

# **Запуск процессов. Работа с реестром и файловой системой**

# Что такое реестр

Реестр в операционной системе Windows - это иерархическая база данных, которая хранит настройки и конфигурационные данные операционной системы, установленных на ней приложений и компонентов.

Реестр представляет собой иерархию ключей и подключей, которые можно рассматривать как папки и подпапки в файловой системе.

# Зачем он нужен

Реестр содержит сведения, на которые Windows постоянно ссылается во время операции, такие как профили для каждого пользователя, приложения, установленные на компьютере, типы документов, которые могут создаваться, параметры таблицы свойств для папок и значков приложений, оборудование, которое установлено в системе, и используемые порты.

# Ветви реестра

- `HKEY_LOCAL_MACHINE`
- `HKEY_CURRENT_USER`
- `HKEY_CLASSES_ROOT`
- `HKEY_USERS`
- `HKEY_CURRENT_CONFIG`







# Ключи


Параметры или ключи реестра имеют имена, представленные в обычном текстовом виде и значения, которые хранятся в виде стандартизированных записей определенного типа

# Работа с реестром

Для просмотра реестра мы можем воспользоваться утилитой regedit. При запуске данная утилита отображает в графическом окне в древовидном виде все содержимое реестра:

Компьютер\HKEY\_LOCAL\_MACHINE

- ▼  Компьютер
- >  HKEY\_CLASSES\_ROOT
- >  HKEY\_CURRENT\_USER
- >  HKEY\_LOCAL\_MACHINE
- >  HKEY\_USERS
- >  HKEY\_CURRENT\_CONFIG

Имя	Тип	Значение
 (По умолчанию)	REG_SZ	(значение не присвоено)

# Работа с реестром на C#

Для управления регистром в пространстве имен `Microsoft.Win32` имеются два класса: `Registry` и `RegistryKey`.

Класс `Registry` позволяет получить доступ к ключам верхнего уровня реестра. А класс `RegistryKey` представляет отдельный ключ реестра.



# Registry

Класс Registry содержит статические свойства, каждое из которых представляет соответствующий ключ верхнего уровня:

- Registry.ClassesRoot
- Registry.CurrentConfig
- Registry.CurrentUser

# Registry

- Registry.DynData
- Registry.Users
- Registry.PerformanceData

# RegistryKey

Для управления ключами в реестре класс RegistryKey определяет ряд свойств и методов. Основные из них:

- Name: возвращает имя ключа реестра
- Close(): закрывает ключ
- CreateSubKey(): создает вложенный ключ, если он не существует
- DeleteSubKey(): удаляет вложенный ключ

# RegistryKey

- `DeleteValue()`: удаляет значение ключа
- `GetSubKeyNames()`: возвращает коллекцию имен вложенных ключей
- `GetValue()`: возвращает значение ключа
- `OpenSubKey()`: открывает вложенный ключ
- `SetValue()`: устанавливает значение ключа

# Пример

```
RegistryKey currentUserKey = Registry.CurrentUser;  
RegistryKey helloKey =  
currentUserKey.CreateSubKey("HelloKey");  
helloKey.SetValue("login", "admin");  
helloKey.SetValue("password", "12345");  
helloKey.Close();
```

# Пример

```
RegistryKey currentUserKey = Registry.CurrentUser;  
RegistryKey helloKey = currentUserKey.OpenSubKey("HelloKey",  
true);  
RegistryKey subHelloKey = helloKey.CreateSubKey("SubHelloKey");  
subHelloKey.SetValue("val", "23");  
subHelloKey.Close();  
helloKey.Close();
```

# Пример

```
RegistryKey currentUserKey = Registry.CurrentUser;  
RegistryKey helloKey = currentUserKey.OpenSubKey("HelloKey");  
string login = helloKey.GetValue("login").ToString();  
string password = helloKey.GetValue("password").ToString();  
helloKey.Close();  
Console.WriteLine(login);  
Console.WriteLine(password);
```

# Пример

```
RegistryKey currentUserKey = Registry.CurrentUser;  
RegistryKey helloKey = currentUserKey.OpenSubKey("HelloKey", true);  
helloKey.DeleteSubKey("SubHelloKey"); // удаляем вложенный  
ключ  
helloKey.DeleteValue("login"); // удаляем значение из ключа  
helloKey.Close();  
currentUserKey.DeleteSubKey("HelloKey"); // удаляем сам ключ
```



# Задание

Добавить блокнот в автозагрузку. Открыть раздел HKLM\Software\Microsoft\Windows\CurrentVersion. Найти подраздел Run. В этом разделе есть строковые ключи , отвечающие за запуск программ. Название ключа может быть произвольным, а в качестве значения у них указывается запускаемая программа, если надо - то с параметрами.

# Работа с процессами

В .NET процесс представлен классом `Process` из пространства имен `System.Diagnostics`. Этот класс позволяет управлять уже запущенными процессами, а также запускать новые.

# Работа с процессами

## Свойства класса Process:

- **Handle:** возвращает дескриптор процесса
- **Свойство Id:** получает ID процесса
- **Свойство MachineName:** возвращает имя компьютера, на котором запущен процесс
- **Свойство ProcessName:** возвращает имя процесса
- **StartTime:** время запуска процесса

# Работа с процессами

Методы класса Process:

- `CloseMainWindow()`: закрывает окно процесса
- `GetProcesses()`: возвращает массив всех запущенных процессов
- `GetProcessesByName()`: возвращает процессы по его имени.

# Работа с процессами

Методы класса Process:

- `GetProcessById()`: возвращает процесс по Id.
- `Kill()`: останавливает процесс
- `Start()`: запускает новый процесс

# Пример

```
var proc = Process.GetCurrentProcess();  
Console.WriteLine($"Id: {proc.Id}");  
Console.WriteLine($"Name: {proc.ProcessName}");  
Console.WriteLine($"Memory: {proc. PageMemorySize64}");
```

# Пример

```
foreach (Process process in Process.GetProcesses()){  
    // выводим id и имя процесса  
    Console.WriteLine($"ID: {process.Id} Name: {process.ProcessName}");  
}
```

# Пример

```
Process.Start(@"C:\Program Files\Google\Chrome\Application\chrome.exe");
```

```
Process.Start("notepad.exe");
```



# Пример

```
ProcessStartInfo procInfo = new ProcessStartInfo();  
// исполняемый файл программы - браузер хром  
procInfo.FileName = @"C:\Program Files\Google\Chrome\  
Application\chrome.exe";  
// аргументы запуска - адрес интернет-ресурса  
procInfo.Arguments = "https://arcotel.ru/";  
Process.Start(procInfo);
```

# Выполнение консольной команды

```
string cmd = "calc.exe";  
var proc = new ProcessStartInfo(){  
    UseShellExecute = true,  
    FileName = @"C:\Windows\System32\cmd.exe",  
    Arguments = "/c " + cmd,  
    WindowStyle = ProcessWindowStyle.Hidden  
};  
Process.Start(proc);
```

# Задание

- Запустить калькулятор (calc.exe)
- Найти процесс, соответствующий запущенному калькулятору и вывести информацию о нём