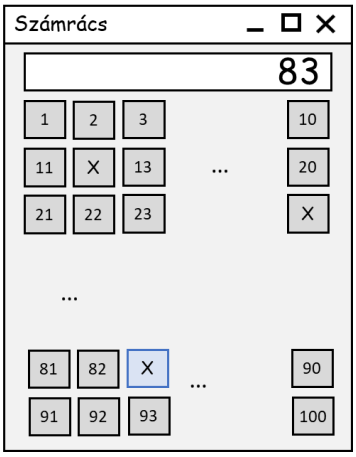
4. hét

Cél: Olyan alkalmazások készítése, amelyek sok, azonos típusú, dinamikusan felhelyezett vezérlőt kezelnek.

1. Számrács

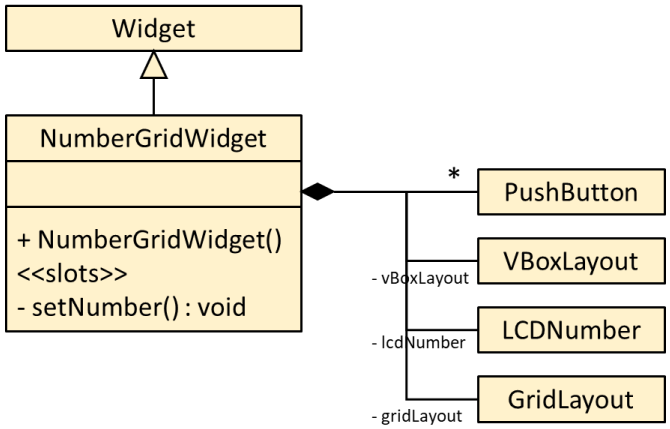
Készítsünk egy olyan alkalmazást, amely egy 10×10-es rácsban száz darab 1-től kezdődően sorszámozott nyomógombot jelenít meg. Egy nyomógombra kattintva a gomb sorszáma jelenjen meg egy központi kijelzőn, a gombon pedig ettől kezdve az ‚X' felirat látszódjon. Később az ilyen feliratú gombokra kattintva már ne történjen semmi.



Projekt létrehozása

Hozzunk létre egy *Qt Widget Application* projektet. A főablakunk ősosztálya legyen a *QWidget*, amelynek grafikus felületét a programkódból építjük fel, ezért a *QtCreator*-ban ne pipáljuk be a *Generate form* lehetőséget.

Osztály diagram



A NumberGridWidget() konstruktor

A konstruktorban hozzuk létre és helyezzük el az ablak (widget) felületen a vezérlőinket.

1. Adjuk meg az ablakunk fejlécének szövegét.

```
setWindowTitle(tr("Számrács"));
```

2. Rögzítsük a méretét.

```
setFixedSize(400,400);
```

3. Példányosítsuk az LCD kijelzőt, a rács-elrendezőt, és a függőleges elrendezőt.

```
_lcdNumber = new QLCDNumber();
_gridLayout = new QGridLayout();
_vBoxLayout = new QVBoxLayout();
```

4. Hozzunk létre egy dupla for ciklussal 10×10 darab nyomógombot, és helyezzük el ezeket rács-elrendezőben (*addWidget(button, i, j)*). Egy nyomógomb felirata az *i*10+j+1* értéke legyen, amennyiben a nyomógomb a rács *i*-dik sorának *j*-dik oszlopában van (0-tól sorszámozunk). Ehhez használjuk a nyomógomb azon konstruktorát, amelynek első paramétere a gomb felirata, második pedig a nyomógombot tartalmazó ablakra mutat (*this*). Rendeljük (*connect*) a nyomógombokhoz ugyanazt az eseménykezelőt: *setNumber()*.

```
for (int i = 0; i < 10; ++i) {
   for (int j = 0; j < 10; ++j){
      QPushButton* button=new QPushButton(QString::number(i*10+j+1),this);
      _gridLayout->addWidget(button, i, j);
      QObject::connect(button, SIGNAL(clicked()), this, SLOT(setNumber()));
   }
}
```

5. Vegyük fel a függőleges elrendezőbe az LCD kijelzőt (*addWidget()*) és a rács-elrendezőt (*addLayout()*), majd rendeljük a függőleges elrendezőt az ablakunkhoz (*setLayout()*).

```
_vBoxLayout->addWidget(_lcdNumber);
_vBoxLayout->addLayout(_gridLayout);
setLayout(_vBoxLayout);
```

A setNumber() eseménykezelő

Egyetlen eseményt kell lekezelnünk: a nyomógombra kattintást: *setNumber()*.

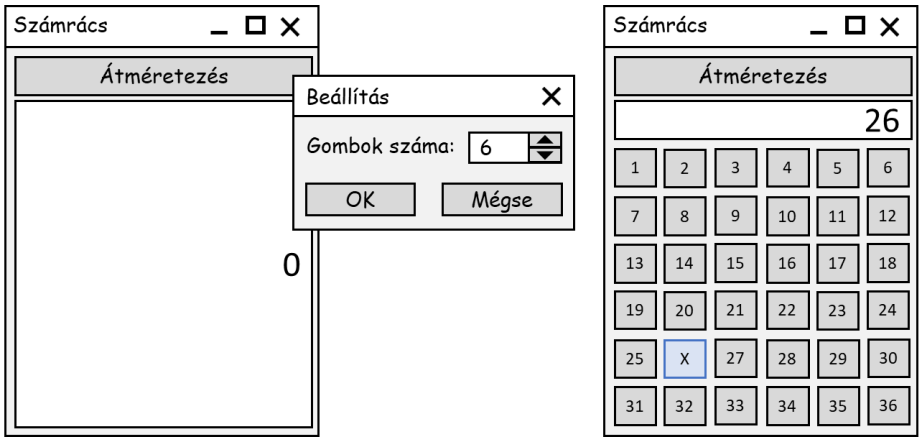
Ebben először azonosítjuk a szignált küldő nyomógombot. Ennek címét a *QWidget sender()* metódusától kaphatjuk meg, de a típusát castolnunk kell, hogy lekérdezhessük a nyomógomb feliratát.

```
QPushButton* senderButton = qobject_cast<QPushButton*>(sender());
```

Ha a felirat (*senderButton->text()*) már 'X', akkor nem kell tenni semmit. Ha a felirat nem 'X', akkor a feliratot kiírjuk (*display()*) az LCD kijelzőbe, majd a gomb feliratát átírjuk 'X'-re.

2. *Számrács méretező dialógussal*

Módosítsuk az előző alkalmazást úgy, hogy a nyomógombok rácsát futás közben át lehessen méretezni. Kezdetben a rács legyen üres. Egy külön nyomógombbal hozhassunk fel egy előugró ablakot, amelyben beállíthatjuk az új méretet. Ennek hatására a rács méreteződjön át, és minden nyomógombján látszódjék annak sorszáma.

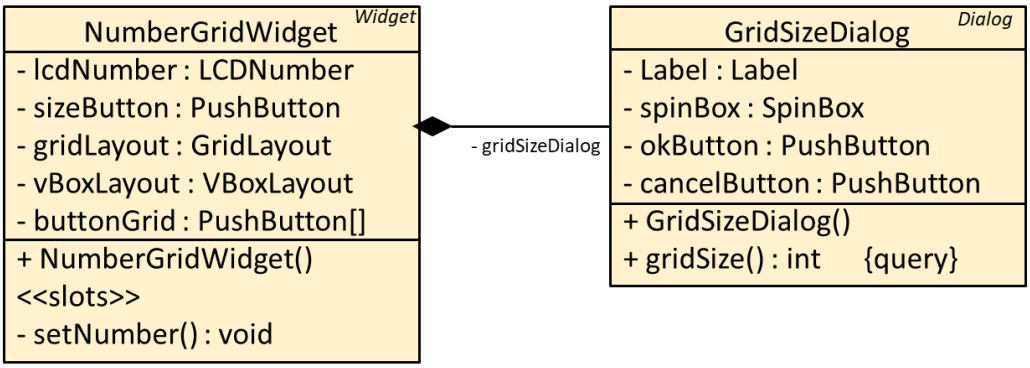


Projekt létrehozása

Vegyünk hozzá az előző projektünkhöz egy új osztályt, amelyet a *QDialog* osztályból származtatunk. Ennek segítségével a rács méretét fogjuk beállítani. Minden vezérlőt a programkódból építünk fel, azaz nem használjuk a *QtDesigner*-t.

Osztály diagram

Definiáljuk az alábbi két osztályt:



A NumberGridWidget() konstruktor

A konstruktor itt is az osztály adattagjait példányosítja (két újabb taggal együtt: *gridSizeDialog*, *sizeButton*), de nem itt kerül sor a rács-szerkezetbe elhelyezett nyomógombok létrehozására: ezt egy külön metódusba szervezzük ki (*resizeGrid()*).

Meg kell adni viszont azt, hogy egyrészt a *_sizeButton* megnyomásakor a *_gridSizeDialog exec()* metódusa hívódjon meg (azaz a dialógus ablak modálisan megjelenjen), másrészt ha a dialógust az *okButton* gomb lenyomásával hagyjuk el, akkor hívódjon meg a *resizeGrid()* metódus, amelyik az új nyomógomb-rácsot elkészíti.

```
connect(_sizeButton,     SIGNAL(clicked()), _gridSizeDialog,SLOT(exec()));
connect(_gridSizeDialog,SIGNAL(accepted()),this,            SLOT(resizeGrid()));
```

A resizeGrid() eseménykezelő

Ez a metódus létrehozza és feliratozza a rács nyomógombjait.

Gondoskodik arról is, hogy mielőtt új rácsot épít fel új nyomógombokból, megszűntesse a régieket. Ehhez kell a *QVector<QPushButton*>* típusú *_buttonGrid* gyűjteményben tárolni a dinamikusan létrehozott nyomógombok címeit.

```
    foreach(QPushButton* button, _buttonGrid)
    {
        _gridLayout->removeWidget(button);
        delete button;
    }
    _buttonGrid.clear();
```

Ezután építi fel az új nyomógomb-rácsot úgy, ahogy azt az előző feladat megoldásában láttuk. Az egyetlen különbség az, hogy a rács méretét a *gridSizeDialog* objektumtól kell lekérdezni (*gridSize()*), és minden újonnan létrehozott nyomógomb címét el kell tárolni a *_buttonGrid* -ben.

A GridSizeDialog() konstruktor

A rácsméretet beállító dialógus ablak konstruktorában példányosítsuk és inicializájuk az adattagokat. A vezérlők elhelyezéséhez használjunk elrendezőket.

Itt kell beállítani azt is, hogy a két nyomógomb speciális funkcióval bírjon: bezárják az ablakot, és az egyik egy accepted(), a másik egy rejected() szignált emittáljon. A *Számrács* osztályban az accepted() szignálhoz rendeltünk eseménykezelőt.

```
connect(_okButton,     SIGNAL(clicked()), this, SLOT(accept()));
connect(_cancelButton, SIGNAL(clicked()), this, SLOT(reject()));
```