

Числови примитиви в C#

упражнение

1. Целочислен тип - **int**

А) Характеристики

- най-често използван за съхраняване на цели числа със знак
- множество на допустимите стойности е в диапазона
от **-2,147,483,648** до **2,147,483,647**
- размерът който заема в паметта е 4 байта (32 бита)

Б) Преобразуване на низ в цяло число

- в графично приложение:

```
int side = int.Parse(textBox1.Text);
```

- в конзолно приложение:

```
int side = int.Parse(Console.ReadLine());
```

В) Математически операции с целочислени данни

- събиране (+)**
- изваждане (-)**
- умножение (*)**
- делене (/)**
- намиране на остатък при делене (%)**

Г) Трикове

- **num % 10** връща последната цифра на числото **num**
- **num / 10** връща числото **num** без последната му цифра
- **num % 2** връща нула, ако числото **num** е четно

Задача 1:

Създайте конзолна програма с име **DigitsPlus**, която чете цяло двуцифрено число и отпечатва сборът на цифрите му.

ВХОД: **35**

ИЗХОД: **8**

ВХОД: **77**

ИЗХОД: **14**

ВХОД: **60**

ИЗХОД: **6**

Задача 2:

Създайте конзолна програма с име **ReverseNum**, която по въведено цяло двуцифрено число, да отпечата числото, записано със същите цифри, но в обратен ред.

ВХОД: **35**

ИЗХОД: **53**

ВХОД: **77**

ИЗХОД: **77**

ВХОД: **60**

ИЗХОД: **06**

Задача 3:

Създайте графично приложение с име **ReverseNum2**, която по въведено цяло трицифрено число, да отпечата числото, записано със същите цифри, но в обратен ред.

ВХОД: **356**

ВХОД: **566**

ВХОД: **274**

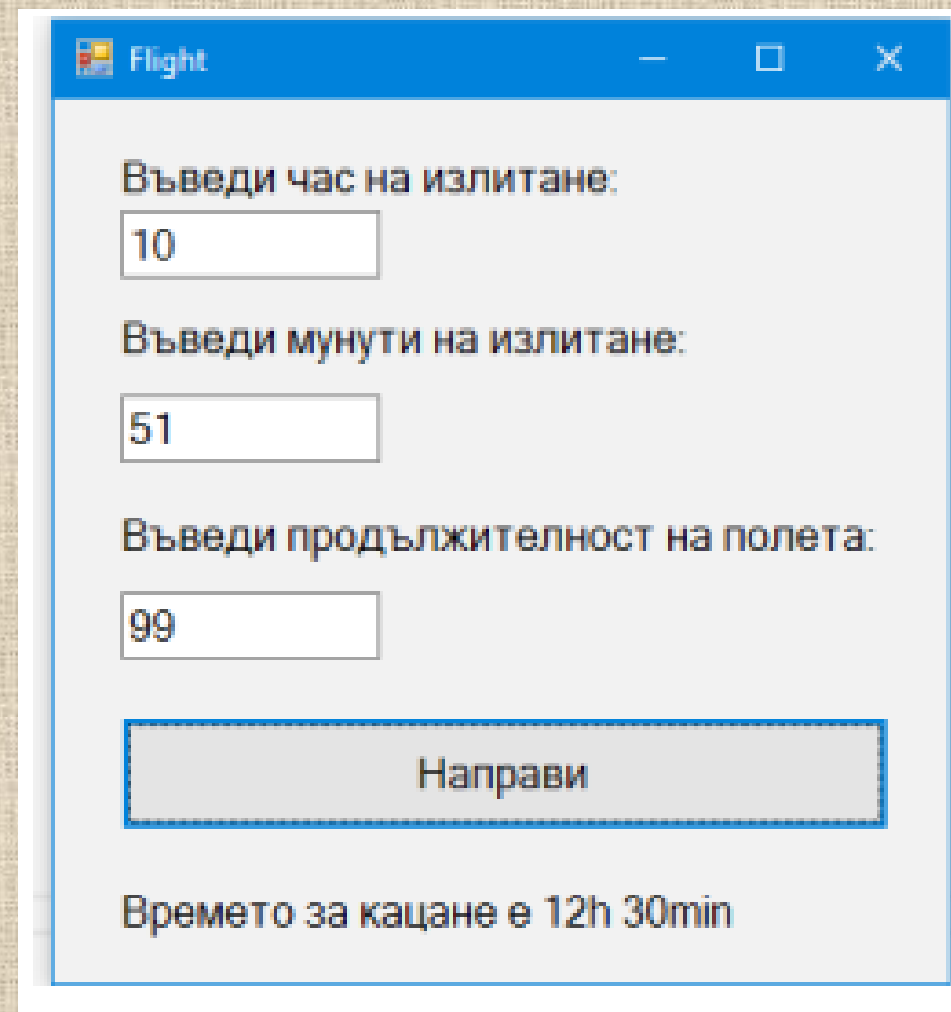
ИЗХОД: **653**

ИЗХОД: **665**

ИЗХОД: **472**

Задача 4:

Направете графично приложение с име **Flight**, в което се въвеждат часа и минутите от началото на часа на излитането на самолет, както и продължителността на полета му в минути. Програмата трябва да изведе часа и минутите, когато самолетът трябва да кацне, при условие, че кацането става в деня на излитане.



The screenshot shows a Windows application window titled "Flight". It contains three input fields for flight data and a "Направи" (Make) button. Below the button, it displays the calculated landing time.

Input Label	Input Value
Въведи час на излитане:	10
Въведи минути на излитане:	51
Въведи продължителност на полета:	99

Направи

Времето за кацане е 12h 30min

2. Реален тип - **double**

А) Характеристики

- най-често използван за съхраняване на дробни числа
- множество на допустимите стойности е в диапазона
от $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10^{308}$
- размерът който заема в паметта е 8 байта (64 бита)

Б) Преобразуване на низ в дробно число

- в графично приложение:

```
double num = double.Parse(textBox1.Text);
```

- в конзолно приложение:

```
double num = double.Parse(Console.ReadLine());
```

В) Математически операции с реални данни

- събиране (+)
- изваждане (−)
- умножение (*)
- делене (/)
- намиране на остатък при делене (%)!!!

Задачи, които демонстрират разликите и спецификите на типовете `float`, `double` и `decimal` в C#

Създайте конзолно приложение с име **RealNumbers, в което се декларират променливи от трите реални типа, на които се присвоява едно и също голямо число. След това в конзолата се отпечатват стойностите им. Анализирайте резултата.**

```
float aFloat = 452637E23f;  
double aDouble = 452637E23;  
decimal aDecimal = 452637E23m;
```

```
Console.WriteLine("float number: " + aFloat);  
Console.WriteLine("double number: " + aDouble);  
Console.WriteLine("decimal number: " + aDecimal);
```

Създайте конзолно приложение с име **Area, в което използвайки различни реални типове променливи за радиус да се изчисли площта на окръжност.**

```
float radiusFloat = 3.5f;  
double radiusDouble = 3.5;  
decimal radiusDecimal = 3.5m;
```

```
float areaFloat = (float)(Math.PI * radiusFloat * radiusFloat);  
double areaDouble = Math.PI * radiusDouble * radiusDouble;  
decimal areaDecimal = (decimal)Math.PI * radiusDecimal *  
radiusDecimal;
```

```
Console.WriteLine("Площта с float:" + areaFloat);  
Console.WriteLine("Площта с double:" + areaDouble);  
Console.WriteLine("Площта с decimal:" + areaDecimal);
```

Създайте конзолна програма **VAT, която изчислява стойността на даден продукт след данъчно облагане (VAT 20%).**

```
float priceFloat = 99.9999999999f; //точност до 7-8 десетични знака  
double priceDouble = 99.999999999999; //точност до 15-16 десетични знака  
decimal priceDecimal = 99.99999999999999999999m; //точност до 28-29 десетични знака
```

```
float vatFloat = priceFloat * 0.2f;  
double vatDouble = priceDouble * 0.2;  
decimal vatDecimal = priceDecimal * 0.2m;
```

```
Console.WriteLine("VAT c float: "+vatFloat);  
Console.WriteLine("VAT c double: "+vatDouble);  
Console.WriteLine("VAT c decimal: "+vatDecimal);
```

Задача 1:

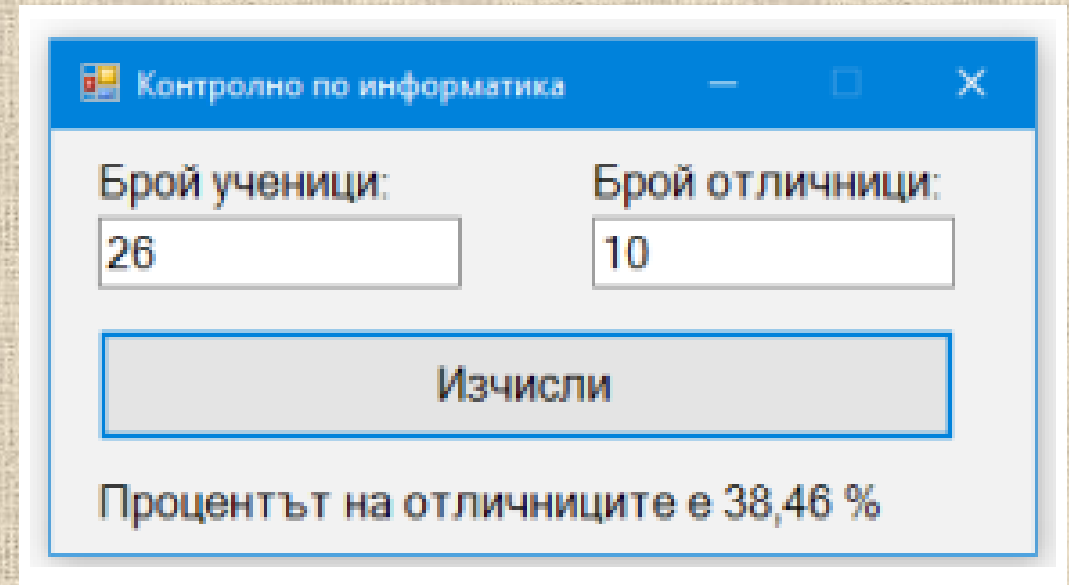
Създайте графично приложение с име **TravelTime**, което при въвеждане на дистанция (в km) и скорост (в km/h) извежда за колко време ще бъде измината даденото разстояние.

вход: **140** km **90** km/h

изход: 1.5555555555555556 h

Задача 2:

Направете графично приложение с име **Excellent**, в което се въвежда броят на учениците в един клас и броят на тези ученици, които са получили отлична оценка на контролна работа по информатика, след което се намира и показва процента на отличниците.



Контролно по информатика

Брой ученици: 26

Брой отличници: 10

Изчисли

Процентът на отличниците е 38,46 %

3. Символен тип - **char**

А) Характеристики

- съхранява числова стойност, представляваща номера на символа в Unicode-таблицата
- множество на допустимите стойности е в диапазона
от **0** до **65,535**
- размерът който заема в паметта е 2 байта (16 бита)
- не може да се третира като целочислен тип, но може да се извърши изрично преобразуване:

```
char chSomeChar = (char)65;
```

```
int nSomeInt = (int)'A';
```

Б) Четене и запис на единичен символ от конзолата

- **Console.Read()** чете символ след натискане на Enter.

```
char s = (char)Console.Read();
```

- **Console.ReadKey()** чете незабавно символа, без да чака натискане на Enter.

```
char symbol = Console.ReadKey().KeyChar;
```

В) Литерали

- Буква, цифра, символ:

`char letter = 'A';` // Присвоява символа 'A'

`char digit = '5';` // Присвоява символа '5'

`char symbol = '#';` // Присвоява символа ,#‘

- Unicode литерали

`char smile = '\u263A';` // Присвоява символа ☺

`char omega = '\u03A9';` // Присвоява символа Ω

- Шестнадесетични литерали

`char symbol = '\x0041';` // Присвоява символа ,A‘

Задача 1:

Създайте конзолно приложение с име **FirstLastLetter**, което чете име от конзолата, след което извежда първата и последната буква от него и поредния им номер в Unicode таблицата.

ВХОД: **Jhonathan**

ИЗХОД: **First letter is J with Unicode number 74**

Last letter is n with Unicode number 110