



Microsoft Partner
Silver Learning

C# Стартовый

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Технический смысл переменной



ITVVDN
IT VIDEO DEVELOPERS NETWORK

Introduction



Александр Шевчук

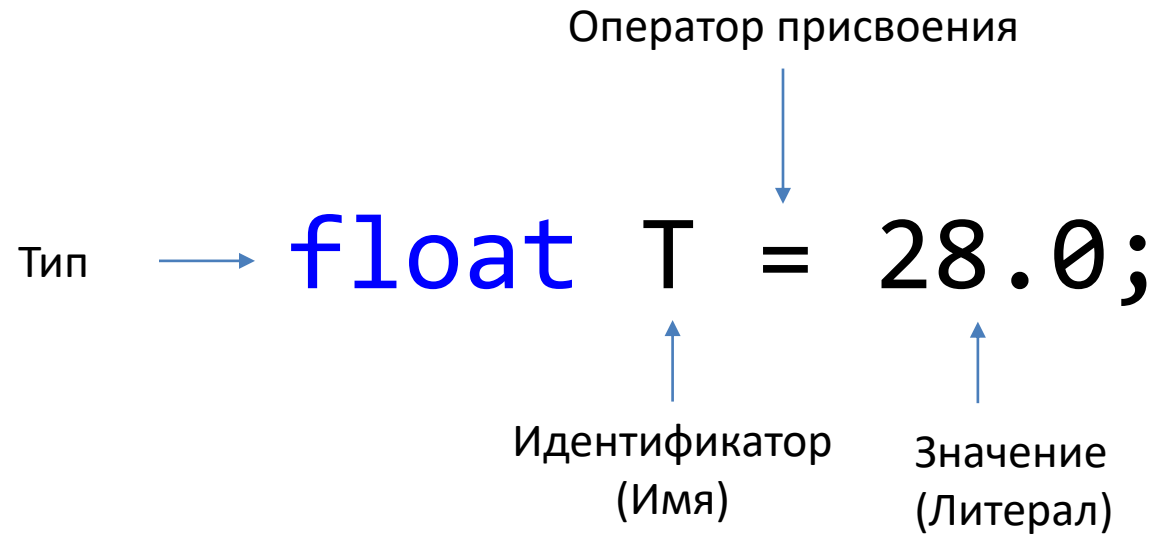


MCID: 9230440

Тема урока

Технический смысл переменной

Переменная

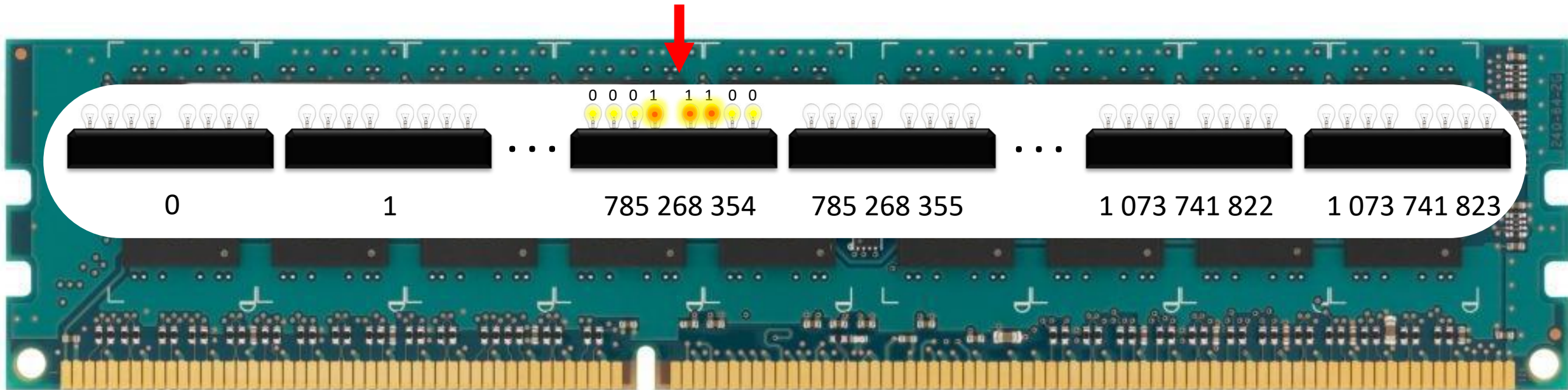


Переменная — это конструкция языка программирования, которая хранит в себе некоторое значение, которое можно изменять в ходе выполнения программы.

Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит в себе некоторое значение, которое можно изменить.

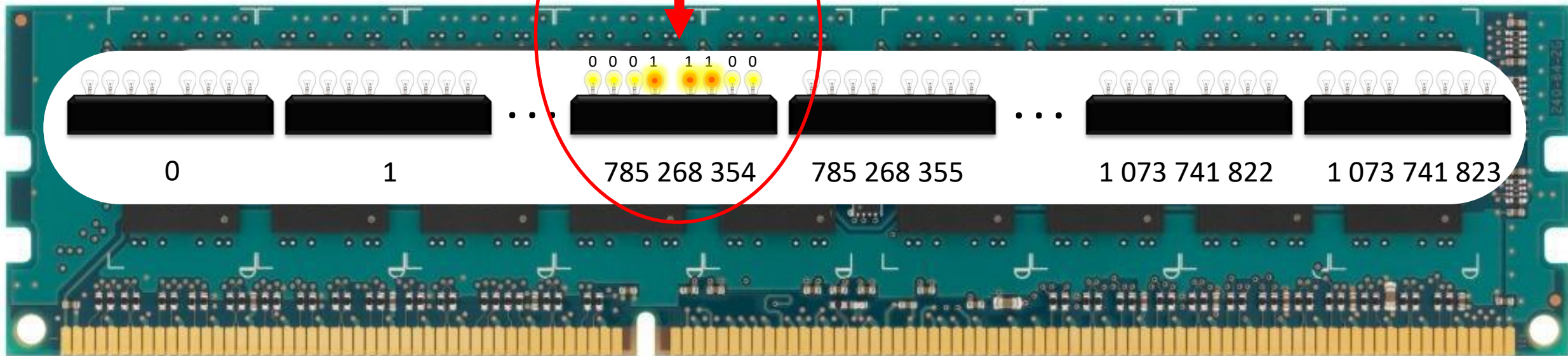
```
byte temperature = 28; // 11100 b = 1C hex
```



Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит в себе некоторое значение, которое можно изменить.

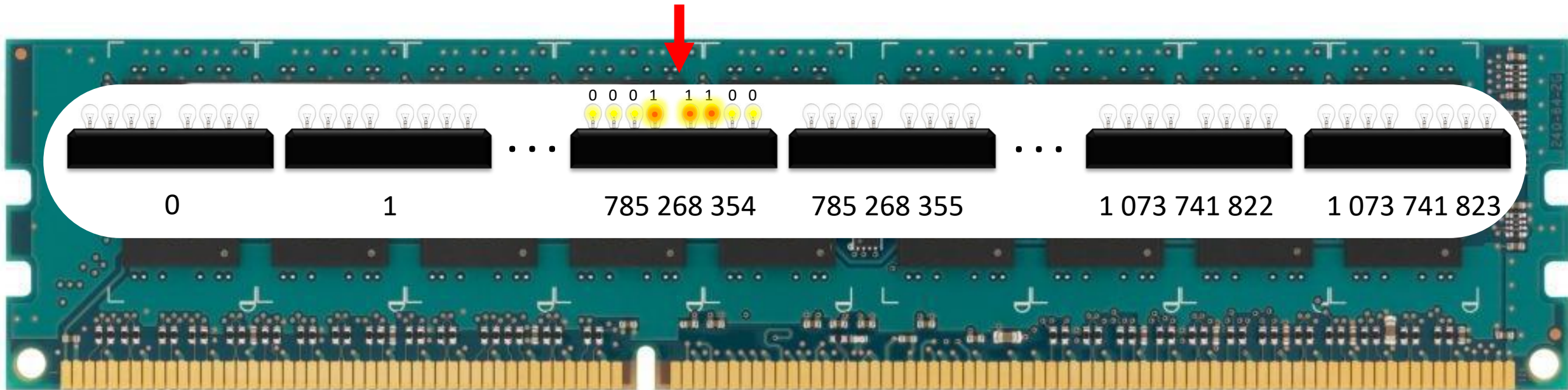
```
byte [785268354] = 28; // 11100 b = 1C hex
```



Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит в себе некоторое значение, которое можно изменить.

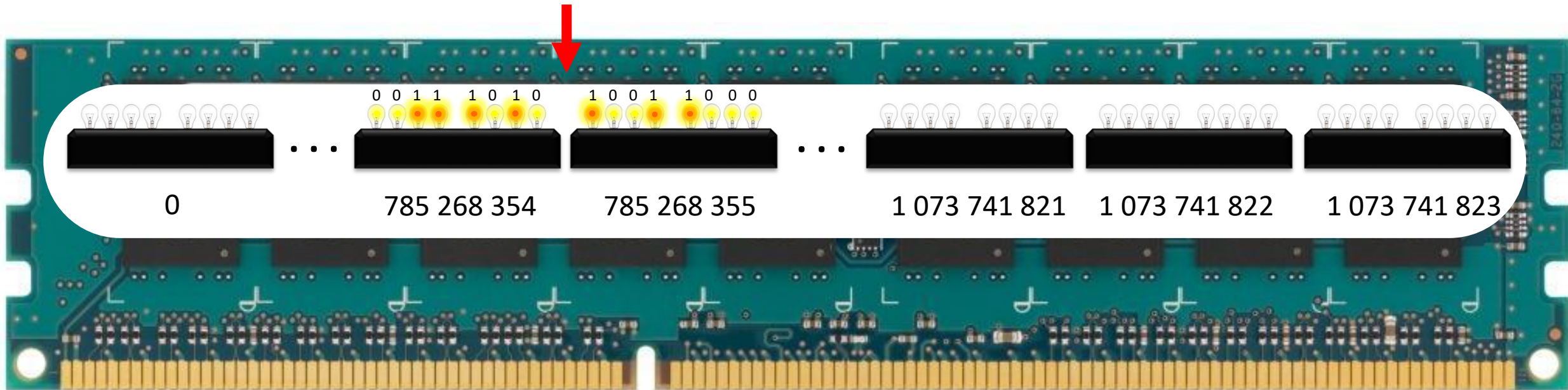
```
byte temperature = 28; // 11100 b = 1C hex
```



Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит в себе некоторое значение, которое можно изменить.

```
ushort salary = 15000; // $ 11101010011000 b = 3A98 hex
```

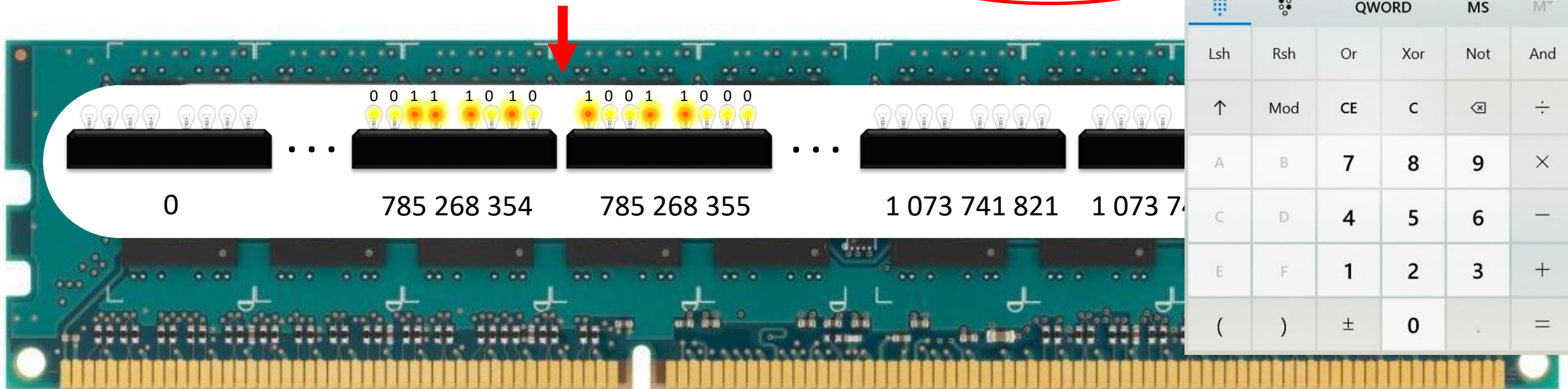


ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит некоторое значение, которое можно изменить.

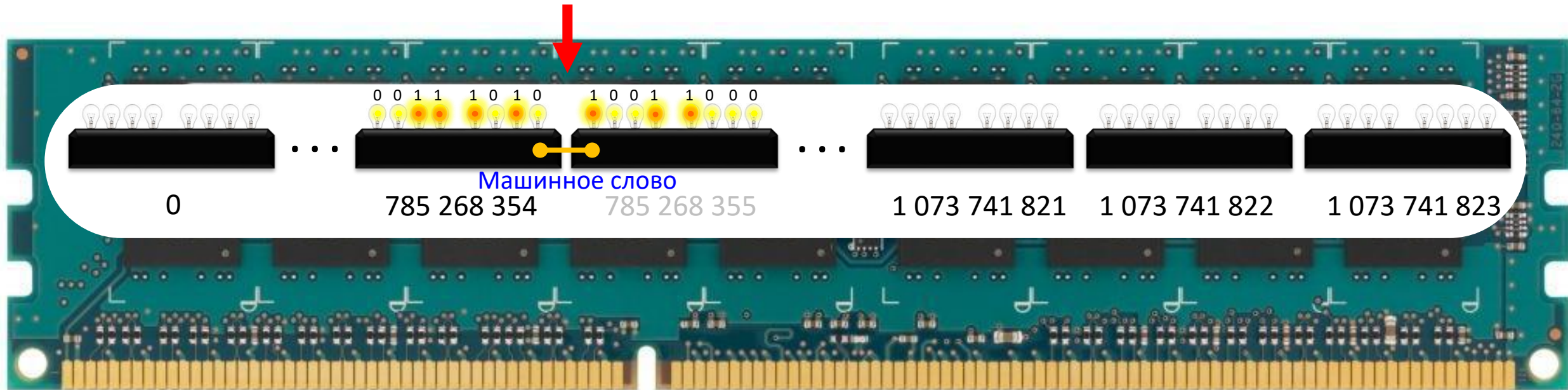
`ushort salary = 15000; // $ 11101010011000`



Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит в себе некоторое значение, которое можно изменить.

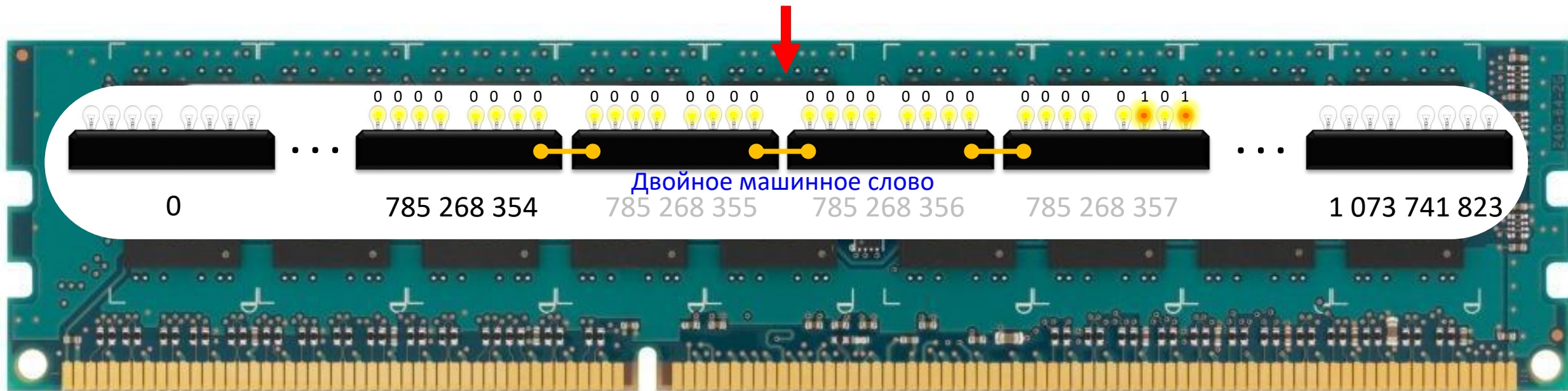
```
ushort salary = 15000; // $ 11101010011000 b = 3A98 hex
```



Переменная в памяти

Переменная – это ячейка, или область оперативной памяти, которая хранит в себе некоторое значение, которое можно изменить.

```
int apples = 5; // 101 b = 5 hex
```



ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Задача: В вазе лежало 5 яблок и 3 груши.
Сколько всего фруктов лежало в вазе?

Дано:

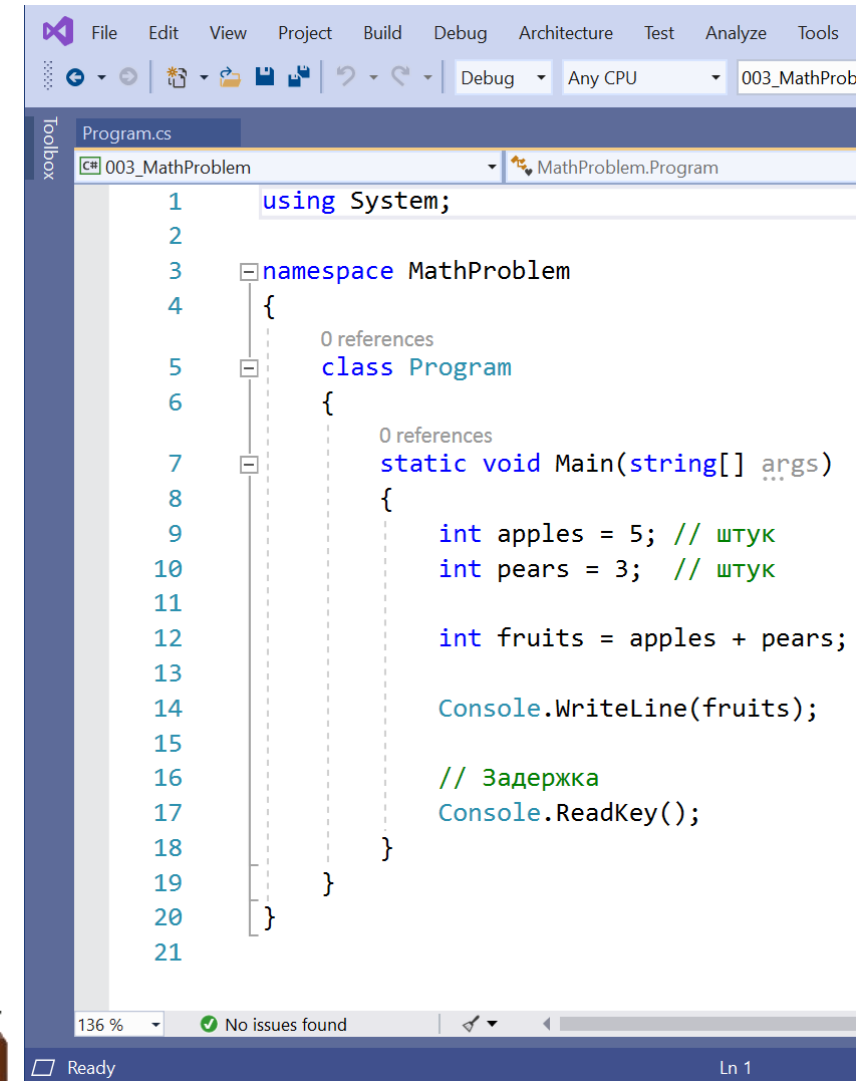
`int` яблоки = 5
`int` груши = 3

`int` фрукты = ?

Решение:

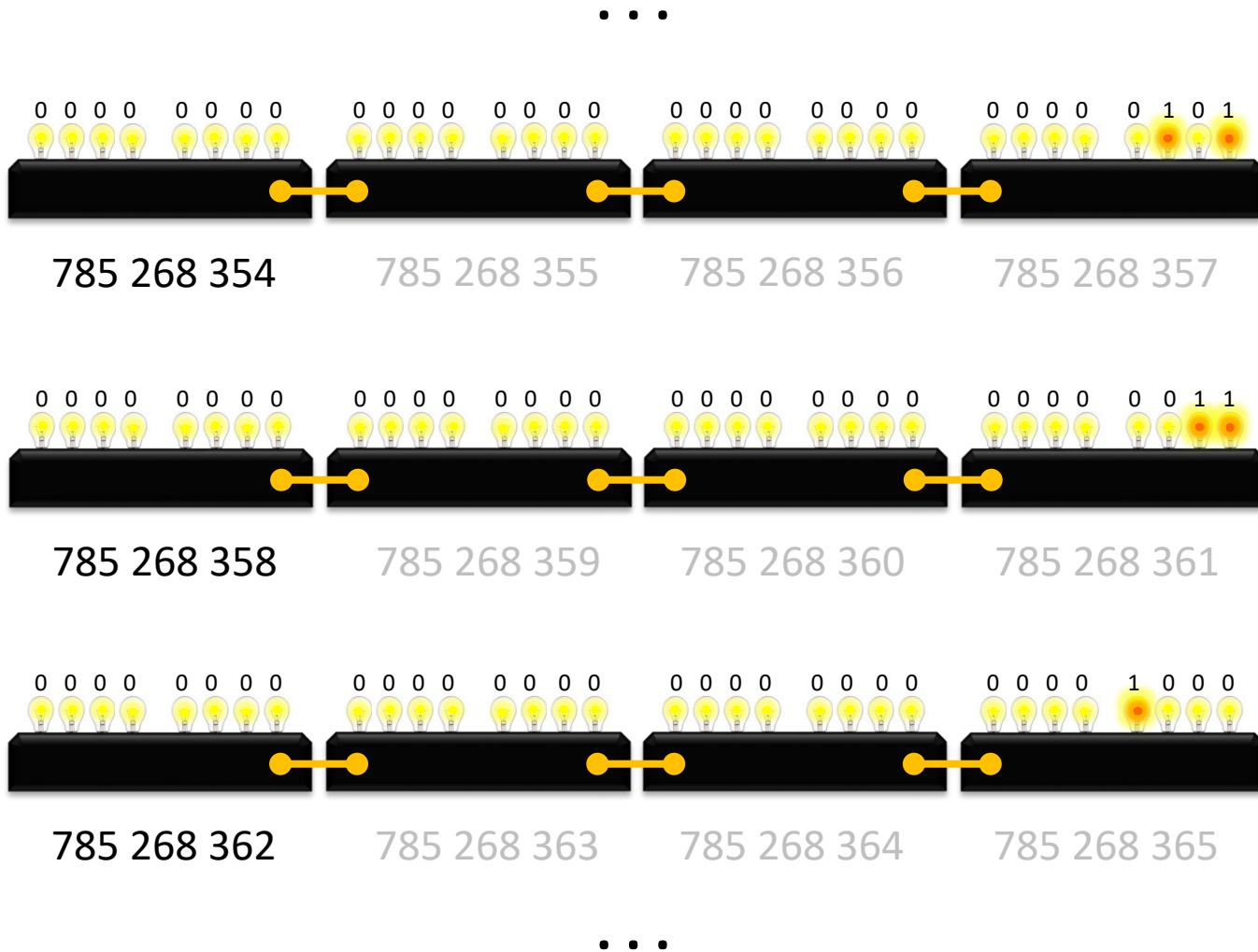
фрукты = яблоки + груши

фрукты = 5 + 3 = 8



```
1  using System;
2
3  namespace MathProblem
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             int apples = 5; // штук
12             int pears = 3; // штук
13
14             int fruits = apples + pears;
15
16             Console.WriteLine(fruits);
17
18             // Задержка
19             Console.ReadKey();
20         }
21     }
```


ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#



The screenshot shows the Visual Studio IDE with a C# project named '003_MathProblem'. The code is as follows:

```
1 using System;
2
3 namespace MathProblem
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            int apples = 5; // штук
12            int pears = 3; // штук
13            int fruits = apples + pears;
14            Console.WriteLine(fruits);
15
16            // Задержка
17            Console.ReadKey();
18        }
19    }
20 }
21
```

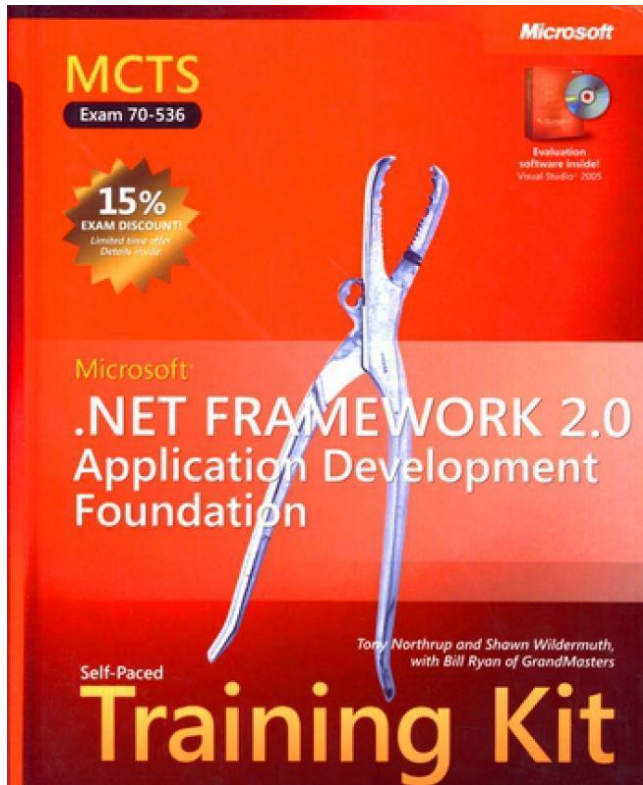
Red lines highlight the following parts of the code:

- Line 9: `int apples = 5; // штук`
- Line 10: `int pears = 3; // штук`
- Line 13: `int fruits = apples + pears;`

The status bar at the bottom indicates 'Ready' and 'No issues found'.

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

СОВЕТ: Оптимизация производительности встроенных типов



Учебный комплект MCTS для самостоятельного обучения (экзамен 70-536): Microsoft®.NET Framework 2.0-Foundation Development Foundation

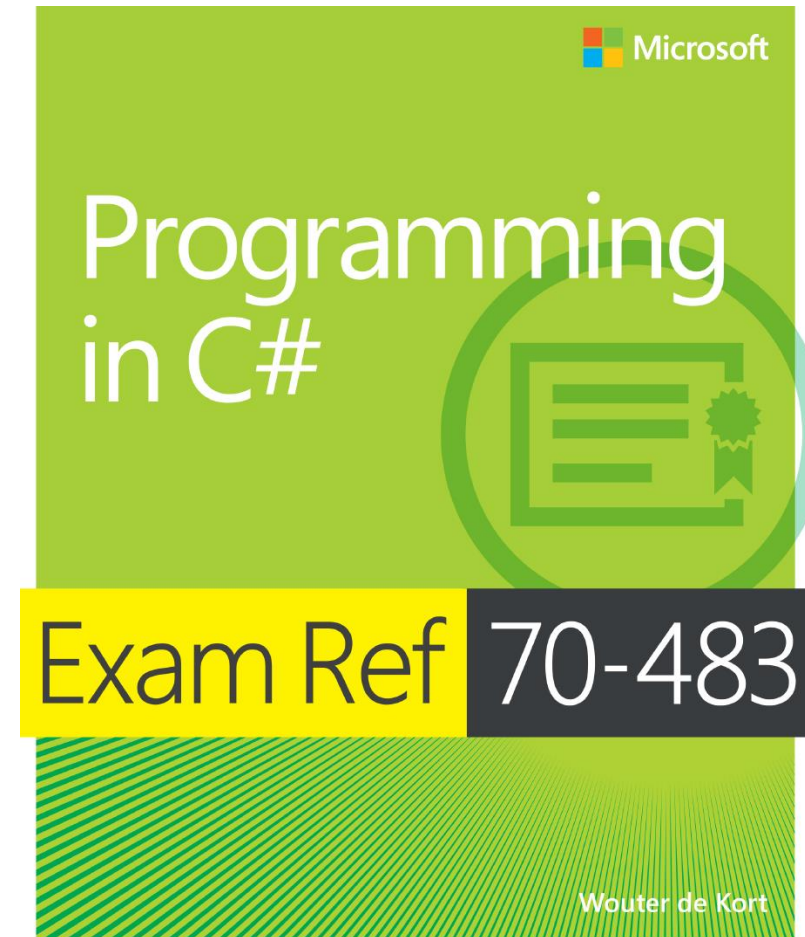
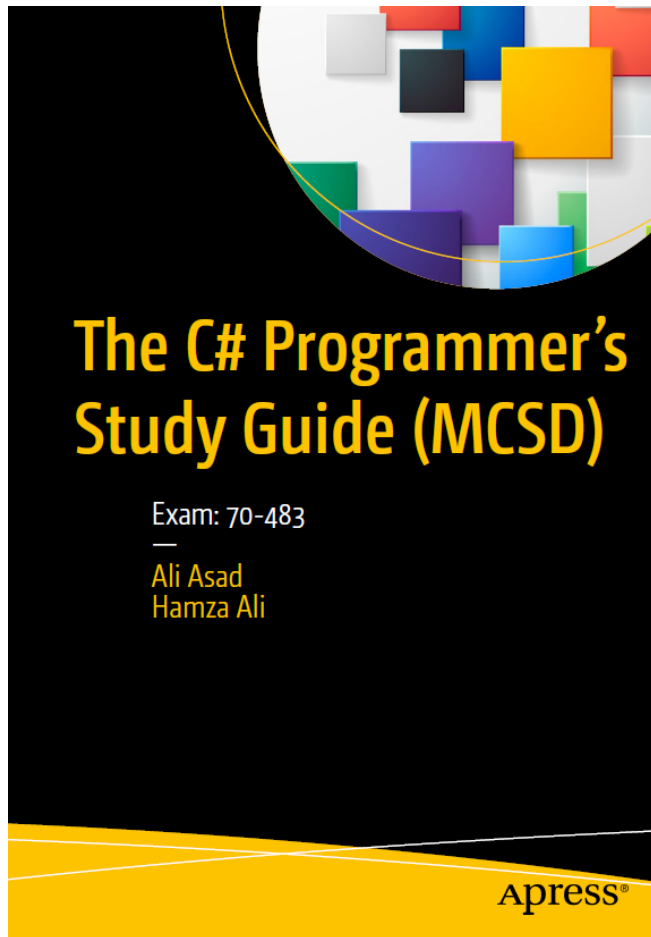
Рабочая среда оптимизирует производительность 32-разрядных целых типов (*Int32* и *UInt32*), поэтому используйте эти типы для счетчиков и других часто используемых целочисленных переменных. Для операций с плавающей запятой, тип *Double* является наиболее эффективным типом, поскольку эти операции оптимизируются с помощью аппаратного обеспечения.

Chapter 1. Framework Fundamentals. Lesson 1: Using Value Types. Built-in Value Types

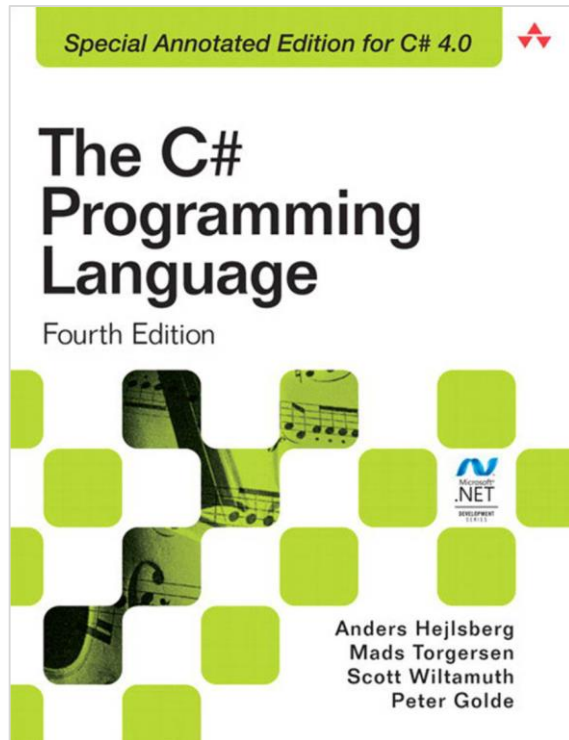
BEST PRACTICES Optimizing performance with built-in types

The runtime optimizes the performance of 32-bit integer types (*Int32* and *UInt32*), so use those types for counters and other frequently accessed integral variables. For floating-point operations, *Double* is the most efficient type because those operations are optimized by hardware.

Exam 70-483



ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#



4.1.5 Integrals Types

Page 145

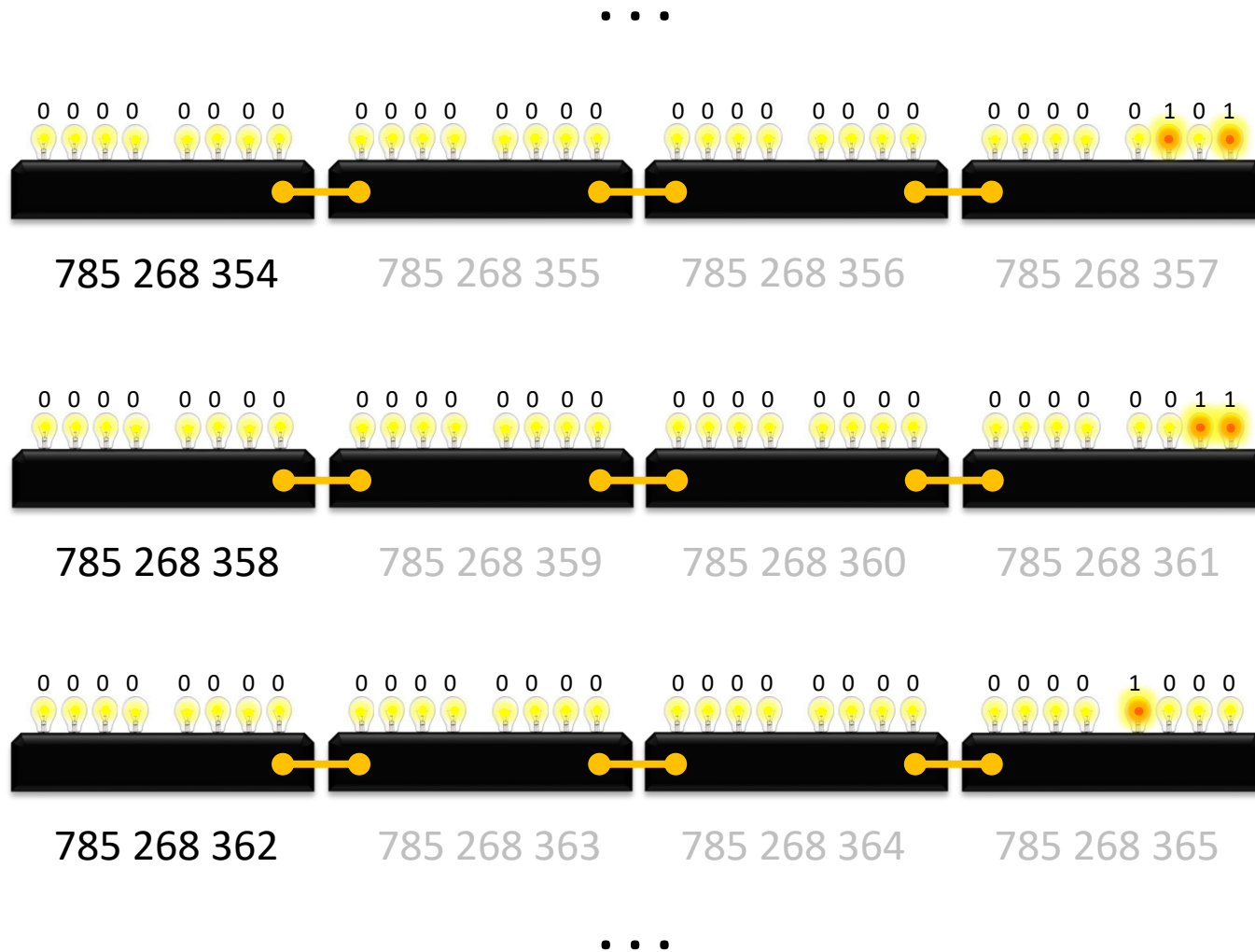
JESSE LIBERTY I have to confess that with the power of modern PCs, and the greater cost of programmer time relative to the cost of memory, I tend to use int for just about any integral (nonfractional) value and double for any fractional value. All the rest, I pretty much ignore.

4.1.5 Целочисленные типы

Стр. 157

ДЖЕСС ЛИБЕРТИ Должен признаться, что, учитывая мощь современных персональных компьютеров и то, что время программиста стоит дороже, чем память компьютера, я склоняюсь к тому, чтобы использовать int почти для всех целочисленных (не вещественных) значений и double для любых вещественных значений. Все остальное я чаще всего игнорирую.

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#



```
1 using System;
2
3 namespace MathProblem
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            int apples = 5; // штук
12            int pears = 3; // штук
13
14            int fruits = apples + pears;
15
16            Console.WriteLine(fruits);
17
18            // Задержка
19            Console.ReadKey();
20        }
21    }
```

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Варианты представления числовых переменных в ОЗУ

1 байт = 8 бит

 byte | sbyte

2 байта = 16 бит (Машинное слово)

 short | ushort

4 байта = 32 бита (Двойное машинное слово)

 int | uint | float

8 байт = 64 бита (Учетверённое машинное слово)

 long | ulong | double

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Спасибо за внимание! До новых встреч!



Александр Шевчук



OLEKSANDR SHEVCHUK

Has successfully completed the requirements to be recognized as a Trainer.

Date of achievement: October 25, 2012
Certification number: E207-8382
Valid until: April 04, 2019

Satya Nadella
Chief Executive Officer

Microsoft
CERTIFIED
Trainer

MCID: 9230440

Информационный видеоресурс для разработчиков программного обеспечения

