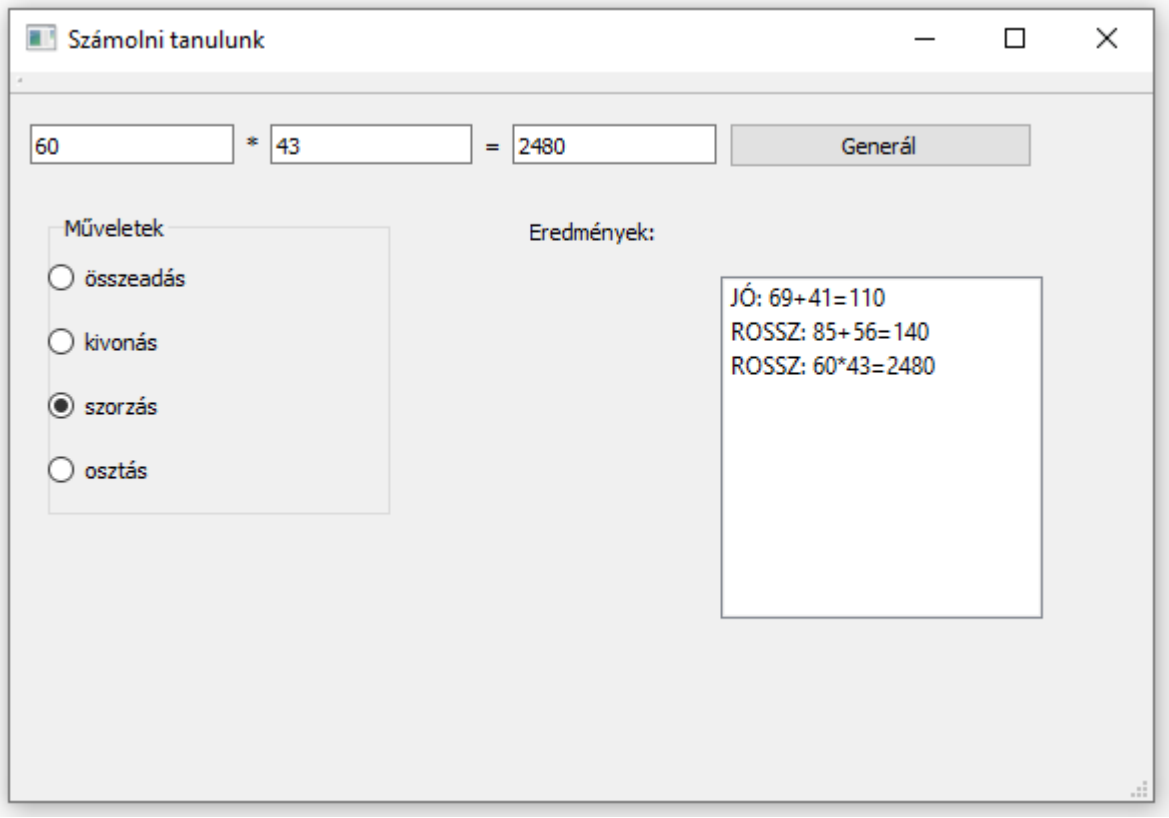
Eseményvezérelt alkalmazások fejlesztése

2. gyakorlat

Cél: A második gyakorlaton olyan egyablakos alkalmazásokat hozunk létre, hol nincs még sok vezérlőnk.

Kisdiák számol

Készítsük el az alábbi ábrának megfelelő ablakot!



1. Amikor műveletek rádiógombjai közül valamelyiket benyomjuk, akkor az a kiválasztott művelet jele jelenjen meg operandusok között.

2. A nyomógomb lenyomásának hatására generáljon véletlen 0 és 100 közé eső számokat, és írja ki azokat az operandusok szövegdobozaiba. Ezután a rádiógombokat ne lehessen benyomni, a nyomógomb új felirata legyen „Értékel”, a válasz szövegdoboza legyen üres és szerkeszthető.

3. Miután beírjuk a választ és lenyomjuk a nyomógombot, értékeljük ki az eredményt. A helyes eredményt fűzzük a listadoboz végére egy JÓ vagy ROSSZ megjegyzés kíséretében (lásd ábra). Legyen a válasz szövegdoboza újra csak olvasható, a rádiógombokat lehessen újra benyomni, és a nyomógomb felirata legyen ismét „Generál”.

4. Tegye lehetővé azt, hogy a válasz begépelése után az <Enter> lenyomásának hatására is végrehajtódjon az előző bekezdésben leírt tevékenység. Végezzen adatellenőrzést is! Hibás formájú válasz esetén adjon figyelmeztetést!

A felhasználni kívánt elemek: Label, Line Edit, Push Button, Radio Button, a rádiógombok számára egy Group Box, az eredmények számára pedig egy List Widget. A felső sort és a 4 rádiógombot érdemes rendezőkbe rakni, hogy szépen nézzen ki az alkalmazás. A Line Editeknek, labeleknek és a rádiógomboknak érdemes beszédes nevet adni (objectName property), hogy egyértelműen lehessen rájuk hivatkozni (pl. rdbSum, lineEditRes, …). Az egyik rádiógomb legyen kiválasztva (checked property) és a neki megfelelő művelet jele legyen a műveletes labelben. Az összes Line Edit legyen csak olvasható (readOnly property).

A rádiógombok kezelhetők absztakt gombokként Qt-ben, ez azt jelenti, hogy rendelhetünk hozzájuk *clicked()* eseményt a szokásos módon, amiben csak a művelet labelének szövegét változtatjuk meg a setText() metódusa segítségével.

A gombunknak is szüksége lesz egy eseménykezelőre, a lenyomást kezeljük. Az eseménykezelőben attól függően, hogy épp milyen állapotban van a program, két ág lehetséges: generálási szakasz és értékelési szakasz. Ezt például a gomb feliratának vizsgálatával tudjuk kezelni. Amennyiben a gomb felirata „Generál”, a két LineEditben elhelyezünk véletlenszámot a setText() metódus segítségével. A rádiógombokat letiltjuk a setEnabled() metódusuk segítségével. Az eredmény Line Edit tartalmát töröljük a clear() metódusával és szerkeszthetőre állítjuk (setReadOnly() metódus). Számról szövegre alakítás: QString::number(rand()%101)

A következő lépésben a gomblenyomás eseménykezelőjében megírjuk az else ágat: amikor a felirata nem „Generál” (hanem „Értékel”). Itt szükség lesz a QString intté konvertálására és azt is nézni kell, hogy a válasz szám-e vagy sem. Erre kiváló eszköz a toInt() metódus (a kezelni kíván Line Edit-re most LineEditRes néven hivatkozunk):

```
bool ok;
int c = ui->lineEditRes->text().toInt(&ok);
```

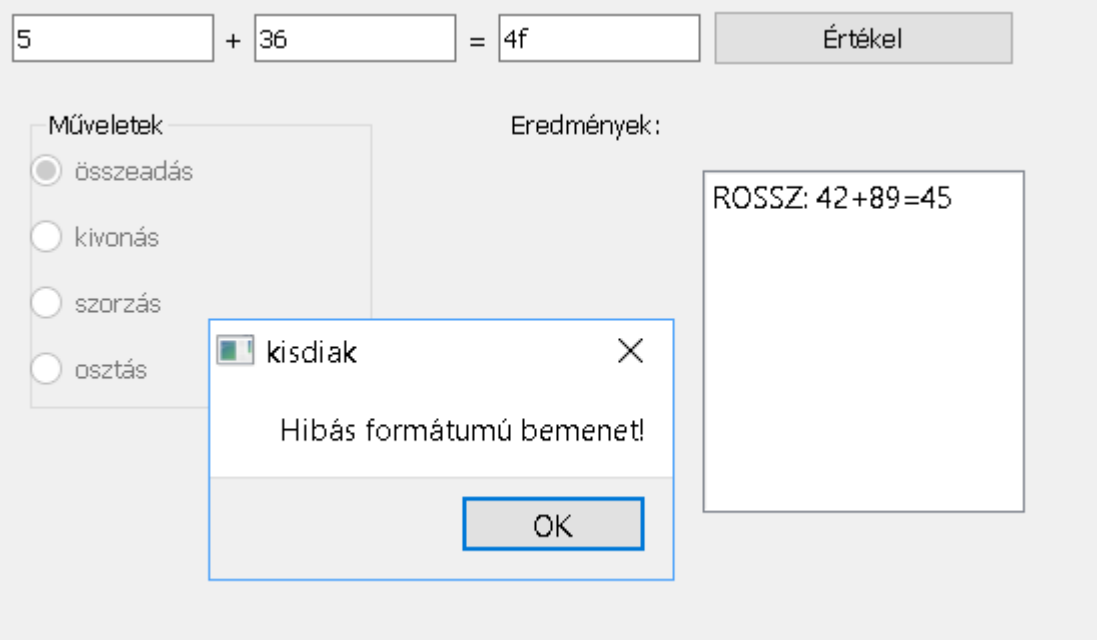
e két sor lefutása után az *ok* változó hamis értéket kapott, ha sikertelen a konverzió. Sikeres esetén pedig *ok* igaz lesz.

Sikeres konverzió után az eredmény vizsgálata következik. Jó eredmény esetén a megfelelő szöveget a listWidget addItem(”JÓ: …”) metódusa fogja elhelyezni a szövegdobozban.

Az utolsó előtti lépés egy Message Box feldobása, ha hibás formátumú a válasz (mert nem sikerült a konverzió). Ebben az esetben jó lenne az ifeket úgy rendezni, hogy lehetőség legyen az eredmény javítására és a listWidgetben ne maradjon nyoma az elgépelésnek.

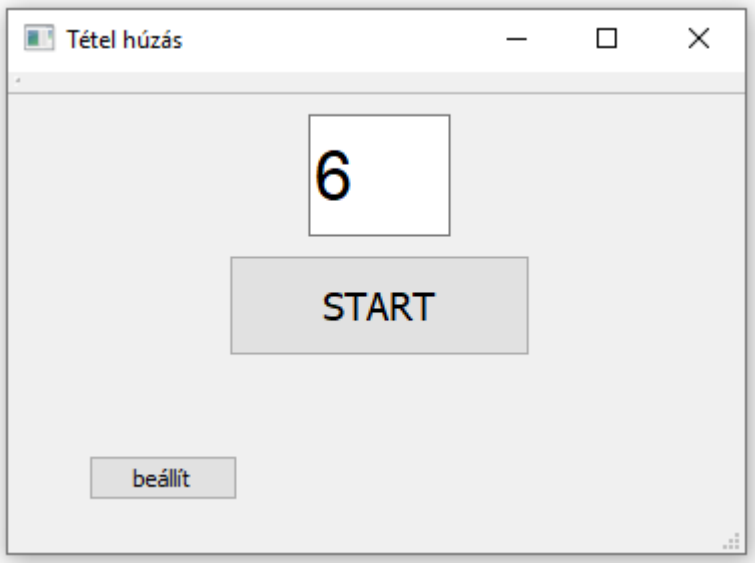
A legutolsó lépés az eredmény begépelésének Enterrel való kezelése. Ezt az osztály konstruktorában tudjuk megtenni a connect utasítás segítségével: a Line Edit returnPressed() szignálját kötjük össze a form on_pushButtonClicked() eseményével:

```
connect(ui->lineEditRes,
SIGNAL(returnPressed()),this,SLOT(on_pushButton_clicked()));
```



Sorsolás

Készítsük el az alábbi ábrának megfelelő ablakot!



1. A START feliratú gomb lenyomására a szövegdobozban az előre megadott tételsorszámok váltakozzanak véletlenszerűen és gyorsan. A nyomógomb felirata változzon STOP-ra. A STOP feliratú nyomógomb lenyomásakor a szövegdoboz „merevedjen meg", és a legutoljára generált tételsorszámot mutassa.

2. Tegye lehetővé újabb nyomógomb (beállít) segítségével, hogy az alkalmazásban állíthassuk be a tételek maximális számát! (Ennek eszköze legyen egy láthatóvá váló szövegdoboz.)

3. Biztosítsa, hogy amíg nincs megfelelő módon (adatellenőrzés!) megadva a tételek maximális száma, addig ne lehessen elindítani a tétel-húzást!

4. Gondoskodjon arról, hogy egy már kiválasztott tétel-sorszámot csak megadott számú (ezt is az alkalmazásban lehessen beállítani) sorsolás után lehessen újra kiválasztani! Ez a szám kisebb kell legyen, mint a tételek száma.

A felhasználni kívánt elemek: Label, Line Edit és Push Button. A beállításra használt eleinte rejtett labeleket és Line Editeket érdemes rendezőbe rakni. A Line Editeknek és gomboknak érdemes beszédes nevet adni, hogy egyértelműen lehessen rájuk hivatkozni (pl. lineEdit_no, lineEdit_max, pushButton_START,…). A beállító Line Editek és labelek legyenek rejtve, ezt a konstruktorban tudjuk beállítani, minden egyes vezérlőre meghívva a saját hide() metódusát. A beállító Line Editek kapjanak kezdőértéket (pl. 20 tétel és 0 kihagyás). A gombokhoz rendeljünk eseménykezelőket.

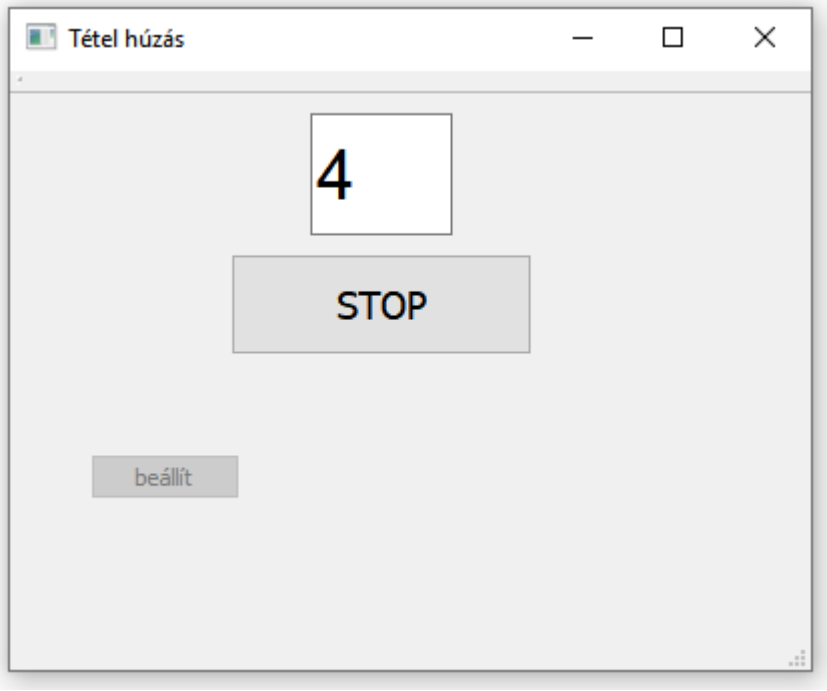
Az alkalmazásnak szüksége lesz néhány privát adattagra: egy QTimer-re (amit a START gomb indít el és állít le), két egész számra tárolni a beállító Line Editek tartalmát (nem kötelező, de megspórolhatunk vele egy csomó konverziót), két vektorra tárolni, hogy mely számok közül lehet sorsolni (_nums), illetve az egyes számoknak hány húzásnyit kell várniuk még, hogy sorsolhatók legyenek (_gaps). Két segédeljárás (reduce() és search()) teheti strukturáltabbá a programot: az egyik csökkenti a _gaps minden pozitív elemét sorsolás után, valamint visszahelyezi a _nums vektorba azokat a számokat, amik újra húzhatóvá váltak. A másik megkeresi, hogy egy bizonyos szám hanyas indexű a _nums vektorban (kiválasztás programozási tétel).

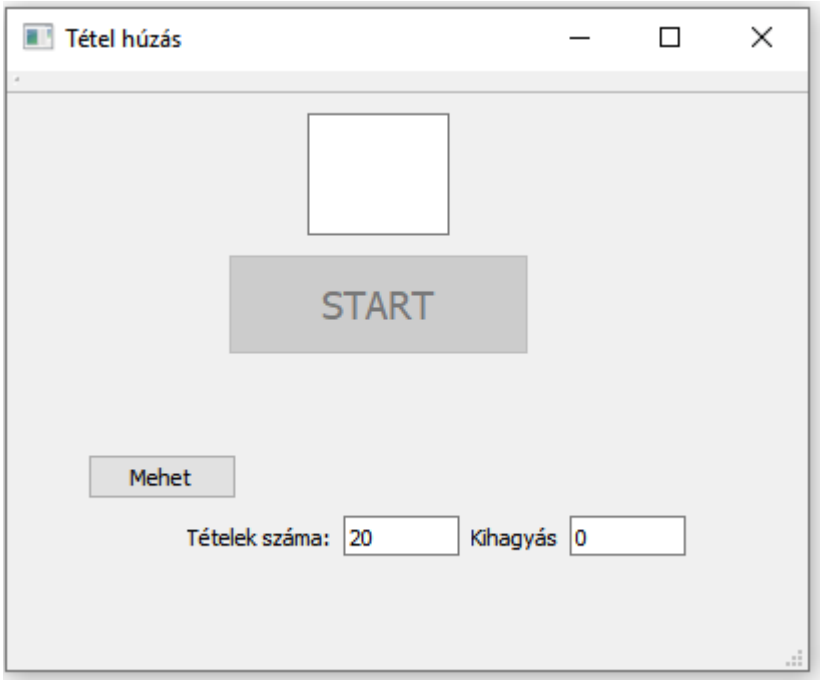
A QTimer számára érdemes írni egy privát eseménykezelőt, amivel a tételszámot kijelző szövegdoboz tartalmát lehet módosítani (changeNumber()). A múlt órai digitális óra mintájára tehetjük ezt meg. A changeNumber törzse nagyon egyszerű, a _nums vektor egy véletlenedik elemét írja ki a kijelzőbe.

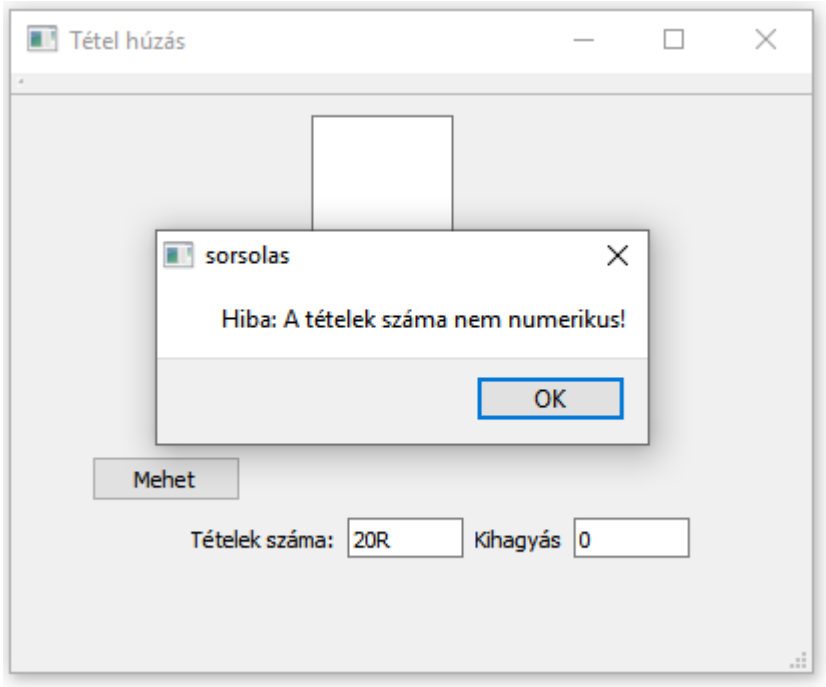
A konstruktorban inicializáljuk az egyes vezérlőinket és a változóinkat (és a random generátort). Pl. a két egész változóba (_max és _gap) belementjük a tételek számát és a kihagyást, amit a két Line Editben meghatároztunk. A _gaps és _nums vektorokat feltöltjük csupa 0-val, illetve 1-től _max-ig egészekkel. A beállításra szánt Line Editeket és labeleket itt tudjuk elrejteni.

A következőkben a két gomb lenyomását kezeljük. Elsőként a START gombot nézzük! Ahogy az előző alkalmazásban, itt is két ágra bontjuk az eseménykezelőt: amikor sorsol és amikor megállt. Megint a gomb felirata alapján döntjük el, hogy épp melyik ágat hajtsa végre. Amennyiben a felirata „START”, csak az időzítőt kell elindítania, a beállító gombot letiltania és a feliratot átállítania, minden más máshol van leprogramozva. Amennyiben a felirata „STOP”, állítsa le az időzítőt, szabadítsa fel a beállítógombot, mentse el a kisorsolt tétel számát, csökkentse a _gaps minden elemét, állítsa be a sorsolt tételnél a _gaps megfelelő értékét, végül törölje a _nums vektorból a sorsolt tételt. Persze ezek egy részét csak akkor kell végrehajtani, ha a kihagyás mértéke nagyobb, mint 0.

Az utolsó feladatunk a beállít gomb eseménykezelőjét megírni. Ezt is két ágra bonjuk: amikor láthatók a hozzá tartozó vezérlők és amikor nem. Érdemes ezt is a gombfelirat alapján ketté szedni. Amikor a felirata még „beállít”, láthatóvá kell tenni a beállításra szánt Line Editeket és labeleket, le kell tiltani a START gombot és lecserélni a feliratát. Egyéb esetben az a feladat, hogy a két Line Edit tartalmát feldolgozzuk. Először beolvassuk őket egy változóba azt is figyelve, hogy a toInt() metódus szerint sikeres volt-e a konverzió, illetve, hogy a tételek száma 1-nél nagyobb legyen, a kihagyás mértéke nemnegatív és a kihagyás mértéke legyen a tételek számánál kisebb. Amennyiben ez mind teljesül, érdemes újra generálni a _nums és a _gaps vektorok tartalmát is. Ne felejtsük ugyanitt felszabadítani a START gombot, elrejteni a beállító szöveges mezőket és visszaállítani a beállító gomb feliratát. Hibás input esetén egy Message Box-szal jelezzük, hogy a felhasználó helytelen adatot adott meg. Itt érdemes többirányú elágazást létrehozni és külön jelezni minden egyes hibát.




Tétel húzás
START
Mehet
Tételek száma: 20
Kihagyás 0


Tétel húzás
sorsolas
Hiba: A tételek száma nem numerikus!
OK
Mehet
Tételek száma: 20R
Kihagyás 0