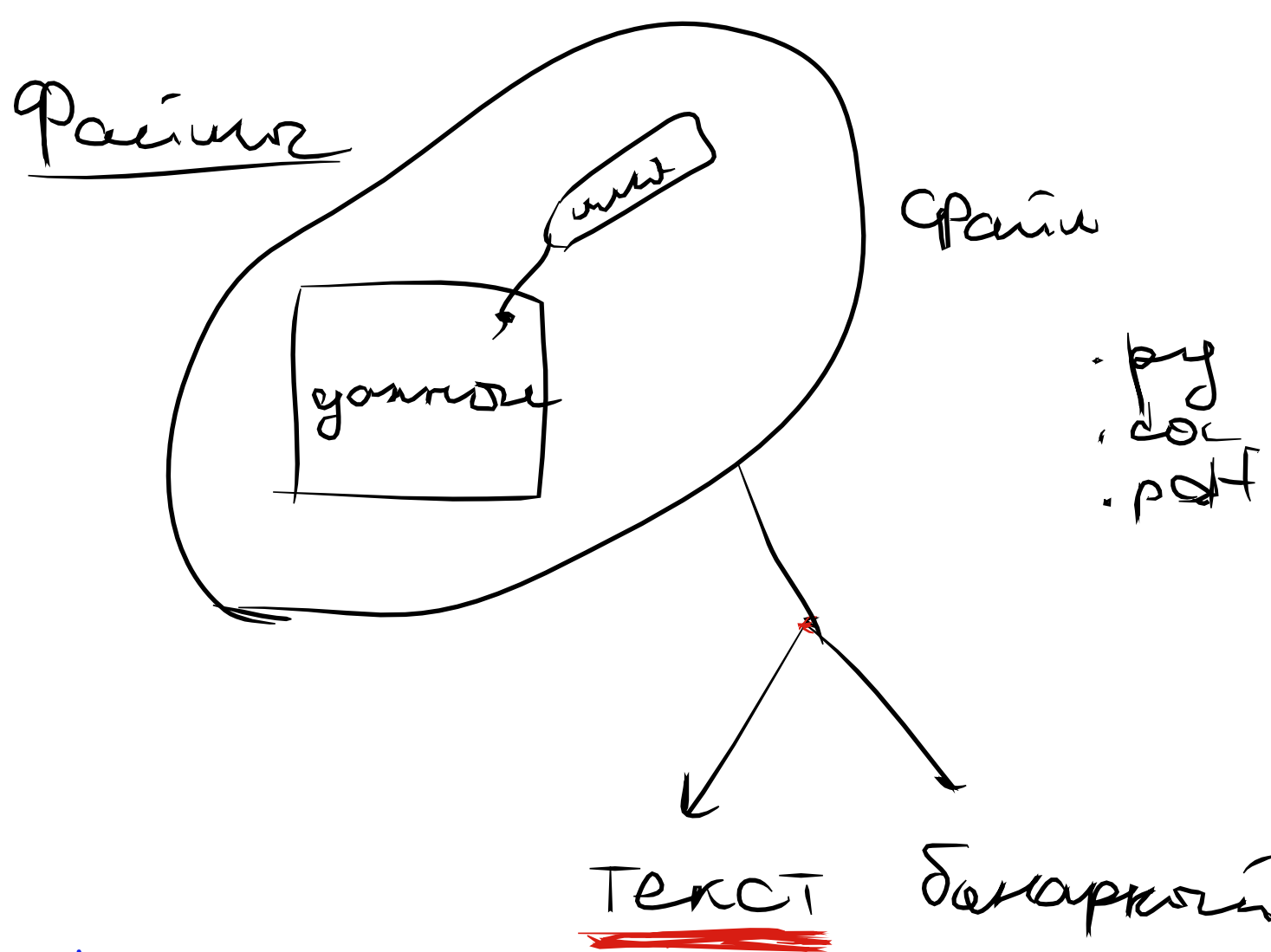


25.11.



## 1. Открытие файла

`open()` # открывает файл, возвращает объект файла  
 filename - имя файла # строка  
 mode - режим работы # строка  
 encoding - кодировка utf-8 # строка  
 windows # 1251

## 2. Режимы работы

w - write (запись)  
 r - read (чтение)

## 3. Работа с файлами

I. `f = open(...)` # открытие

# что-то делаем

`f.close()` # закрываем файл !

II. with `open(...)` as `f` # что-то делаем

with `open("in.txt", "r")` as `input_file`,  
`open("out.txt", "w")` as `output_file`:  
 # что-то делаем

## 4. Чтение из файла

`f = open("in.txt", "r")`

`f.read()` - читает все из файла (не рекомендуется)

✓ `f.readline()` - читает одну строку

`f.readlines()` - читает все строки в список (не рекомендуется)

✓ `f.read(n)` - читает определенное кол-во (n) UTF-8 - самая популярная кодировка байт

1 символ - 1 байт

`f.read(1)` # читать один символ

for line in `f`: # итерационный объект

`print(line)`

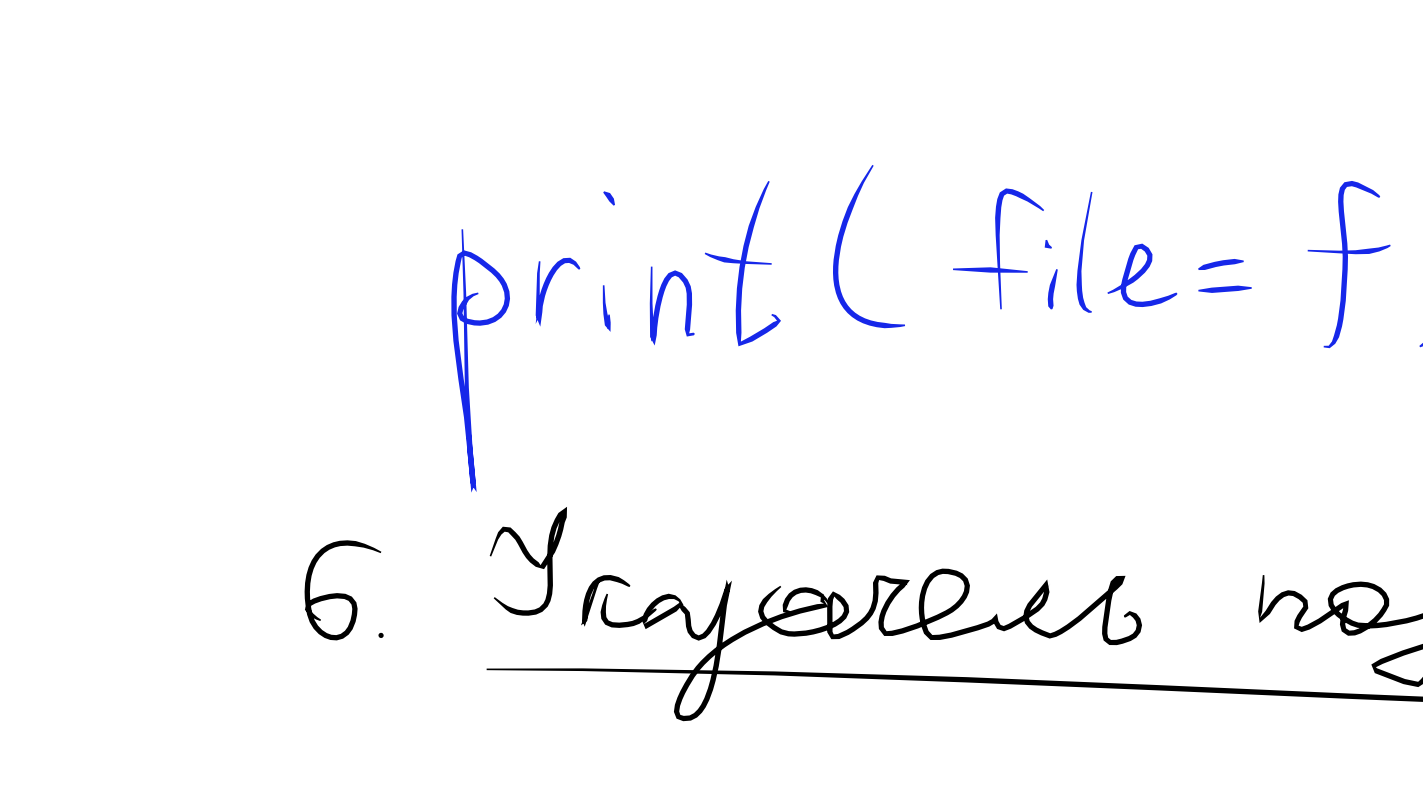
## 5. Запись в файл

`f.write(your_text)` # записывает в файл данные в одну строку и сразу переводит строку

`f.writelines(list_of_strings)` # записывает строки в файл `list[0]` - 1-я строка и т.д.

`print(file=f)` # печатает в файл `f`

## 6. Указатель позиции в файле



Указатель перемещается по файлу. При чтении файла указатель постепенно перемещается к концу файла и после чтения всего файла находится в конце.

`f.seek(n)` - указатель перемещается к позиции `n`

`f.seek(0)` - возврат к началу файла

`f.tell()` - вернуть позицию указателя (число)

## Задача №24 (27686)

Текстовый файл состоит не более чем из  $10^6$  символов X, Y и Z. Определите длину самой длинной последовательности, состоящей из символов X. Хотя бы один символ X находится в последовательности. Для выполнения этого задания следует написать программу. Ниже приведен файл, который необходимо обработать с помощью данного алгоритма.

`max_seq_len = 0` # длина самой длинной  
`seq_len = 0` # длина текущей последовательности  
`f = open("in.txt", "r")`  
 while True:  
`sym = f.read(1)`  
 if not sym: # if sym == "":  
 break  
 if sym == "X":  
`seq_len += 1`  
 else:  
 if `seq_len > max_seq_len`:  
`max_seq_len = seq_len`  
`seq_len = 0`  
`f.close()`  
`print(max_seq_len)`

## Задача №24 (27421)

Текстовый файл состоит не более чем из 106 символов X, Y и Z. Определите максимальное количество идущих подряд символов B, среди которых каждые два соседних различны. Для выполнения этого задания следует написать программу. Ниже приведен файл, который необходимо обработать с помощью данного алгоритма.

`max_seq_len = 0`  
`seq_len = 0`  
`f = open("in.txt", "r")`  
`last_sym = f.read(1)`  
 while `last_sym`:  
`sym = f.read(1)`  
 if `last_sym != sym`:  
`seq_len += 1`  
 else:  
 if `seq_len > max_seq_len`:  
`max_seq_len = seq_len`  
`seq_len = 0`  
`last_sym = sym`  
`print(max_seq_len)`

## Задача №24 (27692)

Текстовый файл состоит не более чем из 106 символов A, B и C. Определите максимальное количество идущих подряд символов B, для выполнения этого задания следует написать программу. Ниже приведен файл, который необходимо обработать с помощью данного алгоритма.

`max_seq_len = 0`  
`seq_len = 0`  
`f = open("in.txt", "r")`  
 while True:  
`sym = f.read(1)`  
 if `sym == "B"`:  
`seq_len += 1`  
 else:  
 if `seq_len > max_seq_len`:  
`max_seq_len = seq_len`  
`seq_len = 0`  
`print(max_seq_len)`

ABCA BB CABC ...  
 5 3

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...

ABCBACBAC ...