

01. Melyik az ős és melyik a leszármazott osztály?

```
class x:y  
{...}
```

Válasz: Az y az ős, az x a leszármazott

02. Helyes-e a C#-ben a következő tömb definíció?

```
shot [] tmb = { 1,2,3};
```

Válasz: Igen

03. C#-ban van-e közös őse a típusoknak?

Válasz: Igen, a System.Object

04. Mely állítások igazak a virtual kulcsszóval definiált metódusokra?

Válaszok: Nem jelölhetjük abstract és static kulcsszóval jelölt metódusoknak is,  
A leszármazott számára jelzi, hogy az adott függvényt felül lehet definiálni.

05. Válogassa szét!

Válasz: Érték típus: Double, struktúra

Referenciatípus: Osztály, leszármazott osztály

06. Melyik kulcsszóval jelölt metódust lehet átdefiniálni az override kulcsszóval a leszármazott osztályban?

Válaszok: Virtual

Abstract

07. C#-ben melyik állítás NEM igaz a struktúra alapértelmezett konstruktorára?

Válasz: Van paramétere

08. Mire jó a C#-ban a + műveleti jel stringek esetén?

Válasz: Összefűzésre

09. C#-ban, ha a Kör osztály egyparaméteres int-et váró konstruktorral rendelkezik, hogyan kell Kör típusú objektumot létrehozni?

Válasz: Kör k = new Kör(5);

10. Léteznek-e globális változók, vagy függvények C#-ban?

Válasz: Nem, egyik sem létezik.

11. `static bool fgv(out int a) { return true; }` Helyes-e a fenti metódus?

Válasz: Nem, mert az out paraméter nem kap értéket benne.

12. Helyes-e a következő C#-nyelvű programrészlet?

```
int [] tmb = new int [] {1,2,30};  
foreach {int x in tmb} x += 5;
```

Válasz: Nem, mert a foreach-csek csak olvasni lehet az elemeket.

13. Lehet-e C#-ban osztályok nélkül programozni?

Válasz: Nem, legalább egy osztály létrehozása kötelező.

14. C#-ban milyen két fajta többdimenziós tömb létezik?

Válasz: Téglalap-alakú és egyenetlen

15. `string szöveg= „abcdefg”;`  
`string [] tmb = szöveg.Split(new char [] { 'a' });`  
Mi lesz a tmb a tömbben?

Válasz: „” és „bc” és „dfg”

16. Hogyan hívjuk az objektum létrehozását?

Válasz: Az osztály példányosításának.

17. Válassza ki az abstract metódusok jellemzőit!

Válaszok: Az override kulcsszó segítségével tudjuk definiálni a leszármazottban.  
Leszármazottban kötelező definiálni.  
Nem lehet definíciója, csak deklaráció.

18. C#-ban a Main() metódusnak milyennek kell lennie?

Válasz: Statikusnak

19. Mit ír ki ez a C#-nyelvű programsor?

```
System.Console.WriteLine(„A számok: {1}, {0}, {2}”, 5, 10, 15);
```

Válasz: 10, 5, 15

20. C#-ban melyik állítás igaz az osztályra?

Válasz: referenciatípus

1. A Microsoft által fejlesztett .NET Framework-ön kívül milyen alternatív megoldásokról hallott?  
SSCLI, Mono, DotGNU
2. C#-ban van-e közös őse a típusoknak?  
igen, a System.Object
3. C#-ban mire használható a sealed kulcsszó?  
osztályokra és metódusokra
4. Az ... osztály típusú változó. Mi hiányzik a mondatból?  
objectum
5. A felsoroltak közül melyik rendelkezik a legnagyobb értéktartománnyal C#-ban?  
a long
6. C#-ban a statikus metódusok ..  
csak statikus adatokon működhetnek és csak statikus metódusokat hívhatnak meg
7. C#-ban a statikus kivételeknek melyik a két nagy csoportja?
8. C#-ban milyen hozzáférés módosítók használhatók egy struktúrabelsejében?
9. Melyik konstruktorfajta fut le először C#-ban?
10. Hogyan lehet megírni a következő metódust C#-ban?  
static double Átlag(params double [] értékek) { ... }  
egy double tömbbel vagy vesszővel elválasztott double értékekkel
11. C#-ban a Console osztály WriteLine () metódusa ...  
statikus
12. Igaz-e, hogy a .NET egy programozási nyelv?  
nem, a .NET egy keretrendszer, egy környezet
13. Hogyan lehet megírni a saját osztályában a következő metódust?  
static void fgv(ref int x) { ... }
14. .Net-ben mit nevezünk szerelvénynek? (assembly-nek)?
15. C#-ban létezik-e olyan osztály, amit nem lehet példányosítani?  
igen, az abstract
16. Mit ír ki a következő programrészlet?  
int x=10;  
Console.WriteLine(x.ToString());  
10

17. Létezik-e ilyen C#-ban?

```
int? i – null;
```

létezik

18. C#-ban több catch ág esetén melyik kerüljön előre?

nem mindegy

19. Mit ír ki a következő programrészlet?

```
int x = 10;  
Console.WriteLine(x.GetType());
```

nem az int”

20. Milyen karakter jelzi a „szó szerinti sztringeket” a c#-ban?

21. Lefordul-e?

```
long lo = 4;  
short s = lo;
```

22. C#-ban milyen hozzáférés módosítók alkalmazhatók a típusok előtt?

23. C#-ban az osztályok és a struktúrák belsejében mi az alapértelmezett hozzáférés módosító?

24. Helyes-e a következő C# nyelvű kód?

```
private string név;  
public int Név  
{  
    get {return név;}  
}
```

25. Léteznek-e globális változók vagy függvények C#-ban?

26. Miben különbözik a string és a StringBuilder típus?

27. C#-ban hogyan kell egy str nevű string elemein végigiterálni foreach ciklussal?

28. Mi a közös őse az értéktípusoknak C#-ban?

29. C#-ban milyen két fajta többdimenziós tömb létezik?

30. C#-ban melyik állítás NEM igaz a struktúrákra?

31. Létezik-e unblock kulcsszó c#-ban?

32. Hogy hívják azt az adatmezőt C#-ban, amihez csak set van hozzárendelve, get nem?

33. C#-ban mire használják az eseményeket?

34. C#-ban mit jelent a delegáltak kontravariáns viselkedése?

35. Ha egy osztálynak van őse is, és interfészt is megvalósít, mi a felsorolás sorrendje C#-ban?
36. C#-ban egy osztály vagy struktúra hány interfészt valósíthat meg? Nem csak 1-et
37. Melyik állítás igaz a C#-beli interfészre?
38. C#-ban a delegált objektum melyik tulajdonsága mondja meg, hogy melyik metódust kapcsoltuk hozzá?
39. Melyik állítás igaz C#-ban?
40. C#-ban mit NEM tárol egy metódusreferencia-objektum?
41. A GetEnumerator () metódus melyik interfész tagja C#-ban?
42. C#-ban, ha egy referencia-típusú változót (pl. egy objektumot) adunk át értékként a metódusnak, az ...  
nem okoz fordítási hibát
43. C#-ban melyik metódust NEM öröklik a típusok az object típustól?  
Nem a GetHashCode ()
44. C#-ban, ha a Kör osztály kétparaméteres int-et és double-t váró konstruktorral rendelkezik, hogyan kell Kör típusú objektumot létrehozni?  
Kör k = new Kör (3, 5.9);
45. C#-ban a string típusú objektum tartalma kezdőértékkadás után...  
nem változtatható meg.
46. .NET alatt különbözik-e az MIL és a CIL?
47. Hogyan szokás C#-ban az osztályok private adatmezőit osztályon kívülről elérni?  
Nem lekérdező metódust írunk hozzá
48. C#-ban mely tagok virtuálisak?
49. Lehet-e C#-ban a konstruktor után base kulcsszót írni?
50. Mit tesz lehetővé a .NET?  
Alkalmazások nyelv-, rendszer- és platformfüggetlen felépítését.
51. Helyes-e a következő C#-nyelvű kód?  
DateTime dt = DateTime(2010,9,14);  
nem, mert hiányzik belőle a new
52. C#-ban mire használható a sealed kulcsszó?  
nem csak metódusokra
53. Mit NEM lehet C#-ban egy object tömbbe belerakni?  
nem objektumot

54. Igaz-e, hogy a .NET egy programozási nyelv?  
nem, a .NET egy keretrendszer, egy környezet

55. Milyen egységekre lehet használni a checked kulcsszót C#-ban?

56. Helyes-e a következő C#-nyelvű kód?

```
class Számla
{
    private int x;
    static Számla(int x0)
    {x=x0}
}
```

nem, mert a statikus konstruktor nem kaphat paramétert

57. C#-ban egy abszrak metódus mit üzen a programozónak?

58. A következő C#-nyelvű programkód melyik része adja a léptetést?

Az x++

59. C#-ban hogyan lehet egy leszármazott osztálybeli metódusban a közvetlen ős fgv() nevű metódusát meghívni?

base.fgv();

60. Milyen utasítások szolgálnak C#-ban kódsorok többszöri végrehajtására?

while, do-while, for, foreach

61. Helyes-e a következő C#-nyelvű kód?

```
class Számla
{
    private int x;
    static Számla(int x0)
    {x=x0;}
}
```

nem, nem helyes

62. C#-ban a sikertelen „as” konverzió eredménye egy  
nem false érték

63. C#-ban mi ez?

```
int [,] tomb = new int [4,5];
```

4sor és 5 oszlopot tartalmazó tömb

64. C#-ban legkevesebb hányszor futnak le egy do-while-ciklus utasításai?  
legalább egyszer.

65. double x=8,2;

object d=x;

C#-ban mi ez?

bedobozolás

66. C#-ban mely hozzáférés módosítók alkalmazhatók csak osztályok belsőjében?

67. A CLR lényegét melyik DLL állomány tartalmazza?  
nem a system32.dll

68. Helyes-e a következő C# nyelvű kód?

```
private string név;  
public string Név  
{  
    get {return név;}  
}
```

Igen, helyes

69. Miben különbözik a string és a StringBuilder típus?  
A string értéke nem változtatható meg, a StringBuilder-é pedig igen.

70. Helyes-e a következő C# nyelvű kód?

```
private string név;  
public string NÉV;  
{  
    get {return Név;}  
}
```

nem, mert a tulajdonság saját magát adja vissza, nem ad adatmezőt.

71. Melyik állítás igaz? C#-ban...

az értéktípusok közvetetten, a referenciák közvetlenül származnak a objectből

72. Mit jelent a következő C# nyelvű kód?

```
class a  
{...}  
class b : a  
{...}
```

a b nevű osztály az a nemű osztály leszármazottja

73. . Net platformon melyik a bővebb rendszer: a CTS vagy a CLS?  
nem a CLS a bővebb

74. Hogyan lehet meghívni a saját osztályába a következő metódust?  
static void fgv(ref int x) {...}

75. C#-ban hol definiálhatunk absztrakt metódust?  
nem csak virtuális osztályban

76. C#-ban melyik állítás NEM igaz a struktúrára?  
nem értéktípust

77. Körülbelül hányféle programozási nyelv használható .NET alatt?  
nem Kb. 20...

78. C#-ban melyik állítás igaz az osztályra?  
referenciatípus

79. Mi a C#-beli sztring típus .NET es teljes neve?  
System.String
80. Mit jelent az öröklési lánc C#-ban?  
egy osztályból származtatunk egy újabb osztályt és abból is származtatunk további osztályokat
81. Mit ír ki ez a C#-nyelvű programsor?  
System.Console.WriteLine(„A számok: {1}, {0}, {2}, 5, 10, 15);  
10,5,15
82. C#-ban, ga egy metódusnak van parancsparamétere, az...
- 83.
84. C#-ban, ha egy referencia-típusú változót (pl. egy objektumot) adunk át  
referenciaként a metódusnak, az ...
85. Hol kezdődik a C#-nyelvű program végrehajtása?  
A Main () metódusnál
86. Mely jellemzők igazak a C# nyelv statikus osztályaira?
87. A következő C#-nyelvű programkód melyik része adja a léptetést?  
for ( x=1; x==1;x++)  
Az x++
88. C#-ban, ha a Kör osztály két paraméteres int-et és double-t váró konstruktorral  
rendelkezik, hogyan kell Kör típusú objektumot létrehozni?  
Kör k = new Kör (3,5.9)
- 89.
90. Lehet-e C#-ban osztályok nélkül programozni?  
Nem, legalább egy osztály létrehozása kötelező.
91. A .NET környezetben minek a rövidítése a BCL?  
Base Class Library
92. Mivel jelölhetjük a megjegyzéseket a C#-nyelvű kódban?  
/\*\*/
93. Milyen típusú változó a struktúra C#-ban?  
nem referencia
94. Van-e olyan metódus, amivel a szöveges állomány tartalmát egy az egyben be tudjuk  
olvasni?  
nem a ReadAll ()
95. C#-ban az IEnumerator interfész milyen tagokat tartalmaz?



96. C#-ban melyik tulajdonság adja vissza a gyűjteményosztályok elemszámát?  
Count
97. A fájlkezeléshez szükséges osztályok melyik névtérben vannak?  
System.IO
98. C#-ban mi ez?  
 $x \Rightarrow x \% 3 == 0$   
egy lambda-kifejezés
99. C#-ban a kiváltott esemény első paramétere ...  
this
100. C#-ban helyes-e a következő?  
public delegate void MR;  
public event MR Esemény;  
nem, mert az MR-nek nincs ()-e
101. C#-ban az IComparer interfészt melyik osztály valósítja meg?  
nem a rendezendő, hanem egy segédosztály
102. C#-ban helyes-e a következő lambda-kifejezés?  
 $i \rightarrow i > 10$   
nem, mert az  $\Rightarrow$  operátor kell bele
103. A kétirányú láncolt lista szerkezetet melyik generikus tárolóosztály valósítja meg C#-ban?  
LinkedList <T>
104. C#-ban mikor lesz egy esemény null értékű?  
ha nincs rá feliratkozó
105. Milyen hozzáférés módosítók léteznek C#-ban?  
public, private, internal, protected, protected internal
106. Melyik NEM C#-nyelvű kulcsszó?  
delete
107. Helyes-e a következő C#-nyelvű programrészlet?  
**int [] tmb = new int[] {1,2,3};**  
**foreach (int x in tmb) Console.WriteLine(x);**  
igen
108. A következő C#-nyelvű programkód melyik része a kezdőértékkadás?  
**for (x =1; x ==1; x++)**  
Az  $x = 1$
109. Mit jelent az öröklési lánc C#-ban?  
egy őosztályból származtatunk egy újabb osztályt és abból is származtatunk további osztályokat

110. Mire jó C#-ban a + műveleti jel stringek esetén?  
összefűzésre
111. Mire jó a C# checked kulcsszava?  
ellenőrzött típuskonverzióra
112. C#-ban tömbtúlindexelés esetén milyen típusú kivételt dob a CLR?  
IndexOutOfRangeException típusú kivételt dob
113. C#-ban a Db és a db név ugyanazt a változót jelenti-e?  
Nem mert C-stílusú nyelv, kis- és nagybetűket megkülönbözteti.
114. A .NET környezetben minek a rövidítése a CLR?  
Common Language Runtime
115. **byte b = 125**  
**int i = b;**  
Mi ez?  
implicit, bővítő típuskonverzió
116. Hogy hívjuk azt az adatmezőt C#-ban, amihez csak get tulajdonság van hozzárendelve, set nem?  
csak olvasható
117. A Microsoft által fejlesztett .NET Framework-on kívül milyen alternatív megoldásokról hallott?  
Mono
118. C#-ban hogy lehet egy leszármazott osztálybeli metódusban a közvetlen ősfgv() nevű metódusát meghívni?  
base.fgv();
119. C#-ban hol állhat a finally blokk?  
az utolsó catch után
120. Mely jellemzők igazak a C# nyelv statikus osztályaira?  
csak statikus mezőket és statikus metódusokat tartalmazhatnak
121. A .NET környezetben minek a rövidítése a BCL?  
Base Class Library
122. C#-ban több catch ág esetén melyik kerüljön előre?  
a legspeciálisabb
123. C#-ban melyik állítás NEM igaz az osztályra?  
értéktípus
124. A .NET környezetben minek a rövidítése a CLR?  
Common Language Runtime
125. Hol kezdődik a C#-nyelvű program végrehajtása?  
A Main() metódusnál

126. Mi a C#-beli string típus .NET teljes neve?  
System.String
127. Mivel lehet kivételt továbbdobni C#-ban?  
throw-val
128. Hogy kell C#-ban 5 elemű, ulong típusú tömböt létrehozni?  
ulong [] tömb = new ulong[5]
129. long lo = 4;  
shot s = lo;  
Mi ez?  
implicit, szűkítő típuskonverzió
130. Hogy hívjuk azt az adatmezőt C#-ban, amihez csak get tulajdonság van hozzárendelve, set nem?  
csak olvasható
131. Mire jó C#-ban a + műveleti jel stringek esetén?  
összefűzésre
132. Hogy lehet meghívni a saját osztályban a következő metódust?  
static void fgv(out int x) {...}  
int y; fgv (out y);
133. Melyik konstruktorfajta fut le először C#-ban?  
először a statikus, utána a nem statikus
134. C#-ban a statikus metódusok...  
csak statikus adatokon működhetnek és csak statikus metódusokat hívhatnak meg
135. Mivel jelölhetjük a megjegyzéseket a C#-nyelvű kódban?  
// vagy /\* \*/
136. A C#-beli kivételeknek melyik a két nagy csoportja?  
az alkalmazásbeli és rendszerkivételek
137. Milyen típusú változó a struktúra C#-ban?  
érték
138. Az IL nyelvű dll-t vagy exe-t a CLR mely része fordítja le a platformfüggő gépi utasításokra?  
A JIT-fordító
139. static bool fgv(out in a) {return true;}  
Helyes-e a fenti metódus  
Nem, mert az out paraméter nem kap értéket benne.