**public void Kvadrat ( int i) {…}**

**ob. Kvadrat ( ref a);**

*using System;*

*class Example {*

*public void Square(int i) {*

*i \*= i;*

*}*

*}*

*class HW {*

*public static void Main() {*

*Example ob = new Example();*

*int a = 8;*

*Console.WriteLine("Before calling Square method:\ta = {0}", a);*

*ob.Square(ref a);*

*Console.WriteLine("After calling Square method:\ta = {0}", a);*

*}*

*}*

**hw1.cs(20,12): error CS1502: The best overloaded method match for `Example.Square(int)' has some invalid arguments**

**hw1.cs(10,17): (Location of the symbol related to previous error)**

**hw1.cs(20,23): error CS1615: Argument `#1' does not require `ref' modifier. Consider removing `ref' modifier**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметр кабыл алган жерде ref кызматчы созу жок болгон учурда,**

**функцияны чакырып жатканда ref созун колдонууга болбойт**

**public void Kvadrat ( ref int i) {…}**

**ob. Kvadrat ( a);**

*using System;*

*class Example {*

*public void Square(ref int i) {*

*i \*= i;*

*}*

*}*

*class HW {*

*public static void Main() {*

*Example ob = new Example();*

*int a = 8;*

*Console.WriteLine("Before calling Square method:\ta = {0}", a);*

*ob.Square(a);*

*Console.WriteLine("After calling Square method:\ta = {0}", a);*

*}*

*}*

**hw1.cs(19,12): error CS1502: The best overloaded method match for `Example.Square(ref int)' has some invalid arguments**

**hw1.cs(9,17): (Location of the symbol related to previous error)**

**hw1.cs(19,19): error CS1620: Argument `#1' is missing `ref' modifier**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметр кабыл алган жерде ref кызматчы созу болгон учурда,**

**функцияны чакырып жатканда да ref созун колдонуу керек**

**public void Kvadrat ( out int i) {…}  
ob. Kvadrat ( ref a);**

*using System;*

class Example {

public void Square(out int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(ref a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(10,9): error CS0269: Use of unassigned out parameter `i'**

**hw1.cs(19,12): error CS1502: The best overloaded method match for `Example.Square(out int)' has some invalid arguments**

**hw1.cs(9,17): (Location of the symbol related to previous error)**

**hw1.cs(19,23): error CS1620: Argument `#1' is missing `out' modifier**

**Compilation failed: 3 error(s), 0 warnings**

**функция параметр кабыл алган жерде out кызматчы созу болгон учурда,**

**функцияны чакырып жатканда ref созун эмес, out созун колдонуу керек**

**public void Kvadrat ( ref int i) {…}  
ob. Kvadrat ( out a);**

using System;

class Example {

public void Square(ref int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(out a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(17,12): error CS1502: The best overloaded method match for `Example.Square(ref int)' has some invalid arguments**

**hw1.cs(8,17): (Location of the symbol related to previous error)**

**hw1.cs(17,23): error CS1620: Argument `#1' is missing `ref' modifier**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметр кабыл алган жерде ref кызматчы созу болгон учурда,**

**функцияны чакырып жатканда out созун эмес, ref созун колдонуу керек**

**public void Kvadrat ( out int i) {…}  
ob. Kvadrat ( a);**

using System;

class Example {

public void Square(out int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(9,9): error CS0269: Use of unassigned out parameter `i'**

**hw1.cs(17,12): error CS1502: The best overloaded method match for `Example.Square(out int)' has some invalid arguments**

**hw1.cs(8,17): (Location of the symbol related to previous error)**

**hw1.cs(17,19): error CS1620: Argument `#1' is missing `out' modifier**

**Compilation failed: 3 error(s), 0 warnings**

**функция параметр кабыл алган жерде out кызматчы созу болгон учурда,**

**функцияны чакырып жатканда да out созун колдонуу керек**

**public void Kvadrat ( int i) {…}  
ob. Kvadrat ( out ref a);**

using System;

class Example {

public void Square(int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(out ref a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(17,22): error CS1525: Unexpected symbol `ref'**

**Compilation failed: 1 error(s), 0 warnings**

**функция параметр кабыл алган жерде out кызматчы созу болгон учурда,**

**функцияны чакырып жатканда да out созун колдонуу керек**

**public void Kvadrat ( ref out int i) {…}  
ob. Kvadrat (ref out a);**

using System;

class Example {

public void Square(ref out int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(ref out a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(8,30): error CS1108: A parameter cannot have specified more than one modifier**

**hw1.cs(17,22): error CS1525: Unexpected symbol `out'**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметри бир гана кызматчы созун кабыл алат**

**public void Kvadrat ( out ref int i) {…}  
ob. Kvadrat ( out ref a);**

using System;

class Example {

public void Square(out ref int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(out ref a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(8,30): error CS1108: A parameter cannot have specified more than one modifier**

**hw1.cs(17,22): error CS1525: Unexpected symbol `'ref**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметри бир гана кызматчы созун кабыл алат**

**public void Kvadrat ( ref out int i) {…}  
ob. Kvadrat ( out ref a);**

using System;

class Example {

public void Square(ref out int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(out ref a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(8,30): error CS1108: A parameter cannot have specified more than one modifier**

**hw1.cs(17,22): error CS1525: Unexpected symbol `out'**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметри бир гана кызматчы созун кабыл алат**

**public void Kvadrat ( out ref int i) {…}  
ob. Kvadrat (ref out a);**

using System;

class Example {

public void Square(out ref int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(ref out a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(8,30): error CS1108: A parameter cannot have specified more than one modifier**

**hw1.cs(17,22): error CS1525: Unexpected symbol `'ref**

**Compilation failed: 2 error(s), 0 warnings**

**функция параметри бир гана кызматчы созун кабыл алат**

**in modificator**

using System;

class Example {

public void Square(in int i) {

i \*= i;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int a = 8;

Console.WriteLine("Before calling Square method:\ta = {0}", a);

ob.Square(a);

Console.WriteLine("After calling Square method:\ta = {0}", a);

}

}

**hw1.cs(8,24): error CS1644: Feature `readonly references' cannot be used because it is not part of the C# 7.0 language specification**

**Compilation failed: 1 error(s), 0 warnings**

**in модификатору параметрди окуу учун гана колдонулат. Ал колдонулган параметрди меоддун ичинде озгортууго болбойт**

**Эки сандын эң чоңун кайтаруучу методду жазыңыз. Метод эки параметрден турат. Методду сынап көрүңүз.**

using System;

class Example {

public int max(int a, int b) {

if (a > b) {

return a;

} return b;

}

}

class HW {

public static void Main() {

Example ob = new Example();

Console.WriteLine(ob.max(6,8));

}

}

***Жыйынтык: 8***

***Берилген эки параметрдин маанилерин алмаштырган методду жазыңыз. Параметрлерди шилтеме аркылуу өткөрүп берүү. Методду сынап көрүңүз.***

*using System;*

class Example {

public void swap(ref int a, ref int b) {

int t = a;

a = b;

b = t;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int m = 9, n = 24;

Console.WriteLine("m = {0}\tn = {1}", m, n);

ob.swap(ref m, ref n);

Console.WriteLine("m = {0}\tn = {1}", m, n);

}

}

***Жыйынтык***

***m = 9 n = 24***

***m = 24 n = 9***

***Сандын факториалын эсептөө методун жазыңыз, эсептөө натыйжаларын чыгуучу параметрге которуңуз. Эгер ыкма ийгиликтүү болгон болсо, анда true кайтарыңыз; эсептөө учурунда ашып кетсе, анда false дегенди кайтарыңыз.***

*using System;*

class Example {

public Boolean fact(int a) {

int res = 1;

for (; a > 1; a--) {

res \*= a;

}

if (res != 0) return true;

return false;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int m = 9;

if (ob.fact(m))

Console.WriteLine("true");

else

Console.WriteLine("false");

}

}

***Жыйынтык true***

***Сандын факториалын эсептөөнүн рекурсивдүү методду жазыңыз***

*using System;*

class Example {

public int fact(int a) {

if (a > 1) {

return a \* fact(a-1);

}

else return a;

}

}

class HW {

public static void Main() {

Example ob = new Example();

int m = 9;

Console.WriteLine(ob.fact(m));

}

}

***Жыйынтык 362880***

***Эки натуралдык сандын эң чоң жалпы бѳлүүчүсүн эсептөөчү методду жазыңыз (Евклиддин алгоритми).***

Үч натуралдык сандын эң чоң жалпы бѳлүүчүсүн эсептөөчү методду жазыңыз (Евклиддин алгоритми).

***18. Фибоначчи катарынын n-санынын маанисин эсептеген рекурсивдүү методду жазыңыз. Фибоначчи катарлары 1, 1, 2, 3, 5, 8, 13 ... натуралдык сандар тизмеги.***