

# ЛАБОРАТОРНАЯ РАБОТА №2 ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ОС MS WINDOWS

## Часть 2. Язык интерпретатора и командные файлы

**Цель работы** – знакомство с языком интерпретатора командной строки ОС MS Windows и командными файлами

### 1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

#### 1.1 Язык интерпретатора Cmd.exe. Командные файлы

Язык оболочки командной строки (shell language) в Windows реализован в виде командных (или пакетных) файлов. **Командный файл** в Windows — это обычный текстовый файл с расширением bat или cmd, в котором записаны допустимые команды ОС (как внешние, так и внутренние), а также некоторые дополнительные инструкции и ключевые слова, придающие командным файлам некоторое сходство с программами, написанными на языке программирования. Например, если записать в файл deltmp.bat следующие команды:

```
C:\
CD %TEMP%
DEL /F *.tmp
```

и запустить его на выполнение (аналогично исполняемым файлам с расширением com или exe), то мы удалим все файлы во временной директории Windows. Таким образом, исполнение командного файла приводит к тому же результату, что и последовательный ввод записанных в нем команд. При этом не проводится никакой предварительной компиляции или проверки синтаксиса кода; если встречается строка с ошибочной командой, то она игнорируется. Очевидно, что если приходится часто выполнять одни и те же действия, то использование командных файлов может сэкономить много времени.

**Примечание:** чтобы командная строка сразу не закрывалась добавьте в конце строку:

PAUSE

Результатом будет ожидание нажатия любой клавиши:

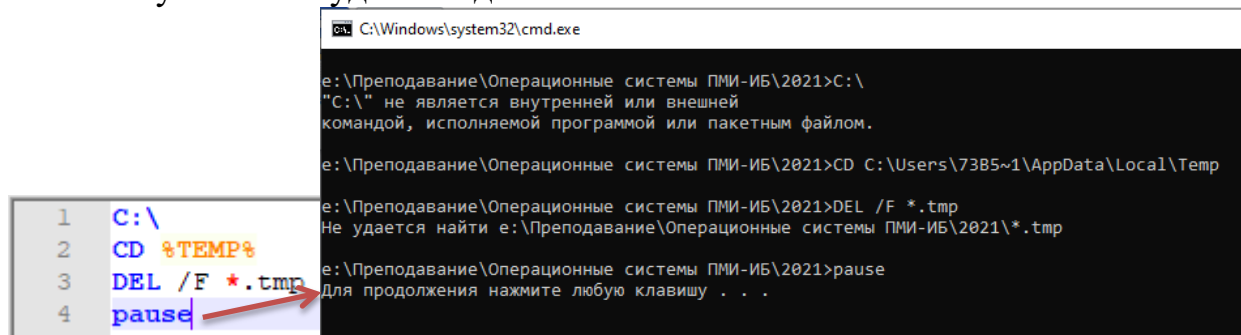


Рисунок 1 – пример использования PAUSE

#### 1.1.1 Вывод сообщений и дублирование команд

По умолчанию команды пакетного файла перед исполнением выводятся на экран, что выглядит не очень эстетично. С помощью команды

ECHO OFF можно отключить дублирование команд, идущих после нее (сама команда ECHO OFF при этом все же дублируется). В первом случае сама команда выводится на экран. После команды ECHO OFF команда не дублируется (Рисунок 2).

Например,

REM Следующие две команды будут дублироваться на экране ...

:: эта строка – такой же комментарий, как и предыдущая

DIR C:\

ECHO OFF

:: А остальные уже не будут

DIR D:\

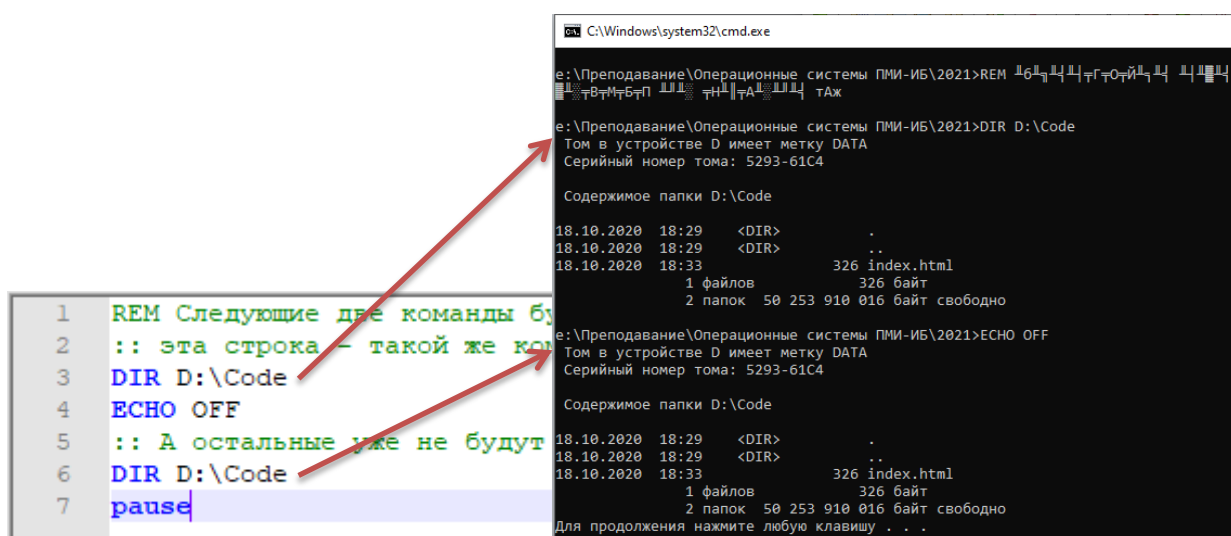


Рисунок 2 – пример работы ECHO OFF

Для восстановления режима дублирования используется команда ECHO ON. Кроме этого, можно отключить дублирование любой отдельной строки в командном файле, написав в начале этой строки символ @, например:

ECHO ON

:: Команда DIR C:\ дублируется на экране

DIR C:\

:: А команда DIR D:\ — нет

@DIR D:\

Таким образом, если поставить в самое начало файла команду @ECHO OFF, то это решит все проблемы с дублированием команд.

В пакетном файле можно выводить на экран строки с сообщениями. Делается это с помощью команды ECHO сообщение

Например:

@ECHO OFF

ECHO Привет!

Команда ECHO. (точка должна следовать непосредственно за словом "ECHO") выводит на экран пустую строку. Например:

@ECHO OFF

ECHO Привет!

ECHO.

ECHO Пока!

Часто бывает удобно для просмотра сообщений, выводимых из пакетного файла, предварительно полностью очистить экран командой CLS.

Используя механизм *перенаправления ввода/вывода* (символы > и >>), можно направить сообщения, выводимые командой ECHO, в определенный текстовый файл. Например:

```
@ECHO OFF
```

```
ECHO Привет! > hi.txt
```

```
ECHO Пока! >> hi.txt
```

С помощью такого метода можно, скажем, заполнять файлы-протоколы с отчетом о произведенных действиях. Например:

```
@ECHO OFF
```

```
REM Попытка копирования
```

```
XCOPY C:\PROGRAMS D:\PROGRAMS /s
```

:: Добавление сообщения в файл report.txt в случае удачного завершения копирования

```
IF NOT ERRORLEVEL 1 ECHO Успешное копирование >> report.txt
```

### ***1.1.2 Использование параметров командной строки***

При запуске пакетных файлов в командной строке можно указывать произвольное число параметров, значения которых можно использовать внутри файла. Это позволяет, например, применять один и тот же командный файл для выполнения команд с различными параметрами.

Для доступа из командного файла к параметрам командной строки применяются символы %0, %1, ..., %9 или %\*. При этом вместо %0 подставляется имя выполняемого пакетного файла, вместо %1, %2, ..., %9 — значения первых девяти параметров командной строки соответственно, а вместо %\* — все аргументы. Если в командной строке при вызове пакетного файла задано меньше девяти параметров, то "лишние" переменные из %1 — %9 замещаются пустыми строками. Рассмотрим следующий пример. Пусть имеется командный файл corier.bat следующего содержания:

```
@ECHO OFF
```

```
CLS
```

```
ECHO Файл %0 копирует каталог %1 в %2
```

```
XCOPY %1 %2 /S
```

Если запустить его из командной строки с двумя параметрами, например

```
corier.bat C:\Programs D:\Backup
```

то на экран выведется сообщение

```
Файл corier.bat копирует каталог C:\Programs в D:\Backup
```

и произойдет копирование каталога C:\Programs со всеми его подкаталогами в D:\Backup.

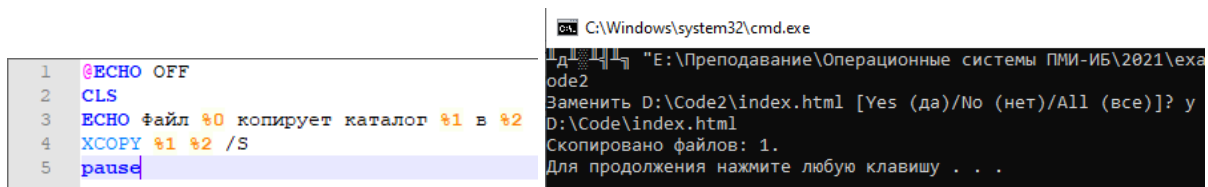


Рисунок 3 – пример использования параметров

При необходимости можно использовать более девяти параметров командной строки. Это достигается с помощью команды SHIFT, которая изменяет значения замещаемых параметров с %0 по %9, копируя каждый параметр в предыдущий, то есть значение %1 копируется в %0, значение %2 – в %1 и т.д. Замещаемому параметру %9 присваивается значение параметра, следующего в командной строке за старым значением %9. Если же такой параметр не задан, то новое значение %9 — пустая строка.

### 1.1.3 Работа с переменными среды

Внутри командных файлов можно использовать так называемые **переменные среды** (или переменными окружения), каждая из которых хранится в оперативной памяти, имеет свое уникальное имя, а ее значением является **строка**. Стандартные переменные среды автоматически инициализируются в процессе загрузки операционной системы. Такими переменными являются:

- WINDIR, которая определяет расположение каталога Windows,
- TEMP, которая определяет путь к каталогу для хранения временных файлов Windows
- PATH, в которой хранится системный путь (путь поиска), то есть список каталогов, в которых система должна искать выполняемые файлы или файлы совместного доступа (например, динамические библиотеки).

Кроме того, в командных файлах с помощью команды SET можно объявлять собственные переменные среды.

Для получения значения определенной переменной среды нужно заключить имя этой переменной в символы %. Например:

@ECHO OFF

CLS

:: Создание переменной MyVar

SET MyVar=Привет

:: Изменение переменной

SET MyVar=%MyVar%

ECHO Значение переменной MyVar: %MyVar%

:: Удаление переменной MyVar

SET MyVar=

ECHO Значение переменной WinDir: %WinDir%

При запуске такого командного файла на экран выведется строка

Значение переменной MyVar: Привет!

Значение переменной WinDir: C:\WINDOWS

```
1 @ECHO OFF
2 CLS
3 :: Создание переменной MyVar
4 SET MyVar=Hello
5 :: Изменение переменной
6 SET MyVar=%MyVar%
7 ECHO MyVar: %MyVar%
8 :: Удаление переменной MyVar
9 SET MyVar=
10 ECHO WinDir: %WinDir%
11 pause
```

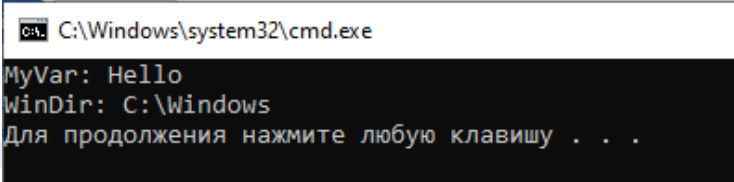


Рисунок 4 – пример работы со строковыми переменными

#### 1.1.4 Преобразования переменных как строк

С переменными среды в командных файлах можно производить некоторые манипуляции. Над ними можно производить операцию конкатенации (соединения). Для этого нужно в команде SET просто написать рядом значения соединяемых переменных. Например,

```
SET A=Раз
SET B=Два
SET C=%A%%B%
```

```
1 @echo off
2 SET A=one
3 SET B=two
4 SET C=%A%%B%
5 echo %A%
6 echo %B%
7 echo %C%
8 pause
```

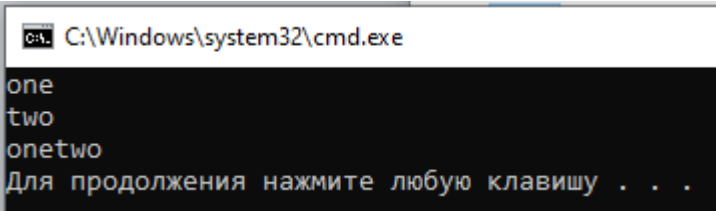


Рисунок 5 – пример работы со строками

После выполнения в файле этих команд значением переменной C будет являться строка 'РазДва'. Не следует для конкатенации использовать знак +, так как он будет воспринят просто в качестве символа. Например, после запуска файл следующего содержания

```
SET A=Раз
SET B=Два
SET C=A+B
ECHO Переменная C=%C%
SET D=%A%+%B%
ECHO Переменная D=%D%
на экран выведутся две строки:
Переменная C=A+B
Переменная D=Раз+Два
```

#### 1.1.5 Операции с переменными как с числами

При включенной расширенной обработке команд (этот режим в Windows используется по умолчанию) имеется **возможность рассматривать значения переменных среды как числа** и производить с ними арифметические вычисления (используются ТОЛЬКО целые числа). Для

этого используется команда SET с ключом /A. Ниже приведен пример пакетного файла add.bat, складывающего два числа, заданных в качестве параметров командной строки, и выводящего полученную сумму на экран:

```
@ECHO OFF
:: В переменной M будет храниться сумма
SET /A M=%1+%2
ECHO Сумма %1 и %2 равна %M%
:: Удалим переменную M
SET M=
```

В команде SET с ключом /A могут использоваться операции – (вычитание), \* (умножение), / (деление нацело), % (остаток от деления). При использовании знака % в качестве знака операции в **командных файлах** он должен быть записан ДВАЖДЫ.

**Рекомендуется** при инициализации числовых переменных использовать ключ /A

```
SET /A col=0
```

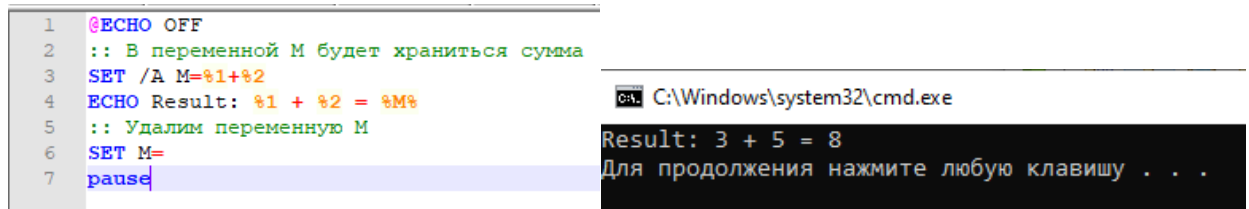


Рисунок 6 – пример работы с переменными как с числами

### 1.1.6 Ввод значения переменной с клавиатуры

Ввод значения переменной при выполнении командного файла выполняется командой SET с ключом /P. Например, для ввода значения переменной M следует использовать команду

```
SET /P M=[введите M]
```

Текст подсказки [введите M] будет выведен на экран.

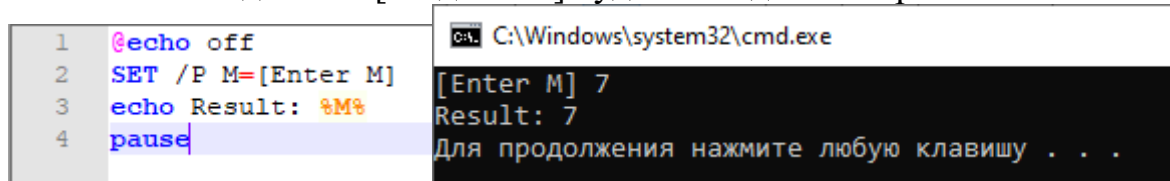


Рисунок 7 – пример ввода данных с клавиатуры

### 1.1.7 Операторы перехода GOTO и вызова CALL

Командный файл может содержать метки и команды GOTO перехода к этим меткам. Любая строка, начинающаяся с двоеточия :, воспринимается при обработке командного файла как метка. Имя метки задается набором символов, следующих за двоеточием до первого пробела или конца строки.

**Пример.** Пусть имеется командный файл следующего содержания:

```
@ECHO OFF
COPY %1 %2
GOTO Label1
ECHO Эта строка никогда не выполнится
:Label1
```

:: Продолжение выполнения

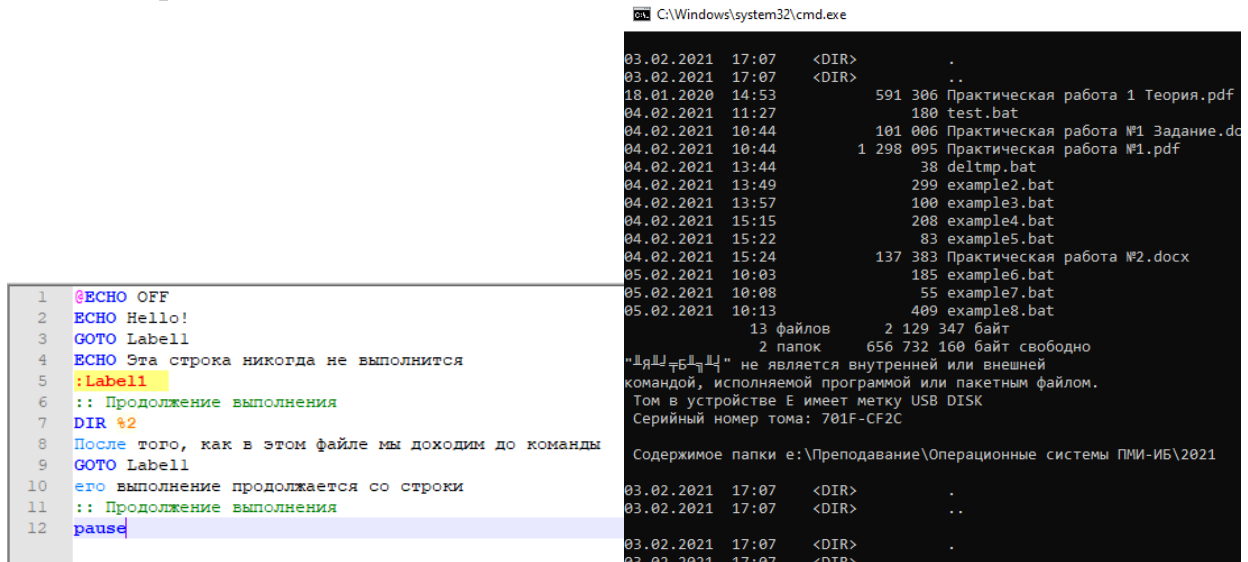
DIR %2

После того, как в этом файле мы доходим до команды

GOTO Label1

его выполнение продолжается со строки

:: Продолжение выполнения



```
1 @ECHO OFF
2 ECHO Hello!
3 GOTO Label1
4 ECHO Эта строка никогда не выполнится
5 :Label1
6 :: Продолжение выполнения
7 DIR %2
8 После того, как в этом файле мы доходим до команды
9 GOTO Label1
10 его выполнение продолжается со строки
11 :: Продолжение выполнения
12 pause
```

```
C:\Windows\system32\cmd.exe
03.02.2021 17:07 <DIR> .
03.02.2021 17:07 <DIR> ..
18.01.2020 14:53 591 306 Практическая работа 1 Теория.pdf
04.02.2021 11:27 180 test.bat
04.02.2021 10:44 101 006 Практическая работа №1 Задание.doc
04.02.2021 10:44 1 298 095 Практическая работа №1.pdf
04.02.2021 13:44 38 deltmp.bat
04.02.2021 13:49 299 example2.bat
04.02.2021 13:57 100 example3.bat
04.02.2021 15:15 208 example4.bat
04.02.2021 15:22 83 example5.bat
04.02.2021 15:24 137 383 Практическая работа №2.docx
05.02.2021 10:03 185 example6.bat
05.02.2021 10:08 55 example7.bat
05.02.2021 10:13 409 example8.bat
13 файлов 2 129 347 байт
2 папок 656 732 160 байт свободно
"Имя файла" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
Том в устройстве E имеет метку USB DISK
Серийный номер тома: 701F-CF2C

Содержимое папки e:\Преподавание\Операционные системы ПМИ-ИБ\2021
03.02.2021 17:07 <DIR> .
03.02.2021 17:07 <DIR> ..
03.02.2021 17:07 <DIR> .
03.02.2021 17:07 <DIR> ..
```

Рисунок 8 - пример использования оператора GOTO

В команде перехода внутри файла GOTO можно задавать в качестве метки перехода строку :EOF, которая передает управление в конец текущего пакетного файла (это позволяет легко выйти из пакетного файла без определения каких-либо меток в самом его конце).

### 1.1.8 Оператор проверки условия IF

С помощью команды IF ... ELSE (ключевое слово ELSE может отсутствовать) в пакетных файлах можно выполнять обработку условий нескольких типов. При этом если заданное после IF условие принимает истинное значение, система выполняет следующую за условием команду (или несколько команд, заключенных в круглые скобки), в противном случае выполняется команда (или несколько команд в скобках), следующие за ключевым словом ELSE.

Первый тип условия используется обычно для проверки значения переменной. Для этого применяются два варианта синтаксиса команды IF:

IF [NOT] строка1==строка2 команда1 [ELSE команда2]

(квадратные скобки указывают на необязательность заключенных в них параметров) или

IF [/I] [NOT] строка1 оператор\_сравнения строка2 команда

Рассмотрим сначала первый вариант. Условие строка1==строка2 (здесь необходимо писать именно два знака равенства – как и в программах на C/C++) считается истинным при точном совпадении обеих строк. Параметр NOT указывает на то, что заданная команда выполняется лишь в том случае, когда сравниваемые строки не совпадают.



Для *группировки команд* могут использоваться круглые скобки. Иногда использование круглых скобок необходимо для правильной работы команды if...else – например для вывода на экран наибольшего из двух параметров, с которыми запущен командный файл, следует использовать оператор

if %1 GTR %2 (echo %1 ) else (echo %2)

Строки могут быть литеральными или представлять собой значения переменных (например, %1 или %ТЕМР%). Кавычки для литеральных строк **не требуются**. Например,

IF %1==%2 ECHO Параметры совпадают!

IF %1==windows ECHO значение первого параметра – windows

```

1 @echo off
2 IF %1==%2 ECHO The strings are identical!
3 IF %1==windows ECHO The value of the first parameter - windows
4 pause

```

C:\Windows\system32\cmd.exe

The strings are identical!  
The value of the first parameter - windows  
Для продолжения нажмите любую клавишу . . .

Рисунок 9 – Пример использования оператора IF

Операторы сравнения представлены в таблице 1.

Таблица 1 – операторы сравнения

Операторы сравнения в IF	Значение
EQL	Равно
NEQ	Не равно
LSS	Меньше
LEQ	Меньше или равно
GTR	Больше
GEQ	Больше или равно

### 1.1.9 Организация циклов Цикл FOR ... IN ... DO ...

В командных файлах для организации циклов используются несколько разновидностей оператора FOR, которые обеспечивают следующие функции:

- выполнение заданной команды для всех элементов указанного множества;
- выполнение заданной команды для всех подходящих имен файлов;
- выполнение заданной команды для всех подходящих имен каталогов;
- выполнение заданной команды для определенного каталога, а также всех его подкаталогов;
- получение последовательности чисел с заданными началом, концом и шагом приращения;
- чтение и обработка строк из текстового файла;



– обработка строк вывода определенной команды.

Самый простой вариант синтаксиса команды FOR для командных файлов имеет следующий вид:

FOR %%переменная IN (множество)

DO команда [параметры]

**Внимание!**

Перед названием переменной должны стоять именно два знака процента (%%), а не один, как это было при использовании команды FOR непосредственно из командной строки!

**Пример.** Если в командном файле заданы строки

@ECHO OFF

FOR %%i IN (Раз, Два, Три) DO ECHO %%i

то в результате его выполнения на экран будет выведено следующее:

Раз

Два

Три

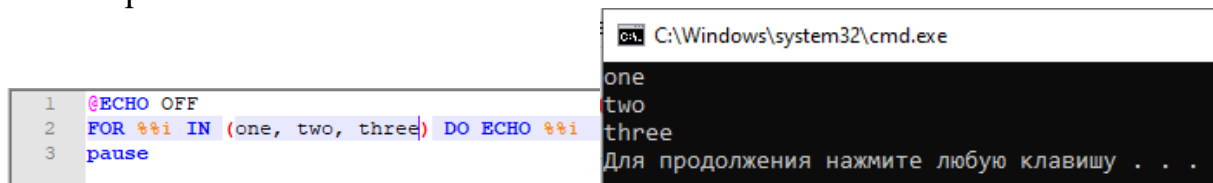


Рисунок 10 – пример использования цикла FOR

## 1.2 Использование пакетных файлов \*.bat в Visual Studio

Ряд практических задач требуют средств, которые среда программирования предоставить не может. Частным случаем является обращение к операционной системе. Так как ОС имеет командную строку и собственные исполняемые файлы, то существует возможность их вызова из кода программы. Для этого необходимо воспользоваться следующим подходом:

1. Подключается библиотека System.Diagnostics для использования класса Process:

```
using System.Windows.Forms;  
// Для использования класса Process  
using System.Diagnostics;
```

2. Создается экземпляр класса Process для того, чтобы инициировать выполнение нового процесса, включающего наш пакетный файл:

```
// Создаем процесс для выполнения выбранного файла  
Process myProcess = new Process();
```

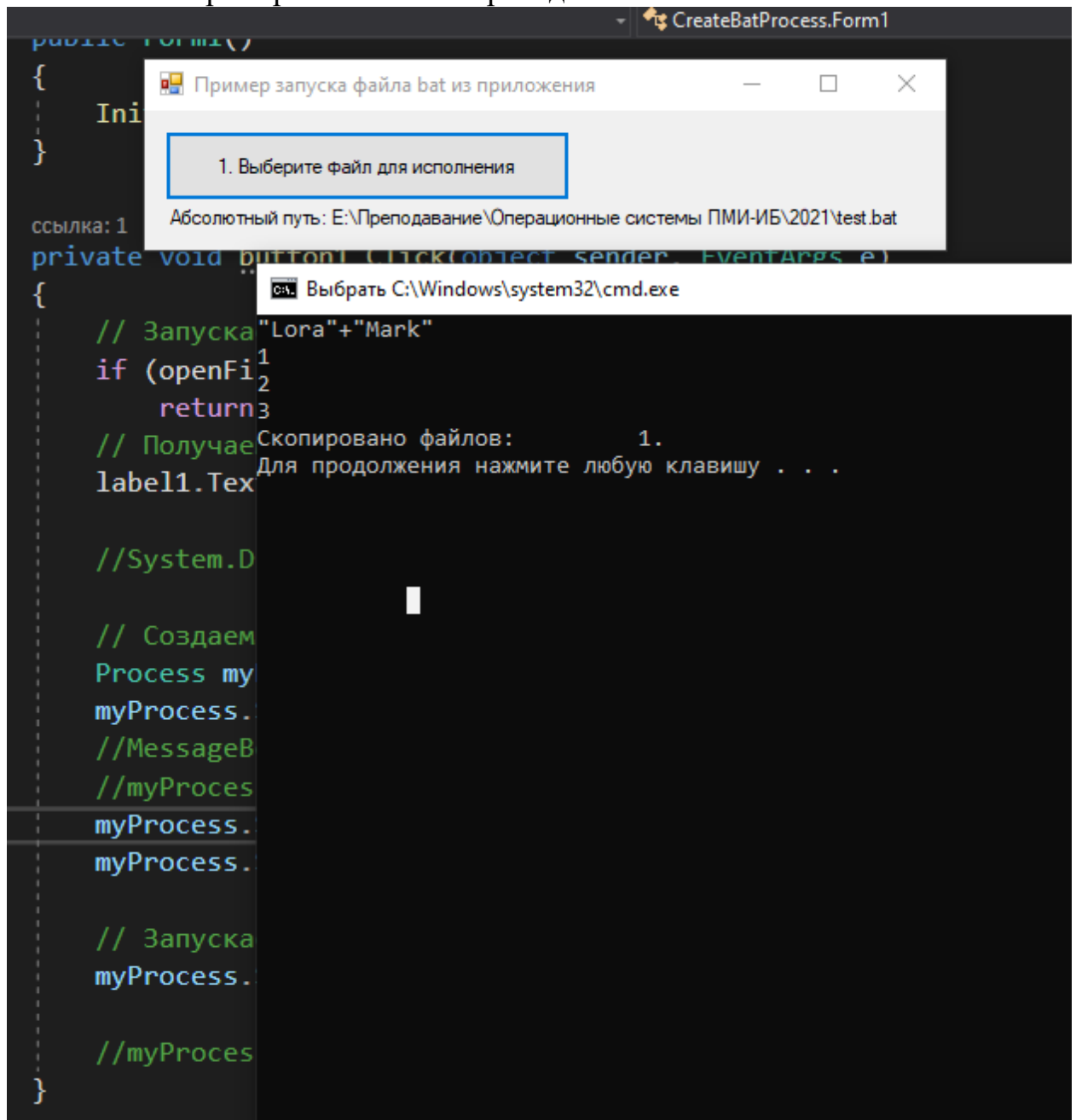
3. Занесем в атрибут FileName экземпляра класса Process путь к пакетному файлу:

```
myProcess.StartInfo.FileName = openFileDialog1.FileName;
```

4. Запустим пакетный файл в процессе:

```
// Запускаем пакетный файл на исполнение  
myProcess.Start();
```

Результатом выполнения будет являться запуск пакетного файла на исполнение. Пример выполнения приведен ниже:



#### Задание на лабораторную работу:

1. Ознакомиться с теоретической частью.
2. Повторить практические примеры и прокомментировать каждую строку кода так, чтобы было ясно что она делает. Ответить на вопросы: что принимает как входные параметры, что является результатом. Каждый пример оформить отдельным пакетным файлом \*.bat.
3. Реализовать пакетный файл (\*.bat) согласно задания из таблицы заданий.
4. Реализовать приложение, позволяющее выбрать пакетный файл и запустить его на исполнение на языке высокого уровня.

Номер варианта	Задание
1	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$
2	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + C^2 + 17 * x, & \text{если } x > 0 \\ 355, & \text{если } x = 0 \\ x * C & \text{иначе} \end{cases}.$
3	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x + C * x, & \text{если } x > 0 \\ -5 * C, & \text{если } x = 0 \\ x^2 + C & \text{иначе} \end{cases}.$
4	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x, & \text{если } x \geq 0 \\ 7 * C^4, & \text{если } 0 > x \geq -1. \\ x + x^2 + C & \text{иначе} \end{cases}.$
5	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + 7 * x + 22, & \text{если } x > 0 \\ -C, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$
6	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ x^2 - C + 77 & \text{иначе} \end{cases}.$
7	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x^3, & \text{если } x > 0 \\ 5, & \text{если } x = 0 \\ 77 * x + (x * C)^2 + C & \text{иначе} \end{cases}.$
8	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x * 8 * C, & \text{если } x > 0 \\ 55 * C, & \text{если } x = 0 \\ x * C + x^2 - C & \text{иначе} \end{cases}.$
9	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} (x + x) * C * x, & \text{если } x > 0 \\ 75 * C, & \text{если } x = 0 \\ (x + x^2)/C & \text{иначе} \end{cases}.$
10	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p>

	$f(x) = \begin{cases} -x + C, & \text{если } x > 0 \\ -C * 5, & \text{если } x = 0. \\ -x - x^2 + C & \text{иначе} \end{cases}$
11	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 10 - 2 * (x^2 + x), & \text{если } x > 0 \\ 64, & \text{если } x = 0 \\ x^3 + x^2 + C & \text{иначе} \end{cases}.$
12	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + 20 + x * C, & \text{если } x > 0 \\ 255 * C, & \text{если } x = 0 \\ 2 * x + x^2 + C & \text{иначе} \end{cases}.$
13	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + C * x + 111, & \text{если } x > 0 \\ 111 + 12 * C, & \text{если } x = 0 \\ x^2 + C & \text{иначе} \end{cases}.$
14	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 2 * x + 2 * x, & \text{если } x > 0 \\ x^2, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$
15	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 5 * x - x^2, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ -x + x^2 + C & \text{иначе} \end{cases}.$
16	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$
17	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + 4 * C + 17 * x, & \text{если } x > 0 \\ 355, & \text{если } x = 0 \\ x * C & \text{иначе} \end{cases}.$
18	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 2 * x + x + C * x, & \text{если } x > 0 \\ -5 * C, & \text{если } x = 0 \\ x^2 + C & \text{иначе} \end{cases}.$
19	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + x, & \text{если } x \geq 0 \\ 7 * C^4, & \text{если } 0 > x \geq -1. \\ x + x^2 + C & \text{иначе} \end{cases}$
20	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 7 * x + 7 * x + 22, & \text{если } x > 0 \\ -C, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$

21	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 16 * x + x, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ x^2 - C + 77 & \text{иначе} \end{cases}.$
22	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 15 * x + x^3, & \text{если } x > 0 \\ 5, & \text{если } x = 0 \\ 77 * x + (x * C)^2 + C & \text{иначе} \end{cases}.$
23	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x * 8 * C, & \text{если } x > 0 \\ 55 * C, & \text{если } x = 0 \\ x * C + x^2 - C & \text{иначе} \end{cases}.$
24	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} (x + x) * C * x, & \text{если } x > 0 \\ 75 * C, & \text{если } x = 0 \\ (x + x^2)/C & \text{иначе} \end{cases}.$
25	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} -x + C, & \text{если } x > 0 \\ -C * 5, & \text{если } x = 0 \\ -x - x^2 + C & \text{иначе} \end{cases}.$
26	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 10 - 2 * (x^2 + x), & \text{если } x > 0 \\ 64, & \text{если } x = 0 \\ x^3 + x^2 + C & \text{иначе} \end{cases}.$
27	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + 20 + x * C, & \text{если } x > 0 \\ 255 * C, & \text{если } x = 0 \\ 2 * x + x^2 + C & \text{иначе} \end{cases}.$
28	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x + C * x + 111, & \text{если } x > 0 \\ 111 + 12 * C, & \text{если } x = 0 \\ x^2 + C & \text{иначе} \end{cases}.$
29	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 2 * x + 2 * x, & \text{если } x > 0 \\ x^2, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$
30	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} 5 * x - x^2, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ -x + x^2 + C & \text{иначе} \end{cases}.$
31	<p>Вычислить значение функции <math>f(x)</math>. Значения <math>x</math> и <math>C</math> вводятся пользователем с клавиатуры.</p>

	$f(x) = \begin{cases} 13 * x + x, & \text{если } x > 0 \\ 75, & \text{если } x = 0 \\ x + x^2 + C & \text{иначе} \end{cases}.$
32	<p>Вычислить значение функции f(x). Значения x и C вводятся пользователем с клавиатуры.</p> $f(x) = \begin{cases} x - C^3 + 17 * x, & \text{если } x > 0 \\ 355, & \text{если } x = 0 \\ x * C & \text{иначе} \end{cases}.$