

Szoftverfejlesztés és -tesztelés a gyakorlatban

Szakképesítés azonosító száma, megnevezése:

5-0613-12-03 Szoftverfejlesztő és -tesztelő technikus

Vizsgarészhez rendelt vizsgafeladat megnevezése:

Asztali- és webes szoftverfejlesztés, adatbázis-kezelés feladatsor

A vizsgatevékenység során elérhető pontszám: **65 pont**

A gyakorlat vizsgatevékenység időtartama: **240 perc**

A gyakorlati vizsgatevékenység időpontja: **2023. év.....hó.....nap**

Nógrád Megyei Szakképzési Centrum Szent-Györgyi Albert Technikum

A gyakorlati vizsgafeladat elfogadását javaslom: dátum a vizsgaszervező intézmény képviselője
A gyakorlati vizsgafeladattal egyetértek: dátum a vizsgabizottság tagja
A gyakorlati vizsgafeladatot jóváhagyom: dátum a vizsgabizottság elnöke

1. Konzolos és grafikus részt egyaránt tartalmazó asztali alkalmazás fejlesztése

Konzolos program fejlesztése (15 pont)

A következő feladatban használt autók eladásával kapcsolatos szöveges állomány áll a rendelkezésre, melyekkel programozási feladatokat kell megoldania. A mellékelt adatállomány eredetileg 24199 használt járműből áll, amelyek 2023-ban értékesítésre kerültek a kanadai Ontario állam Toronto belvárosának 25 km-es körzetében. Ezek közül a legdrágább autók adatait tartalmazza a forrás.

A feladat megoldása során vegye figyelembe a következőket:

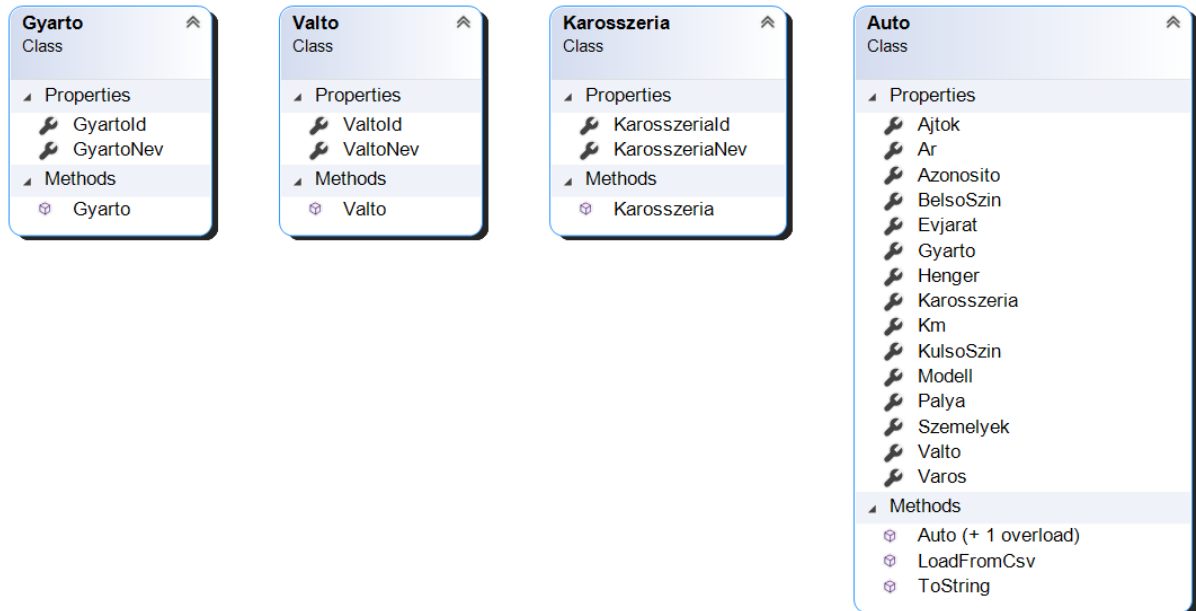
- *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3. feladat:)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes azonosítók és kiírások is elfogadottak.*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
- *A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon*

Az adatforrás „carsData.csv” UTF-8 kódolású szöveges állomány. Az állomány első sora tartalmazza a mezőneveket. Az adatok pontosvesszővel vannak elválasztva.

Az egyes mezőnevek tartalma, és típusa:

- | | |
|--------------------------------|--|
| - azonosito | az autó egyedi azonosítója |
| - evjarat | az autó gyártási éve |
| - gyarto | az autó gyártója, önálló osztály (gyarto azonosito, gyarto) |
| - modell | az autó modelljének megnevezése |
| - kilometerora allas | az autó kilométeróra állása |
| - karosszeria | az autó karosszériája, önálló osztály (karosszeria azonosito, karosszeria) |
| - hengerek szama | a hengerek száma a motorban |
| - valto azonosito | az autó váltója, önálló osztály (valto azonosito, valto) |
| - kulsoszin | az autó színe |
| - belso szín | az autó utasterének színe |
| - szemelyek szama | hány személyes az autó |
| - ajtok | ajtók száma |
| - fogyasztas varosban | az autó fogyasztása városi forgalomban |
| - fogyasztas autopalyan | az autó fogyasztása autópályán |
| - ar | az autó bruttó ára kanadai dollárban |

1. Készítsen konzolos alkalmazást a következő feladatok megoldására, melynek projektjét **AutoKonzol** néven mentse el!
2. Készítsen saját osztályokat **Auto**, **Karosszeria**, **Valto**, és **Gyarto** azonosítóval, amelyeket az osztálydiagramok szemléltetnek!



3. Készítsen statikus metódust az **Auto** osztályhoz az adatforráshoz történő kapcsolódásra, és az adatforrásban levő adatok betöltésére! A metódus neve **LoadFromCsv** legyen. A metódus visszatérési értéke az **Auto** osztályból képzett lista legyen! A metódus paramétere a betöltendő fájl neve legyen!
4. Készítsen az **Auto** osztályban egy virtuális metódust a ToString() felülírására, amivel az autók főbb adatait ki tudja írni a minta alapján. Az első adat az autó azonosítója. A fogyasztás értéke a városi és az autópályán megadott fogyasztás *átlaga* legyen, 2 tizedesjegyre kerekítve.

1. autó:

```
Gyártó - modell: Ford - Mustang
Fogyasztás: 18,05 l/100 km
Kilométeróra állása: 96560 km
Váltó típusa: 4 Speed Manual
Ára (CAD): 499000
```

5. Olvassa be az adatforrás adatait és tárolja az adatokat az **Auto** osztály segítségével egy olyan összetett adatszerkezetben, amely használatával a további feladatok megoldhatóak!
6. Írja a képernyőre a minta szerint, hogy hány Rolls-Royce márkájú, sedan karosszériájú, 8 hengeres használt autót adtak el az adott időszakban.
7. Határozza meg és írja ki a minta szerint a legdrágább autó árát forintban. Feltételezheti, hogy csak egy legdrágább autó van. A kanadai dollár árfolyama: 248,02 Ft. A számított értéket egész számra kerekítse. A kiírás első részéhez használja a már elkészített virtuális metódust.

8. A vevők gyakran hajlandóak többet fizetni az olyan autókért, amiket a Forma-1-es futamokon is látnak, mint pl. Aston Martin, Ferrari, McLaren, Mercedes-Benz, Porsche. Gyűjtse ki egy szöveges fájlba a felsorolt márkájú autóknek a következő adatait: azonosító, gyártó neve, modell, évjárat, kilométeróra állás. Az összetartozó adatok egy sorban pontosvesszővel elválasztva kerüljenek kiírásra. A fájl neve **forma-1.txt** legyen.

A fájl tartalmának részlete:

```
86;Porsche;911; 2022; 3446
91;Ferrari;Roma; 2021; 5257
92;Porsche;911; 2022; 5851
110;Porsche;911; 2022; 2317
112;Aston Martin;Martin; 2022; 4703
113;Aston Martin;Martin; 2022; 4703
127;Ferrari;F8; 2021; 900
132;Porsche;911; 2022; 1900
138;Porsche;911; 2021; 3413
143;Porsche;911; 2021; 6000
```

Minta:

6. feladat - 8 hengeres, Rolls-Royce sedanok száma az adott időszakban: 1 db

7. feladat - A legdrágább autó

19. autó:

Gyártó - modell: Lamborghini - Aventador

Fogyasztás: 33,95 l/100 km

Kilométeróra állása: 1004 km

Váltó típusa: 7 Speed Automatic

Ára (CAD): 1699998

Ára (HUF): 421633504

8. feladat - Adatok fájlba írása

Grafikus program fejlesztése (10 pont)

9. Készítsen grafikus alkalmazást a következő feladatok megoldására, melynek projektjét **AutoGUI** néven mentse el! A feladat forrása megegyezik az **AutoKonzol** projektnél megadott forrással. Az **AutoKonzol** projektből használhatja az ott létrehozott osztályokat (**Auto**, **Gyarto**, **Valto**, **Karosszeria**).
10. Egyetlen ablakot kell elkészítenie a mintának megfelelően. A form-on lévő vezérlők típusai és a feliratok feleljenek meg a mintának.
11. Az ablak megjelenésekor a fájlból olvassa be az autók adatait, és a bal oldali listbox vezérlőben jelenjenek meg az autók gyártóinak nevei, mindegyik csak egyszer. A gyártók közül legyen az első gyártó a kiválasztott állapotú.
12. Az ablak megjelenésekor a karosszériához tartozó listboxban jelenjenek meg az autók karosszériáinak lehetséges értékei, mindegyik csak egyszer. Nem kell kiválasztania értéket.
13. Az ablak megjelenésekor a váltóhoz tartozó listboxban jelenjenek meg az autók váltóinak lehetséges értékei, mindegyik csak egyszer. Nem kell kiválasztania értéket.
14. Az ablak megjelenésekor az ajtók száma opciók értékeiből nem kell választania.
15. Az ablak megjelenésekor az „Évek száma” szöveg alatti első és második szöveges mezőbe a felhasználó bevihet egész számokat, ami az autó korára vonatkozó intervallumot adja meg. Pl. a 10 - 20 év megadása azt jelenti, hogy most, 2023-ban a 2003-2013 közötti évjáratú autókat keressük. Kezdetben itt ne jelenjen meg semmi.
16. A keresés gomb megnyomásakor keresse meg az összes, a feltételeknek megfelelő autót. Feltételezheti, hogy előtte a felhasználó minden választási lehetőséget helyesen beállított. Az autók korát a rendszeridő évéhez viszonyítsa. A 2023-as évben gyártott autó 0 évesnek számít. A jobboldali listboxban jelenítse meg a kikeresett autók következő adatait, kötőjellel elválasztva a minta szerint: azonosító, évjárat, modell, km, ár.
17. Mielőtt a következő kereséskor új találatokat jelenítené meg, törölje az eredménylistát.

A minta a következő oldalon található.

Használt autók eladása Torontóban - 2023

Autó gyártók

Ford
Porsche
Dodge
Ferrari
Lamborghini
McLaren
Chevrolet
Bentley
Rolls-Royce
Maserati
Aston Martin
Mercedes-Benz
Audi

Ajtók száma

☒ 2
☐ 3
☐ 4
☐ 5

Évek száma

1 - 10

Karosszéria típusa

Roadster

Váltó típusa

7 Speed Automatic

Keresés

A keresésnek megfelelő autók

14. - 2022 - F8 - 327 km - 599987 CAD
15. - 2022 - F8 - 656 km - 659987 CAD
22. - 2021 - F8 - 2455 km - 629987 CAD

2. Reszpónzív weboldal készítése (10 pont)

1. Csatolja a style-main.css-t a HTML oldalhoz.
2. A sidebar-on belül található két twin osztályú elem, az egyik háttérszíne #2185C5, a másik háttérszíne #FF7F66 legyen. A feladat megoldása során, a HTML fájlban semmilyen módosítást nem hajthat végre. A betűk színe fehér legyen.
3. A láblécben helyezze a szöveget középre.
4. A media query-ket helyezze el egy olyan lekérdezésben, ami akkor hajtja végre a benne foglalt media query-ket, ha böngésző támogatja a CSS-GRID-et.
5. Ha a böngésző legalább 27em széles, akkor a képek és a szövegek egymás mellett helyezkedjenek el úgy, hogy a kép 8em széles legyen, a mellette található szöveg automatikusan töltse ki a maradék rendelkezésre álló teret.
6. Állítsa be a cards osztályú elemek szélességét a repeat függvény használatával úgy, hogy két 25em szélességű oszlop álljon rendelkezésre a képeknek, 65 és 45 em képernyő szélesség esetében is.
7. A double osztályú képek 3 sort foglaljanak el az oldalon 65 és 45 em képernyő-szélesség esetében is.
8. Ha a képernyő 65em széles:
 1. Szelektálja ki a site osztályú elemet és állítsa be úgy a tulajdonságait és értékeit, hogy két oldalt 15em szélesség álljon rendelkezésre a sidebar két twin elemének az elhelyezkedéshez, középen pedig automatikus legyen a szélesség.
 2. Állítsa be a masthead és main-nav osztályú elemeket, hogy az oldal teljes szélességét kitöltsék.
 3. Állítsa be a sidebar osztályú konténer két elemét úgy, hogy azok a képernyő két szélén helyezkedjenek el.
 4. Állítson be 1em távolságot a cards gyermek elemei között.
9. Ha a képernyő 27em széles, a képek és a szöveg két oszlopban helyezkedjen el egymás mellett úgy, hogy a képnek 8em szélesség álljon rendelkezésre, a többi helyet automatikusan töltse ki a szöveg.

Get your photos exhibited in galleries around the world


Home

About

Resources


Tutorials

Contact




Lines of Growth

Everything that grows creates lines; the abstract telling of the story of a life.




Pattern Recognition

Human beings are hard-wired to look for patterns in all things.




In the Field

Did you know the word "silhouette" is the last name of a French 18th century politician?




Spelling the Future

Long before they made colourful plastic bricks they made toys from hardwood.




Water, Earth, and Sky

For some, water is a means of bodily sustenance. For others, it's a medium of transportation.




Flying Carpet

Looking at this photo it's easy to see how the idea of a flying carpet came about.




Lines of Growth

Everything that grows creates lines; the abstract telling of the story of a life.




Pattern Recognition

Human beings are hard-wired to look for patterns in all things.




In the Field

Did you know the word "silhouette" is the last name of a French 18th century politician?




Spelling the Future

Long before they made colourful plastic bricks they made toys from hardwood.



Water, Earth, and Sky

For some, water is a means of bodily sustenance. For others, it's a medium of transportation.



Flying Carpet

Looking at this photo it's easy to see how the idea of a flying carpet came about.

Grid lets you do unusual things

Creating this layout, where two items share the total available viewport width evenly alongside full-width and centered content, is surprisingly complex if you don't have CSS Grid available.

Grid requires a rethink

Seeing this layout you may still think it's better to just use old methods with auto-margins and max-widths. And you can, but looking at the CSS you'll see the Grid approach is both cleaner and easier to understand.

All photos: [GuruShots](#)

For more information, visit [GuruShots Exhibitions](#).

Get your photos exhibited in galleries around the world


Home

About

Resources


Tutorials

Contact




Lines of Growth

Everything that grows creates lines; the abstract telling of the story of a life.




Pattern Recognition

Human beings are hard-wired to look for patterns in all things.




In the Field

Did you know the word "silhouette" is the last name of a French 18th century politician?




Spelling the Future

Long before they made colourful plastic bricks they made toys from hardwood.




Water, Earth, and Sky

For some, water is a means of bodily sustenance. For others, it's a medium of transportation.




Lines of Growth

Everything that grows creates lines; the abstract telling of the story of a life.




In the Field

Did you know the word "silhouette" is the last name of a French 18th century politician?




Spelling the Future

Long before they made colourful plastic bricks they made toys from hardwood.




Water, Earth, and Sky

For some, water is a means of bodily sustenance. For others, it's a medium of transportation.



Pattern Recognition

Human beings are hard-wired to look for patterns in all things.



Flying Carpet

Looking at this photo it's easy to see how the idea of a flying carpet came about.

Grid lets you do unusual things

Creating this layout, where two items share the total available viewport width evenly alongside full-width and centered content, is surprisingly complex if you don't have CSS Grid available.

Grid requires a rethink

Seeing this layout you may still think it's better to just use old methods with auto-margins and max-widths. And you can, but looking at the CSS you'll see the Grid approach is both cleaner and easier to understand.

All photos: [GuruShots](#)

For more information, visit [GuruShots Exhibitions](#).

27 em képernyő szélesség esetén

45 em képernyő szélesség esetén

8

Get your photos exhibited in galleries around the world

[Home](#) [About](#) [Resources](#) [Tutorials](#) [Contact](#)

Grid lets you do unusual things

Creating this layout, where two items share the total available viewport width evenly alongside full-width and centered content, is surprisingly complex if you don't have CSS Grid available.



Lines of Growth

Everything that grows creates lines; the abstract telling of the story of a life.



Pattern Recognition

Human beings are hard-wired to look for patterns in all things.



In the Field

Did you know the word "silhouette" is the last name of a French 18th century politician?



Spelling the Future

Long before they made colourful plastic bricks they made toys from hardwood.



Water, Earth, and Sky

For some, water is a means of bodily sustenance. For others, it's a medium of transportation.



Flying Carpet

Looking at this photo it's easy to see how the idea of a flying carpet came about.



Lines of Growth

Everything that grows creates lines; the abstract telling of the story of a life.



In the Field

Did you know the word "silhouette" is the last name of a French 18th century politician?



Spelling the Future

Long before they made colourful plastic bricks they made toys from hardwood.



Pattern Recognition

Human beings are hard-wired to look for patterns in all things.



Water, Earth, and Sky

For some, water is a means of bodily sustenance. For others, it's a medium of transportation.



Flying Carpet

Looking at this photo it's easy to see how the idea of a flying carpet came about.

Grid requires a rethink

Seeing this layout you may still think it's better to just use old methods with auto-margins and max-widths. And you can, but looking at the CSS you'll see the Grid approach is both cleaner and easier to understand.

All photos: [GuruShots](#)
For more information, visit [GuruShots Exhibitions](#).

60em szélességű képernyő esetén

3. Backend programozás (15 pont)

Az alábbi feladatban egy hajótársaság weboldalának backend szerverét kell elkészítenie.

1. Hozzon létre backend szerver projektet az Ön által választott programnyelven, illetve fejlesztési környezetben! A projektmappát „Vezetéknév_Keresznév_backend” formában nevezze el!

2. Készítsen adatbázist **menetrend** néven, amely a createdatabase.php fájlban kerüljön mentésre! Ha az adatbázis létrejött, írassa ki: „Az adatbázist sikeresen létrehozta!”, ellenkező esetben: „Sikertelen a létrehozás!”

3. Hozzon létre egy connect.php nevű fájlt, amelyben létrehozza a kapcsolatot az újonnan létrehozott adatbázissal. Ellenőrizze, hogy létre jön-e kapcsolat, majd az ellenőrzést tegye kommentbe. Az újabb feladatoknál az itt létrehozott kapcsolatot használja.

4. Készítsen egy createtable.php nevű fájlt, amely létrehozza hajok nevű táblát az alábbi oszlopokkal:

```
hajok (  
id int(11) NOT NULL,  
jarat varchar(255) COLLATE utf8_hungarian_ci NOT NULL,  
honnan varchar(255) COLLATE utf8_hungarian_ci NOT NULL,  
hova varchar(255) COLLATE utf8_hungarian_ci NOT NULL,  
indul time NOT NULL,  
erkezik time NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci;
```

Ha a tábla létrejött, írassa ki: „A hajók tábla sikeresen létrejött!”, ellenkező esetben: „Sikertelen a tábla létrehozása”.

5. Készítsen egy insertdata.php nevű fájlt, amelyben 5 sort szúr be a meglévő adatbázisba, melyek a következők:

```
(1, 'A1', 'Siófok', 'Balatonalmádi', '09:30:00', '10:30:00'),  
(2, 'A2', 'Balatonalmádi', 'Alsóörs', '10:35:00', '11:10:00'),  
(3, 'A2', 'Alsóörs', 'Csopak', '11:10:00', '11:40:00'),  
(4, 'A2', 'Csopak', 'Siófok', '11:40:00', '12:20:00'),  
(5, 'A3', 'Siófok', 'Csopak', '16:00:00', '16:45:00').
```

Ügyeljen arra, hogy az összes sor beszúrára kerüljön, ne csak az utolsó sor!

Ha sikeres a beszúrás, az „Adatok mentésre kerültek.” szöveg jelenjen meg.

6. Készítsen egy createjson.php nevű fájlt, amely egyben egy végpont is, ahol az adatok a következőképpen jelennek meg:

```

<?php array (
  0 =>
    array (
      'id' => '1',
      'jarat' => 'A1',
      'honnan' => 'Siófok',
      'hova' => 'Balatonalmádi',
      'indul' => '09:30:00',
      'erkezik' => '10:30:00',
    ),
  1 =>
    array (
      'id' => '2',
      'jarat' => 'A2',
      'honnan' => 'Balatonalmádi',
      'hova' => 'Alsóörs',
      'indul' => '10:35:00',
      'erkezik' => '11:10:00',
    ),
  2 =>
    array (
      'id' => '3',
      'jarat' => 'A2',
      'honnan' => 'Alsóörs',
      'hova' => 'Csopak',
      'indul' => '11:10:00',
      'erkezik' => '11:40:00',
    ),
)

```

Ugyanakkor a fájl megnyitásával egyúttal mentésre kerüljön automatikusan az összes adat egy hajok.json fájlba is.

7. Készítsen egy delete.php fájlt, amely törli azt a sort, amelynek a id-jét az url-ben megadjuk:

delete.php?id=1

Sikeresen törlés esetén írassa ki: „A törlés sikeres”, ellenkező esetben: „A törlés sikertelen”.

4. Frontend programozás (15 pont)

Készítsen egy új fájlt menetrend.php néven, ahol az előző feladatban elkészített adatbázis adatait olvassa be a következő táblázatos formátumban:

Hajók menetrendje

Keresés a listában:

Járat	Honnan	Hová	Indulás	Érkezés	Módosítás
A1	Siófok	Balatonalmádi	09:30:00	10:30:00 perc	Módosítás
A2	Balatonalmádi	Alsóörs	10:35:00	11:10:00 perc	Módosítás
A2	Alsóörs	Csopak	11:10:00	11:40:00 perc	Módosítás
A2	Csopak	Siófok	11:40:00	12:20:00 perc	Módosítás
A3	Siófok	Csopak	16:00:00	16:45:00 perc	Módosítás

Helyezze el h1-es címsorban a címet. Alá helyezzen el kereső mezőt, ahol járat, honnan és hová adatokra kereshetünk.

A sorok végén a Módosítás linkre kattintva a update.php oldalra navigáljon a program.


Menetrend módosítása

Járat:


Honnan:

Hová:

Indulás:

Érkezik:

[Vissza a menetrendhez](#)

A megjelenő űrlap olvassa be a módosítandó adatokat.

Tisztítsa meg a Járat input mezőbe került értéket, hogy sem előtte, sem mögötte ne legyenek üres szóközök.

Ha a járat input mezőt üresen hagyná a felhasználó, jelenjen meg egy üzenet: „A járat mező nem maradhat üresen!”

Ha nincs hiba, akkor az adatbázisban történjen meg az adatok módosítása.