



Произвольный доступ к файлам

Урок №31

Разбор домашней работы





ЧТО НАС ЖДЕТ ?

ПЛАН УРОКА

- Повторение: файлы
- Теория: перемотка позиции
- Решаем задачи
- Теория: получение позиции
- Решаем задачи
- Теория: “Подсматривание” символа
- Решаем задачи
- Подводим итоги

Повторение

Повторение



Какие типы данных используются
для файловых потоков?

Повторение

Какие типы данных используются для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Повторение

Как считать символ из
файлового потока “input”?

Какие типы данных используются
для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Повторение

Как считать символ из
файлового потока “input”?

```
char ch = input.get();
```

Какие типы данных используются
для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Повторение

Как считать символ из
файлового потока “input”?

```
char ch = input.get();
```

Какие типы данных используются
для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Как обозначается конец файла?

Повторение

Как считать символ из
файлового потока “input”?

```
char ch = input.get();
```

Какие типы данных используются
для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Как обозначается конец файла?

```
input.eof();
```

Повторение

Как считать символ из
файлового потока “input”?

```
char ch = input.get();
```

Как считать строку из
файлового потока “input”?

Какие типы данных используются
для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Как обозначается конец файла?

```
input.eof();
```

Повторение

Как считать символ из
файлового потока “input”?

```
char ch = input.get();
```

Как считать строку из
файлового потока “input”?

```
getline(input, str);
```

Какие типы данных используются
для файловых потоков?

`ifstream` - поток ввода

`ofstream` - поток вывода

Как обозначается конец файла?

```
input.eof();
```

Теория



15 минут





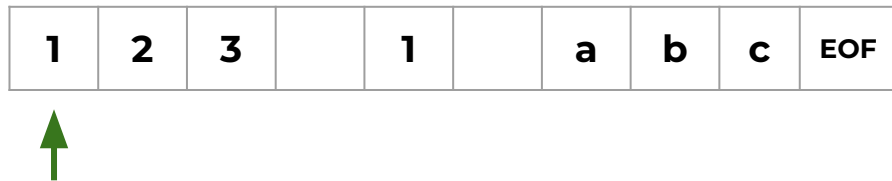
Чтение файла

Чтение файла

Во время создания потока чтения файла создается указатель на начало файла.

```
ifstream input("input.txt");
```

Байтовое представление файла



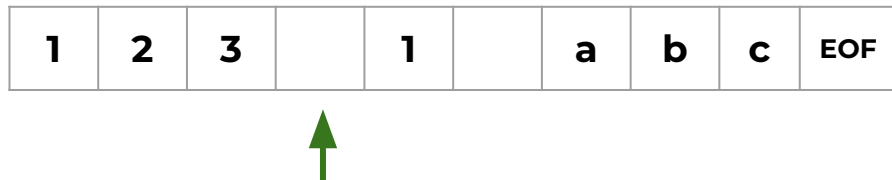
Чтение файла

Во время создания потока чтения файла создается указатель на начало файла.

```
ifstream input("input.txt");
```

```
int number_1;
```

```
input >> number_1;
```



При чтении файла, каретка движется по файлу **вправо**.

Чтение файла

Во время создания потока чтения файла создается указатель на начало файла.

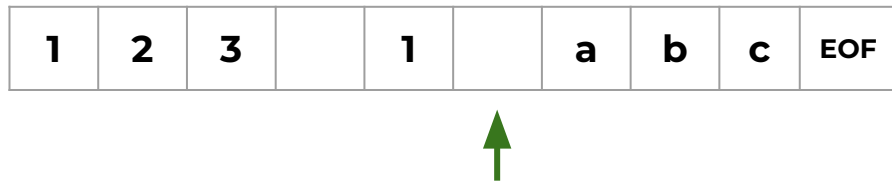
```
ifstream input("input.txt");
```

```
int number_1;
```

```
input >> number_1;
```

```
int number_2;
```

```
input >> number_2;
```



Чтение файла

Во время создания потока чтения файла создается указатель на начало файла.

```
ifstream input("input.txt");
```

```
int number_1;
```

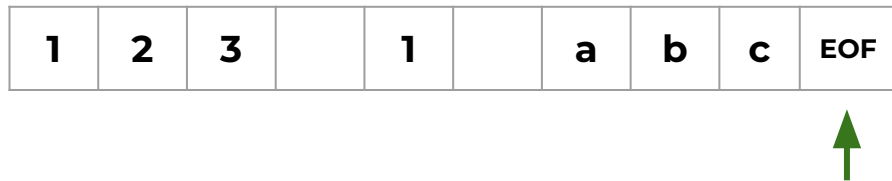
```
input >> number_1;
```

```
int number_2;
```

```
input >> number_2;
```

```
string str;
```

```
input >> str;
```



Чтение файла

Во время создания потока чтения файла создается указатель на начало файла.

```
ifstream input("input.txt");
```

```
int number_1;
```

```
input >> number_1;
```

```
int number_2;
```

```
input >> number_2;
```

```
string str;
```

```
input >> str;
```

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Пока что мы умеем двигать каретку только вправо и только в момент считывания данных в переменную

Чтение файла

Во время создания потока чтения файла создается указатель на начало файла.

```
ifstream input("input.txt");
```

```
int number_1;
```

```
input >> number_1;
```

```
int number_2;
```

```
input >> number_2;
```

```
string str;
```

```
input >> str;
```

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Хотим научиться двигать каретку в произвольное место

**Перемотка позиции
следующего символа**



Перемотка позиции следующего СИМВОЛА

```
input.seekg(pos, start)
```

Где:

pos - сколько символов перематываем

start - откуда начинаем перемотку:

- **ios::beg** - с начала файла
- **ios::cur** - с текущей позиции
- **ios::end** - с конца файла

Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

```
input.seekg(2, ios::cur)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

```
input.seekg(2, ios::cur)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

```
input.seekg(2, ios::cur)
```

```
input.seekg(-3, ios::cur)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

```
input.seekg(2, ios::cur)
```

```
input.seekg(-3, ios::cur)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

```
input.seekg(2, ios::cur)
```

```
input.seekg(-3, ios::cur)
```

```
input.seekg(-2, ios::end)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

```
input.seekg(1, ios::beg)
```

```
input.seekg(2, ios::cur)
```

```
input.seekg(-3, ios::cur)
```

```
input.seekg(-2, ios::end)
```

На какую позицию встанем?

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

Давайте попробуем считать символ, находясь в состоянии EOF

```
string s;  
input >> s;
```

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



Перемотка позиции следующего символа

Давайте попробуем считать символ, находясь в состоянии EOF

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



```
string s;
```

```
input >> s; // в s будет лежать input.eof()
```

После такой операции файловый поток ввода будет считать, что он завершил работу с файлом и **seekg** **перестанет работать**

Перемотка позиции следующего символа

Для возвращения в начало и повторного считывания файла:

```
input.clear(); // разблокирует seekg  
input.seekg(0, ios::beg);
```

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----





Перемотка позиции следующего СИМВОЛА

Для перемещения каретки во время **вывода** данных в файл существует аналогичная функция:

`output.seekp(pos, start)`

Где:

pos - сколько символов перематываем

start - откуда начинаем перемотку:

- `ios::beg` - с начала файла
- `ios::cur` - с текущей позиции
- `ios::end` - с конца файла

Практика



15 минут



Теория



10 минут



**Как узнать на какой позиции
мы находимся в данный
момент?**

Как узнать на какой позиции мы находимся в данный момент?

Есть решение!



Позиция следующего символа

Для получения текущей позиции каретки есть функция:

```
int pos = input.tellg();
```

Пример:

1	2	3		1		a	b	c	EOF
---	---	---	--	---	--	---	---	---	-----



0



4

*Итерация начинается с нуля,
аналогично массивам*

Как узнать длину файла?

Как узнать длину файла?

Есть предположения?



Длина файла

Узнать длину файла:

```
ifstream input("input.txt");  
input.seekg(0, ios::end);    // ставим каретку в позицию EOF  
int length = input.tellg();
```

Практика



10 минут



Теория



5 минут



“Подглядывание” символа



“Подглядывание” символа

Обычно мы читаем символ и сразу сдвигаем позицию на следующий:

```
char ch = input.get();
```

Но если нужно читать символ, и не двигать позицию на следующий, есть функция:

```
char ch = input.peek();
```

“Подглядывание” символа

```
char ch = input.peek(); // ch = 'H'
```

H	e	l	l	o	!	EOF
---	---	---	---	---	---	-----



“Подглядывание” символа

```
char ch = input.peek(); // ch = 'H'
```

```
char ch = input.peek(); // ch = 'H'
```

H	e	l	l	o	!	EOF
---	---	---	---	---	---	-----



“Подглядывание” символа

```
char ch = input.peek(); // ch = 'H'
```

```
char ch = input.peek(); // ch = 'H'
```

```
char ch = input.get(); // ch = 'H'
```

H	e	l	l	o	!	EOF
---	---	---	---	---	---	-----



“Подглядывание” символа

```
char ch = input.peek(); // ch = 'H'
```

```
char ch = input.peek(); // ch = 'H'
```

```
char ch = input.get(); // ch = 'H'
```

```
char ch = input.peek(); // ch = 'e'
```

H	e	l	l	o	!	EOF
---	---	---	---	---	---	-----



Практика



10 минут



Итоги урока

1. Какая функция перематывает позицию в файле?
2. Какая функция позволяет получить позицию в файле?
3. Как узнать длину файла?
4. Как считать символ без передвижения позиции?