

Budapesti Műszaki Egyetem
Villamosmérnöki és Informatikai Kar
Automatizálási és Alkalmazott Informatika Tanszék

Tanulást segítő kliensalkalmazás készítése

Kliensalkalmazások fejlesztése – házi feladat dokumentáció

Talabér Gábor László
WL7A6D

Tartalomjegyzék

Feladatkiírás2

Használati útmutató3

Kérdésbankok kezelése3

Szerkesztő mód3

A kulcsszó mező használata:4

Böngésző mód4

Teszt mód5

A szoftver működésének bemutatása5

A dokumentum6

A nézetek6

EditorView6

BrowseView7

TestView8

Függelék10

Felhasznált szoftverek10

Feladatkiírás

A feladat egy olyan alkalmazás kifejlesztése, melyet tetszőleges tananyag kikérdezésére használhatunk.

A kikérdezendő tananyagokat különböző csv fájlokban tároljuk. A szoftver így tartalmazni fog egy importáló funkciót, mely beolvassa a kérdés adatbázist. A kikérdezésre, a felkészültség ellenőrzésére néhány funkciót használhat a felhasználó:

- Teszt mód: a szoftver előre beállítható számú kérdést tesz fel a felhasználónak a kérdésbankból véletlenszerűen kiválasztva, majd azok megválaszolása után kiértékeli a megadott válaszokat, és megjeleníti százalékban az elért eredményt. A teszt módot követően a program grafikusan, egy diagram segítségével megjeleníti a teszt eredményét
- Tanuló mód: a szoftver feltesz véletlenszerűen egy kérdést, majd kattintásra megjeleníti hozzá a választ is. A kérdést tovább lehet léptetni egy következő véletlenszerű kérdésre, ez így folytatható a végtelenségig.

A kérdés és a válaszok mellett az adatbázis tartalmaz egy kulcsszó mezőt is, melyeket teszt üzemmód esetén használ fel a szoftver. A teszt üzemmód alatt egy kérdésre adott választ akkor fogad el az alkalmazás, ha az tartalmazza az összes megadott kulcsszót.

- Az alkalmazás lehetőséget ad arra, hogy kérdés bankot lehessen szerkeszteni, illetve létrehozni (szerkesztő mód).

A szoftver Document-View architektúrát fog alkalmazni: a kérdésbank lesz a dokumentum, a teszt, tanuló és a szerkesztő mód pedig annak nézetei.

Használati útmutató

Kérdésbankok kezelése

A program úgynevezett kérdésbankokat használ, melyek egy-egy kérdésgyűjteményt jelentenek. Ezek a gyűjtemények kérdéseket, a hozzájuk tartozó válaszokat és kulcsszavakat tartalmaznak. A kérdésbankokat csv fájlokban tárolja.

A fájl menüpontra kattintva az alábbi funkciók közül választhatunk:

- Új kérdésbank: új, üres kérdésbank létrehozása és szerkesztése
- Kérdésbank megnyitása: már mentett kérdésbankok böngészése, megnyitása, betöltése. A program kizárólag csv formátumú fájlokat fogad el!
- Kérdésbank mentése: újonnan létrehozott kérdésbank esetén a mentés helyének megadása után menthetjük el a fájlt. Ha megnyitott fájlról van szó, akkor a kérdésbank a megnyitott fájl felülírásával mentődik.
- Kérdésbank mentése másként: Ezt a funkciót használva minden esetben megadhatjuk a fájl mentésének helyét és a fájlnevet a mentés előtt.
- Kérdésbank bezárása: a megnyitott kérdésbank bezárása.

A kérdésbank kérdései szabadon szerkeszthetők, illetve kérdések hozzáadhatóak, törölhetőek (lásd: szerkesztő mód).

Szerkesztő mód

Szerkesztő módban az aktuálisan megnyitott kérdésbank kérdéseit szerkeszthetjük. A program 10-es csoportokban kilistázza az összes kérdést, amit a kérdésbank tartalmaz. A

kérdések között navigálhatunk, áttekinthetjük a felvett kérdéseket, illetve törölhetjük azokat. A mód tartalmaz egy mezőt, ahol új kérdést is hozzáadhatunk a kérdésbankhoz.

Fontos, hogy egy kérdésbejegyzés hozzáadása akkor történhet meg, ha annak minden mezője (kérdés, válasz, kulcsszó) ki lett töltve.

A kulcsszó mező használata:

A kulcsszó mező a teszt és böngésző módban a felhasználó beírt válaszok ellenőrzésére szolgál. Működése: a mezőben vesszővel elválasztva találhatóak szavak, kifejezések lesznek a kulcsszavak. A program akkor fogadja el a választ, ha az ÖSSZES kulcsszó megtalálható a válaszban.

A szerkesztésnél így az alábbi elvek alapján válasszuk meg a kulcsszavakat:

- Ha pontos egyezést szeretnénk, akkor másoljuk át a válaszmezőbe írt választ. Ez például nyelvtanulás esetén lehet hasznos, mikor a szó helyesírása is fontos.
- Ha a válaszban szöveget várunk, akkor érdemes csak a fontosabb szavakat vesszővel elválasztva felsorolnunk.
- Ha a várt szónak ragozott formája is elfogadható akkor érdemes csak a szótövet felvenni kulcsszóként.
- Ha feleletválasztós kérdést szeretnénk feltenni, akkor a kérdésben adjuk meg az egyes lehetőségeket, pl: a), b), c) felsorolással, és a kulcsszó legyen a helyes válasz karaktere.

Böngésző mód

Ez az üzemmód a tananyag tanulási fázisában hasznos. Ha már elkezdtek a tanulást, de még nem ismerjük mélyrehatóan a tananyagot, akkor még nem érdemes gyakorló teszteket írni, de már tesztelhetjük a tudásunkat azzal, hogy megnézzük egy-egy kérdésre tudnánk-e a választ. A böngésző mód ezt lehetővé teszi:

- A módban egyszerre egy kérdés szerepel az oldalon
- A feltett kérdésre beírhatunk egy választ vagy csak egyszerűen az „előző” és „következő” gombokkal elnavigálhatunk a kérdésről.
- Függetlenül attól, hogy írtunk-e választ a válasz mezőbe, a helyes választ megtekinthetjük a „Elküld” gomb segítségével.

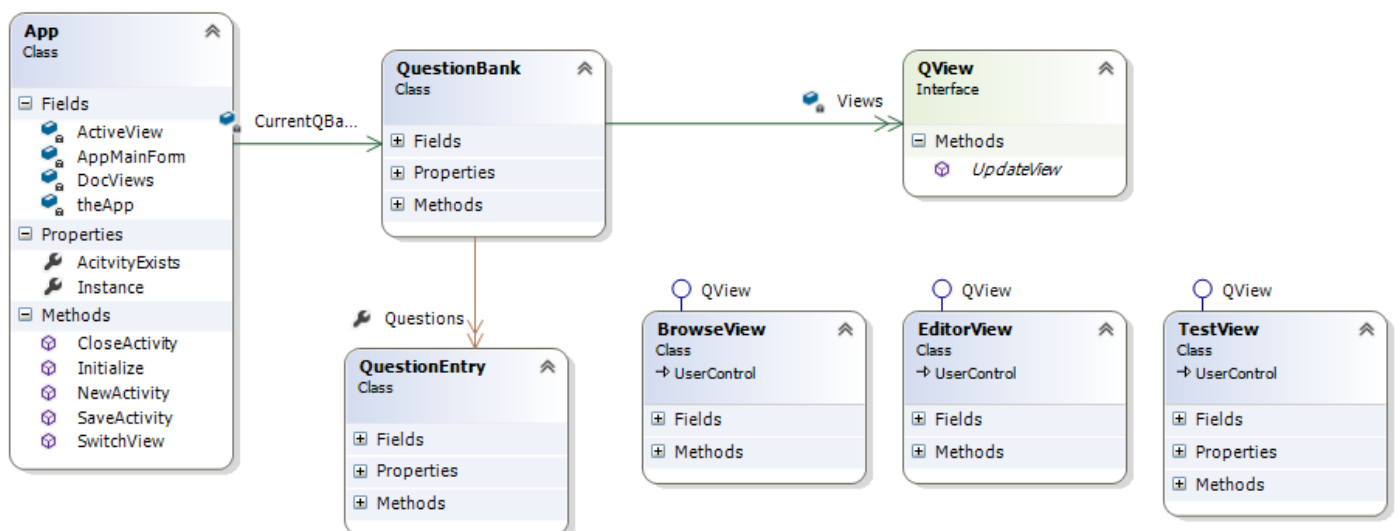
- Alapesetben a kérdéseket sorban tekinthetjük meg a böngésző módban. A kérdések sorrendje a kérdésbankba felvitt sorrend. Lehetőség van azonban arra is, hogy a kérdéseket véletlenszerűen böngésszük. A „Véletlenszerű kérdések” bepipálásával véletlen sorrendben ellenőrizhetjük tudásunkat.

Teszt mód

A funkció segítségével saját tesztet készíthetünk a kérdésekből. A kérdések számát, illetve a rendelkezésre álló időt mi választhatjuk meg.

- A kérdések számának természetesen 1 és N közé kell esnie, ahol N a kérdésbank kérdéseinek száma.
- A teszthez rendelkezésre álló időt másodperc pontossággal adhatjuk meg.
- Az idő lejártá után, vagy a teszt befejezése esetén a program kiértékeli a kérdésekre adott válaszokat, és százalékosan, illetve egy diagram segítségével megjeleníti az elért eredményt

A szoftver működésének bemutatása



A szoftver Document-View architektúrát alkalmaz. Dokumentumként a kérdésbank szolgál, amit a QuestionBank osztály valósít meg. A kérdések egyes nézetei pedig a különböző QView-k (BrowseView, EditorView, TestView).

Az App osztály az alkalmazás logikát valósítja meg. A main form az App-ot példányosítja. Az App osztályon keresztül hozza létre és hívja meg az aktuális kérdésbankon végzendő műveleteket, illetve a GUI elemei is így hívnak meg függvényeket.

A dokumentum

A QuestionBank osztály feladata a kérdésbank kezelése. Tartalmaz egy listát a nézetekről (QView), illetve egy listát, melyben a kérdés bejegyzések szerepelnek. A saját kérdésbankjának létrehozása, mentése, betöltése is az ő feladata. Kezeli továbbá az egyes nézetek adminisztrálását, csatolását, lecsatolását.

Az egyes kérdés bejegyzések (QuestionEntry) külön osztályba rendezettek. Az osztály feladata a kérdésbejegyzés, mint logikai egység kezelése. A legfontosabb tagváltozói így: a kérdés, válasz és a kulcsszó mezők. Kiemelendő emellett még a kérdéshez tartozó egyedi azonosító (ID). Ez az osztály tartalmaz egy IDGenerator statikus változót, melynek köszönhetően minden kérdéshez más azonosítót tudunk rendelni. Az egyedi azonosítót felhasználva, a későbbiekben a bejegyzések kezelése egyszerűbbé válik.

A nézetek

Az egyes nézetek a QView osztálynak interfészei. Közös bennük így az *UpdateView()* függvény, melynek feladata a nézet frissítése. A nézetek közti különbséget a következőkben tekintjük át.

EditorView

A szerkesztő nézet feladata, hogy a kérdésbank bejegyzéseket lehessen hozzáadni, törölni, illetve módosítani. A kérdésbejegyzések tehát dinamikusán módosulhatnak a program futása közben, így a bejegyzések megjelenítését programozottan valósítottam meg, ezért ennél a nézetnél egy TableLayoutPanel control használtam. A kérdések tehát táblázatosan jelennek meg a felhasználó számára. Az UpdateView() függvény hívásakor a táblázat programozottan újrarajzolódik, így tartva azt konzisztensen.

Böngészés Teszt indítása Szerkesztés

Kérdés sorszáma

Előző 10
Következő 10
31
Mutasd

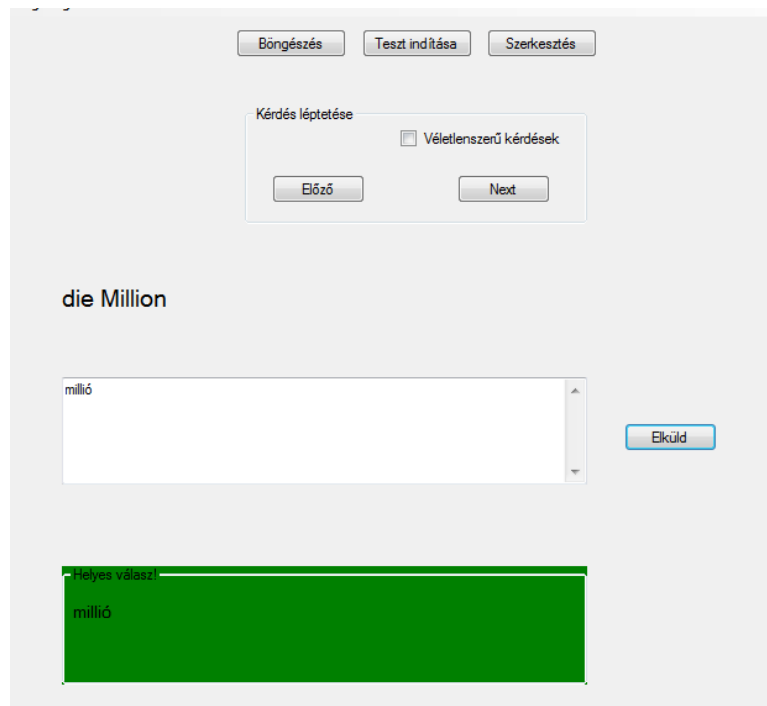
Kérdés	Válasz	Kulcsszavak	
das Viertel	negyed(ik)	negyed(ik)	Törés
die Währung	valuta	valuta	Törés
die Werbung	hirdetés	hirdetés	Törés
die Zeitschrift	magazin	magazin	Törés
der Zustand	állam	állam	Törés
ausgeben	költeni	költeni	Törés
bekommen	kapni	kapni	Törés
brauchen	szükségeltetik/kell	szükségeltetik/kell	Törés
finden	találni	találni	Törés
funktionieren	funkcionálni	funkcionálni	Törés
<div style="height: 40px;"></div>	<div style="height: 40px;"></div>	<div style="height: 40px;"></div>	Hozzáadás

Editor view

BrowseView

A böngésző nézet felépítése statikus. A nézet tartalma néhány címkét (Label), melyek a szöveget jelenítik meg (pl.: az aktuális kérdés). Emellett még tartalmaz egy szövegbeviteli mezőt, ahol a kérdésre a választ adhatjuk, és ehhez tartozik egy küldés gomb. A kérdéseket léptető gombokkal lehet váltani, illetve tartalmaz egy „Véletlenszerű kérdések” checkbox-ot is, mely aktiválása esetén a következő kérdés véletlenszerűn választódik, a *System.Random* osztály segítségével.

A beviteli mezőbe beírt válasz helyességét az AnswerCheck statikus osztály ellenőrzi. A válasz helyességének ellenőrzése után megjelenik a válasz kiértékelése, illetve a helyes válasz.



Browse view

TestView

Ez a nézet tesztek írására szolgál. Fontos, hogy a teszt indítása után a konzisztens állapot fenntartása érdekében a teszt kérdései nem módosulhatnak. Ezért úgy valósítottam meg a tesztet, hogy az aktuálisan futó teszt kérdéseit külön változóban tárolja a kód. Ennek köszönhetően az *UpdateView()* függvény nem írja felül / nem módosítja az aktuálisan futó teszt kérdéseit, azonban a teszt befejezése után, ha a felhasználó új tesztet indít, akkor már a friss értékeket használva generálja a kód az új tesztet.

Paraméteresen megadható a teszt kérdéseinek száma, illetve a teszt hossza. A start gomb lenyomása után a *System.Random* osztályt felhasználva véletlenszerűen kiválasztunk a megadott számnak megfelelő kérdést a kérdésbankból. Ezt követően egy Timert felparaméterezünk a megfelelő értékekkel, majd elindítjuk. A hátralévő időt számláló címkét másodpercenként frissíteni kell, ezt a timer tick eseményére feliratkozva tesszük meg.

A tesztet követően az *AnswerCheck* statikus osztályt meghívva a válaszokat kiértékeljük, majd megjelenítjük az eredményt numerikusan és grafikusan is. A grafikus megjelenítéshez a GDI+ szolgáltatásait használtam. A százalékos eredmény egy tortadiagramon jelenik meg.

Teszt paramétereit

Rendelkezésre álló idő (mp)

240

Kérdések száma

7


Start!

Hátralévő idő

02:54

Befejezés

A teszt véget ért. A helyes válaszok száma : 4/7 (57,14 %)

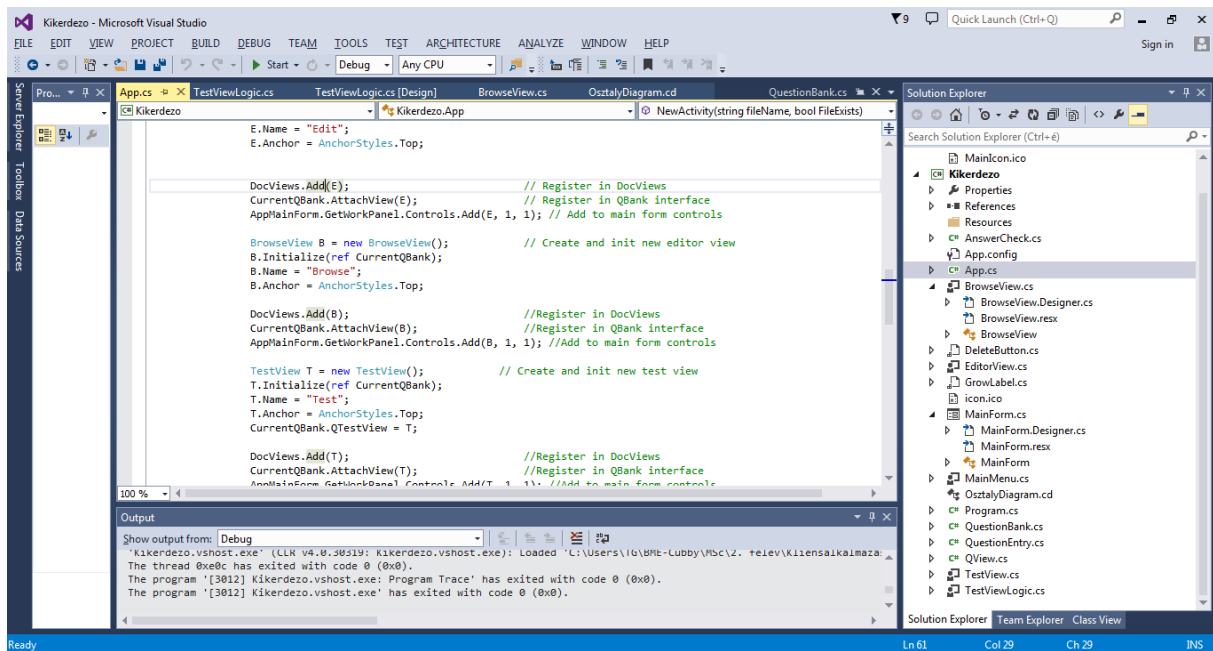


das Erdgeschoss	földszint	Helyes válasz! földszint
die Farbe	x	Helytelen válasz! szín
die Abteilung	osztály	Helyes válasz! osztály
das Angebot	ajánlat	Helyes válasz! ajánlat

Test view

Függelék

Felhasznált szoftverek



Microsoft Visual Studio 2013

master		BugFix + SaveFile	
History		talaberg	7db1bc9
▼ EditView_OldVersion 19 days ago by talaberg		► Kikerdezo\DataExample - Copy.csv	3
▼ BugFix + SaveFile 19 days ago by talaberg		► Kikerdezo\Kikerdezo\App.cs	21
▼ TestView implemented 21 days ago by talaberg		► Kikerdezo\Kikerdezo\BrowseView.cs	64
▼ TestView skeleton done 21 days ago by talaberg		► Kikerdezo\Kikerdezo\BrowseView.Designer.cs	83
▼ Answer check implemented 21 days ago by talaberg		► Kikerdezo\Kikerdezo\EditorView.cs	26
▼ Browse, edit checkpoint 21 days ago by talaberg		► Kikerdezo\Kikerdezo\MainForm.cs	37
▼ ViewUpdate Debug 21 days ago by talaberg		► Kikerdezo\Kikerdezo\MainForm.Designer.cs	60
▼ Menu, Open, Close 21 days ago by talaberg		► Kikerdezo\Kikerdezo\MainForm.resx	3
▼ BrowseView, EditView skeleton 22 days ago by talaberg		► Kikerdezo\Kikerdezo\QuestionBank.cs	61
▼ FileOpen implemented, BrowseView started 24 days ago by talaberg		► Kikerdezo\Kikerdezo\TestViewLogic.cs	57
▼ Slight: Q.A.K variables and properties added 2 months ago by talaberg			
▼ Project added 2 months ago by talaberg			

A GitHub felhasználói felülete