

A laptop sits on a wooden desk. To its left is a potted plant with green, succulent-like leaves in a light-colored, textured pot. The laptop screen displays a code editor with a dark theme. The editor shows a file explorer on the left with a project named 'APIBitcoin-ticker'. The main area contains JavaScript code for an Express.js application. The code includes a GET endpoint for the root, a POST endpoint for receiving crypto and fiat values, and a function to fetch data from an API. The background is slightly blurred, showing a window with natural light.

# Обработка ошибок. Работа с процессами

Введение в C#

# Обработка исключительных ситуаций

1. Блок `try-catch`.
2. Оператор `finally`.
3. Стандартные исключения  
.NET.

# Блок try-catch

```
1 try
2 {
3     int x = 10;
4     int y = x / 0;
5 }
6 catch
7 {
8     Console.WriteLine("Во время выполнения деления произошла ошибка");
9 }
10 Console.WriteLine("Продолжение выполнения программы");
```

# Оператор `finally`

```
1 string log = string.Empty; // пустая строка - ""
2 log += $"Начало работы приложения";
3 try
4 {
5     int x = 10;
6     int y = x / 0;
7     log += $"{Environment.NewLine}Результат деления: {y}";
8 }
9 catch
10 {
11     log += $"{Environment.NewLine}Во время выполнения деления произошла ошибка";
12 }
13 finally
14 {
15     File.WriteAllText("log.txt", log);
16 }
```

# Конструкция catch

```
1 try
2 {
3     // ...
4 }
5 catch (AddressException ex) when (ex.Code == ErrorCodes.NoAddress)
6 {
7
8     Console.WriteLine("Здание не содержит адреса");
9 }
10 catch (AddressException ex) when (ex.Code == ErrorCodes.NoBuilding)
11 {
12
13     Console.WriteLine("Адрес не содержит номер дома");
14 }
```

# Процессы

1. Определение процесса.
2. Входные аргументы процесса.
3. Коды возврата.

**Приложение, загруженное в память компьютера, называется процессом.**



# Входные аргументы процесса

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine(args.Length); // 0
4 }
```



# Передача аргументов в процесс

1. Командная строка
2. Visual Studio

# Коды возврата

```
1 static int Main(string[] args)
2 {
3     return 22;
4 }
```

**Чтение стандартного потока вывода**

**Запуск файла в ассоциированной  
программе**

# Получение списка процессов