

Специфика разработки консольных приложений

Что это такое

Консольные приложения - это программы, которые работают в командной строке. Они часто используются для автоматизации задач, обработки данных и других подобных процессов.

1. Аргументы командной строки

Консольные приложения обычно принимают аргументы командной строки, которые определяют, как программа будет работать. Эти аргументы могут быть переданы при запуске программы, и должны быть правильно обработаны в коде.

1. Аргументы командной строки

Важно убедиться, что приложение обрабатывает все возможные варианты входных параметров и сообщает об ошибках, если аргументы введены неправильно

Microsoft Windows [Version 10.0.19043.1586]

(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>bcdedit

Диспетчер загрузки Windows

идентификатор	{bootmgr}
device	partition=\Device\HarddiskVolume3
path	\EFI\Microsoft\Boot\bootmgfw.efi
description	Windows Boot Manager
locale	ru-RU
inherit	{globalsettings}
default	{current}
resumeobject	{9ada1c18-10ba-11ec-a046-a772c9b60c62}
displayorder	{current}
	{9ada1c16-10ba-11ec-a046-a772c9b60c62}
	{9ada1c1a-10ba-11ec-a046-a772c9b60c62}
toolsdisplayorder	{memdiag}
timeout	5

Загрузка Windows

 Администратор: C:\Windows\System32\cmd.exe

```
C:\Windows\system32>bcdedit /export C:\bcd.bak  
Операция успешно завершена.
```

```
C:\Windows\system32>_
```

C:\Windows\system32>bcdedit /?

BCDEDIT - редактор хранилища данных конфигурации загрузки

Инструмент командной строки Bcdedit.exe изменяет хранилище данных конфигурации загрузки. Хранилище данных конфигурации загрузки содержит параметры конфигурации загрузки и управляет загрузкой операционной системы. Эти параметры ранее хранились в файле Boot.ini (в операционных системах на основе BIOS) или в энергонезависимых записях ОЗУ (в операционных системах на основе EFI). Bcdedit.exe используется для добавления, удаления, изменения и дополнения записей в хранилище данных конфигурации.

Для получения подробных сведений о командах и параметрах введите bcdedit.exe /? <команда>. Например, чтобы получить информацию о команде /createstore, введите:

```
bcdedit.exe /? /createstore
```

Для получения алфавитного списка статей в этом файле справке введите "bcdedit /? TOPICS".

Команды операций над хранилищами

=====

Администратор: C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.19043.1586]

(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>bcdedit /kakoi-to parameter

Указана неизвестная команда.

Для получения справки введите в командной строке "bcdedit /?"

C:\Windows\system32>



```
neo@neo-VirtualBox:~$ lshw
```

ПРЕДУПРЕЖДЕНИЕ: эту программу нужно запускать от имени суперпользователя.

```
neo-virtualbox
```

```
описание: Computer
```

```
разрядность: 64 bits
```

```
возможности: vsyscall32
```

```
*-core
```

```
описание: Motherboard
```

```
физический ID: 0
```

```
*-память
```

```
описание: Системная память
```

```
физический ID: 0
```

```
размер: 4GiB
```

```
*-cpu
```

```
продукт: AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
```

```
производитель: Advanced Micro Devices [AMD]
```

```
физический ID: 1
```

```
сведения о шине: cpu@0
```

```
версия: 23.17.0
```

```
разрядность: 64 bits
```

```
возможности: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic
```

```
sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext f
```

```
pxsr_opt rdtscp x86-64 constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid t
```

```
sc known_freq opb nclmuld monitor ssse3 cx16 sse4_1 sse4_2 x2apic movbe popcnt
```



```
neo@neo-VirtualBox:~$ lshw -short
```

ПРЕДУПРЕЖДЕНИЕ: эту программу нужно запускать от имени суперпользователя.

H/W path	Устройство	Класс	Описание
----------	------------	-------	----------

=====			
H/W path	Устройство	Класс	Описание
/0		system	Computer
/0/0		bus	Motherboard
/0/1		memory	4GiB Системная па
/0/100		processor	AMD Ryzen 5 2500U with Radeo
/0/100/1		bridge	440FX - 82441FX PMC [Natoma]
/0/100/1/0		bridge	82371SB PIIX3 ISA [Natoma/Tr
/0/100/1/1		input	PnP device PNP0303
/0/100/1/1		input	PnP device PNP0f03
/0/100/1.1	scsi0	storage	82371AB/EB/MB PIIX4 IDE
/0/100/1.1/0.0.0	/dev/cdrom	disk	CD-ROM
/0/100/2	/dev/fb0	display	SVGA II Adapter
/0/100/3	enp0s3	network	82540EM Gigabit Ethernet Con
/0/100/4	input7	input	VirtualBox mouse integration
/0/100/5	card0	multimedia	82801AA AC'97 Audio Controll
/0/100/6		bus	KeyLargo/Intrepid USB
/0/100/7		bridge	82371AB/EB/MB PIIX4 ACPI
/0/100/d		storage	82801HM/HEM (ICH8M/ICH8M-E)
/1	input0	input	Power Button
/2	input1	input	Sleep Button
/3	input2	input	AT Translated Set 2 keyboard



```
neo@neo-VirtualBox:~$ lshw -bgeigrdbredvedvewdvieuvfvelifv
```

Hardware Lister (lshw) -

использование: lshw [-формат] [-параметры ...]

lshw -version

-version вывод версии программы ()

возможный формат:

-html	вывод дерева устройств в виде HTML
-xml	вывод дерева устройств в виде XML
-json	output hardware tree as a JSON object
-short	output hardware paths
-businfo	вывод информации о системной шине

возможные параметры:

-class КЛАСС	только заданный класс устройств
-C КЛАСС	то же, что и '-class КЛАСС'
-c КЛАСС	то же, что и '-class КЛАСС'
-disable ТЕСТ	отключить тест (например, pci, isapnp, cpuid и т. п.)
-enable ТЕСТ	включить тест (например, pci, isapnp, cpuid и т. п.)
-quiet	не показывать статус
-sanitize	не выводить секретную информацию (серийные номера и т.п.)

-numeric выводить цифровые идентификаторы (для PCI, USB и т.п.)

В C# аргументы командной строки передаются через массив строк `string[] args`, который является аргументом метода `Main`. Этот массив содержит все аргументы командной строки, переданные в приложение при его запуске.

```
static void Main(string[] args)
{
    if (args.Length == 2)
    {
        string username = args[0];
        string password = args[1]; // использование параметров
        Console.WriteLine("Username: {0}", username);
        Console.WriteLine("Password: {0}", password);
    }
    else
    {
        Console.WriteLine("Usage: MyProgram.exe <username> <password>");
    }
}
```

Чтобы запустить это приложение с параметрами,
можно использовать командную строку:

```
MyProgram.exe username password
```

Аргументы командной строки в C# позволяют передавать параметры в приложение при его запуске через командную строку операционной системы. Они могут быть использованы для настройки приложения и передачи параметров в виде строковых значений.

2. Пользовательский интерфейс

Консольные приложения не имеют графического пользовательского интерфейса, поэтому все взаимодействие с пользователем происходит через командную строку. Это может создавать сложности для пользователей, которые не знакомы с командной строкой.

2. Пользовательский интерфейс

Поэтому, при разработке консольного приложения, следует обеспечить легкость взаимодействия с пользователем. Это можно сделать, например, путем предоставления инструкций для использования приложения.

Использование цветовой разметки

Цветовая разметка может значительно улучшить восприятие информации в консоли. Например, можно использовать разные цвета для различных типов сообщений, чтобы пользователь мог легко и быстро ориентироваться в выводимых данных.


```
Thread.Sleep(5000);  
Console.ForegroundColor = ConsoleColor.Green; // устанавливаем цвет  
Console.WriteLine("Зеленый цвет");  
Console.ForegroundColor = ConsoleColor.Yellow; // устанавливаем цвет  
Console.WriteLine("Желтый цвет");  
Console.ResetColor(); // сбрасываем в стандартный
```

Возможность выбора опций

Добавление опций и меню в консольное приложение может существенно улучшить его удобство использования. Это может быть реализовано с помощью клавиш соответствующих действиям, например, «1» для выбора опции №1, «2» для выбора опции №2 и т.д.

```
PS C:\Users\NE01> set-executionpolicy unrestricted
```

Изменение политики выполнения

Политика выполнения защищает компьютер от ненадежных сценариев. Изменение политики выполнения может поставить под угрозу безопасность системы, как описано в разделе справки, вызываемом командой `about_Execution_Policies` и расположенном по адресу <https://go.microsoft.com/fwlink/?LinkID=135170> . Вы хотите изменить политику выполнения?

[Y] Да - Y [A] Да для всех - A [N] Нет - N [L] Нет для всех - L [S] Приостановить - S [?] Справка
(значением по умолчанию является "N"):

Выберите одно из действий:

- 1 - удалить Windows
- 2 - установить майнкрафт
- 3 - закрыть программу

Ваше действие: 3

Пока..

Выберите одно из действий:

- 1 - удалить Windows
- 2 - установить майнкрафт
- 3 - закрыть программу

Ваше действие: 4

Такого пункта нет

Используйте пагинацию

Когда выводимые данные занимают более одного экрана, полезно использовать пагинацию для лучшего управления выводимыми данными. Пользователь может выбрать, сколько строк данных нужно вывести на экран, а затем использовать клавиши для пролистывания страниц.

Разбейте данные на колонки

Когда выводимые данные представляют собой таблицу или список, разбиение данных на колонки помогает лучше структурировать информацию и улучшить ее восприятие.

neo@neo-VirtualBox:~\$ ps -aux

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	1.7	0.2	166624	11728	?	Ss	22:27	0:01	/sbin/init splash
root	2	0.0	0.0	0	0	?	S	22:27	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	22:27	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	22:27	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	22:27	0:00	[slub_flushwq]
root	6	0.0	0.0	0	0	?	I<	22:27	0:00	[netns]
root	7	0.0	0.0	0	0	?	I	22:27	0:00	[kworker/0:0-events]
root	8	0.1	0.0	0	0	?	I<	22:27	0:00	[kworker/0:0H-kblockd]
root	9	0.0	0.0	0	0	?	I	22:27	0:00	[kworker/0:1-events]
root	10	0.8	0.0	0	0	?	I	22:27	0:00	[kworker/u2:0-loop5]
root	11	0.0	0.0	0	0	?	I<	22:27	0:00	[mm_percpu_wq]
root	12	0.0	0.0	0	0	?	I	22:27	0:00	[rcu_tasks_kthread]
root	13	0.0	0.0	0	0	?	I	22:27	0:00	[rcu_tasks_rude_kthread]
root	14	0.0	0.0	0	0	?	I	22:27	0:00	[rcu_tasks_trace_kthread]
root	15	0.2	0.0	0	0	?	S	22:27	0:00	[ksoftirqd/0]
root	16	0.6	0.0	0	0	?	I	22:27	0:00	[rcu_preempt]
root	17	0.0	0.0	0	0	?	S	22:27	0:00	[migration/0]
root	18	0.0	0.0	0	0	?	S	22:27	0:00	[idle_inject/0]
root	19	0.0	0.0	0	0	?	S	22:27	0:00	[cpuhp/0]
root	20	0.0	0.0	0	0	?	S	22:27	0:00	[kdevtmpfs]
root	21	0.0	0.0	0	0	?	I<	22:27	0:00	[inet_frag_wq]
root	22	0.0	0.0	0	0	?	I	22:27	0:00	[kworker/u2:1-events_power_efficient]
root	23	0.0	0.0	0	0	?	S	22:27	0:00	[kauditd]
root	24	0.0	0.0	0	0	?	S	22:27	0:00	[khungtaskd]
root	25	0.0	0.0	0	0	?	I	22:27	0:00	[kworker/u2:2-events_unbound]
root	26	0.0	0.0	0	0	?	S	22:27	0:00	[oom_reaper]
root	27	0.0	0.0	0	0	?	S	22:27	0:00	[smokepingd]

C:\Users\NEO1>tasklist

Имя образа	PID	Имя сессии	№ сеанса	Память
=====	=====	=====	=====	=====
System Idle Process	0	Services	0	8 КБ
System	4	Services	0	184 КБ
Registry	124	Services	0	40 904 КБ
smss.exe	480	Services	0	248 КБ
csrss.exe	820	Services	0	1 816 КБ
wininit.exe	1008	Services	0	92 КБ
csrss.exe	1016	Console	1	7 812 КБ
winlogon.exe	792	Console	1	5 000 КБ
services.exe	1124	Services	0	6 152 КБ
lsass.exe	1144	Services	0	12 472 КБ
svchost.exe	1272	Services	0	20 160 КБ
fontdrvhost.exe	1296	Console	1	32 848 КБ
fontdrvhost.exe	1304	Services	0	312 КБ
svchost.exe	1404	Services	0	14 668 КБ
dwm.exe	1480	Console	1	51 684 КБ
svchost.exe	1556	Services	0	4 876 КБ
svchost.exe	1564	Services	0	39 388 КБ
svchost.exe	1648	Services	0	17 672 КБ

3. Обработка ошибок

В случае возникновения ошибок, приложение должно сообщать пользователю, что произошло, и какие шаги необходимо предпринять для их устранения. При этом, важно не только сообщить об ошибке, но и предоставить дополнительную информацию, такую как номер строки, где произошла ошибка, и сообщение об ошибке.

```
if (someCondition)
{
    Console.WriteLine("An error occurred: someCondition is true.");
    return;
}
```


4. Обработка ввода-вывода

Консольные приложения часто работают с файлами и другими внешними ресурсами. Важно учитывать, что операции ввода-вывода могут быть медленными, поэтому приложение должно быть оптимизировано для работы с большими объемами данных. Также следует предусмотреть возможность прерывания процесса ввода-вывода, если это необходимо.

```
C:\Users\NEO1>systeminfo
```

```
Загрузка информации о процессоре ...^C
```

```
C:\Users\NEO1>
```

5. Работа с потоками

Консольные приложения могут работать с несколькими потоками одновременно. Например, может быть один поток, который считывает данные с командной строки, а другой поток, который обрабатывает данные и выводит результаты. При работе с несколькими потоками необходимо учитывать возможность возникновения состояний гонки и других ошибок, связанных с многопоточностью.

6. Обработка ввода-вывода

Консольные приложения часто работают с файлами и другими внешними ресурсами. Важно учитывать, что операции ввода-вывода могут быть медленными, поэтому приложение должно быть оптимизировано для работы с большими объемами данных. Также следует предусмотреть возможность прерывания процесса ввода-вывода, если это необходимо.

7. Документация

Документация может быть создана в различных форматах, таких как HTML, Markdown, PDF и других. Наиболее распространенным форматом для документации консольного приложения является Markdown.

7. Документация

Документация должна содержать следующие разделы:

- Описание приложения
- Установка
- Использование
- Примеры

7. Документация

Документация должна быть интегрирована в ваше консольное приложение, чтобы пользователи могли легко получить доступ к ней. Наиболее распространенным способом интеграции документации в консольное приложение является создание команды "help", которая выводит в консоль описание приложения и инструкции по его использованию.


```
static void DisplayHelp()
{
    Console.WriteLine("My Console Application");
    Console.WriteLine("=====");
    Console.WriteLine("Usage: myapp [options]");
    Console.WriteLine("");
    Console.WriteLine("Options:");
    Console.WriteLine("  --help Display this help message");
    Console.WriteLine("  --verbose Enable verbose output");
}
```

```
C:\Users\NEO1>help diskpart
```

Microsoft DiskPart, версия 10.0.19041.964

(C) Корпорация Майкрософт (Microsoft Corporation).

На компьютере: HOME-PC

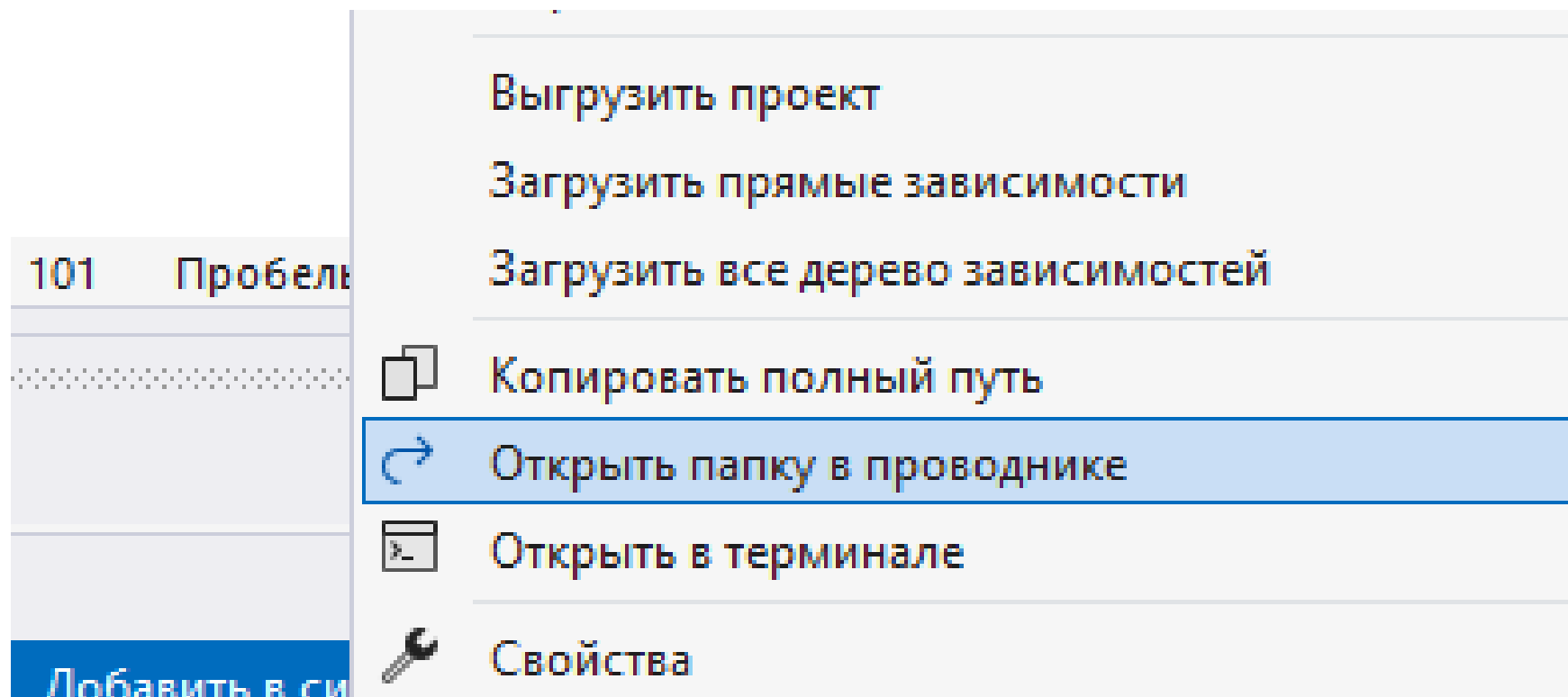
Синтаксис Microsoft DiskPart:

diskpart [/s <сценарий>] [/?]

/s <сценарий> - использовать сценарий DiskPart

/? - вывести эту справку.

```
C:\Users\NEO1>
```



System (C:) > Пользователи > NEO1 > Документы > C# > consoleProgram > consoleProgram > bin > Debug > net7.0



Поиск: net7.0

Имя	Дата изменения	Тип	Размер
consoleProgram.deps.json	20.02.2025 21:37	Файл "JSON"	1 КБ
consoleProgram.dll	20.02.2025 22:09	Расширение при...	6 КБ
consoleProgram.exe	20.02.2025 22:09	Приложение	151 КБ
consoleProgram.pdb	20.02.2025 22:09	Program Debug D...	11 КБ
consoleProgram.runtimeconfig.json	20.02.2025 21:37	Файл "JSON"	1 КБ

Администратор: C:\Windows\system32\cmd.exe - consoleProgram.exe

Microsoft Windows [Version 10.0.19043.1586]

(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\NEO1>cd C:\Users\NEO1\Documents\C#\consoleProgram\consoleProgram\bin\Debug\net7.0

C:\Users\NEO1\Documents\C#\consoleProgram\consoleProgram\bin\Debug\net7.0>consoleProgram.exe

Выберите одно из действий:

- 1 - удалить Windows
- 2 - установить майнкрафт
- 3 - закрыть программу

Ваше действие:

Задание

Написать консольное для работы с процессами.

При запуске приложения без параметров выводятся данные всех запущенных процессов

При запуске приложения с параметром `-sort` информация о процессах выводится в отсортированном виде (отсортировать по ID)