Programming Widget Layout

Using the gained knowledge to create forms.

QHBOXLayout :

The QHBoxLayout class lines up widgets horizontally.

Exercise 1:

dialog.h :

```
class Dialog : public QWidget
{
public:
   explicit Dialog(QWidget *parent =nullptr) ;

protected:
   void createWidgets();
   void placeWidgets();
   void makeConnexions();

protected:
    QLabel* name;
    QLineEdit* line;
    QPushButton* button;
};
```

dialog.cpp:

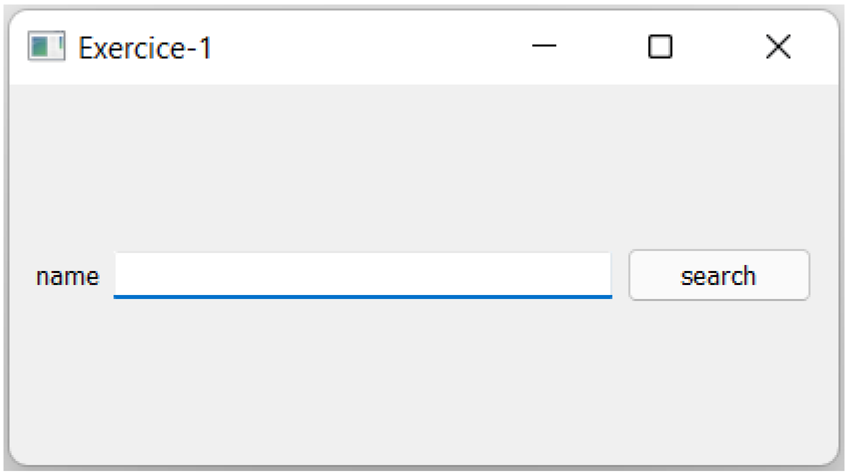
```
Dialog::Dialog(QWidget *parent) : QWidget(parent)
{
   createWidgets();

   placeWidgets();

   makeConnexions();
}
```

```
void Dialog::createWidgets(){
   name = new QLabel ("name");
   line = new QLineEdit;
   button = new QPushButton ("search");
}

void Dialog::placeWidgets(){
   auto layout = new QHBoxLayout;
   setLayout(layout);

   layout-> addWidget (name);
   layout-> addWidget (line);
   layout-> addWidget (button);

}
void Dialog ::makeConnexions(){
   connect(button,&QPushButton::clicked,qApp,&QApplication::exit);
}
```

main.cpp:

```
int main(int argc, char *argv[])
{
   QApplication a(argc, argv);

   Dialog *D=new Dialog;

   D->show();

   return a.exec();
}
```

example :



A QHBoxLayout example.

Nested Layouts:

By the term Nested we mean one Layout inside of another Layout.

The goal of the exercise is to learn to analyse the construction of a form and then code it using Nested layouts.

Exercise 2:

layouts.h:

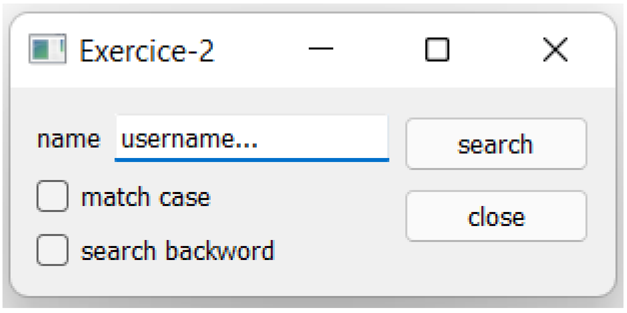
```
class layouts : public QWidget
{
public:
   explicit layouts(QWidget *parent =nullptr) ;

protected:
   void createWidgets();
   void placeWidgets();
   void makeConnexions();

protected:
    QLabel* name;
    QLineEdit* Nick;
    QPushButton* search;
    QPushButton* close;
    QCheckBox* matchcase;
    QCheckBox* backword;
    QLineEdit* line;
};
```

layout.cpp :

```
layouts::layouts(QWidget *parent) : QWidget(parent){

       createWidgets();

       placeWidgets();

       makeConnexions();

}
void layouts::createWidgets(){
   name = new QLabel ("name");
   Nick = new QLineEdit ("nick");
   search = new QPushButton ("search");
   close = new QPushButton ("close");
   matchcase = new QCheckBox ("match case");
   backword = new QCheckBox ("search backword");
   line = new QLineEdit("username...");
}

void layouts :: makeConnexions(){
   connect((close), &QPushButton :: clicked ,
           qApp ,& QApplication :: exit );
}
void layouts :: placeWidgets(){

       //main layout
   auto mainLayout = new QHBoxLayout;
   auto rightLayout = new QVBoxLayout;
   auto leftLayout = new QVBoxLayout;
   auto leftUpLayout = new QHBoxLayout;

   setLayout(mainLayout);
   mainLayout-> addLayout(leftLayout);
   mainLayout->addLayout(rightLayout);
   leftLayout-> addLayout(leftUpLayout);

   leftUpLayout->addWidget(name);
   leftUpLayout->addWidget(name);
   leftUpLayout->addWidget(line);
   leftLayout->addWidget(matchcase);
   leftLayout->addWidget(backword);

   rightLayout->addWidget(search);
   rightLayout->addWidget(close);
   rightLayout->addSpacerItem(new QSpacerItem(10,10, QSizePolicy
:: Expanding));
}
```

main.cpp :

```
int main(int argc, char *argv[])
{
   QApplication a(argc, argv);

   auto D= new layouts;
   D->show();
   return a.exec();
}
```



Nested Layout.

Form:

bugrep.h:

```
class bugreport : public QWidget{
    Q_OBJECT
  public:
//constructor
    bugreport(QWidget *parent = nullptr);
    void createWidgets();
    void positionWidgets();
  private:
    QLineEdit* nameEdit ;
    QLineEdit* companyEdit ;
    QLineEdit* phoneEdit ;
    QLineEdit* emailEdit ;
    QLineEdit* problemEdit ;
    QTextEdit* summaryEdit ;
    QComboBox* reproducibilityCombo;
    QDialogButtonBox* buttonBox;
};
```

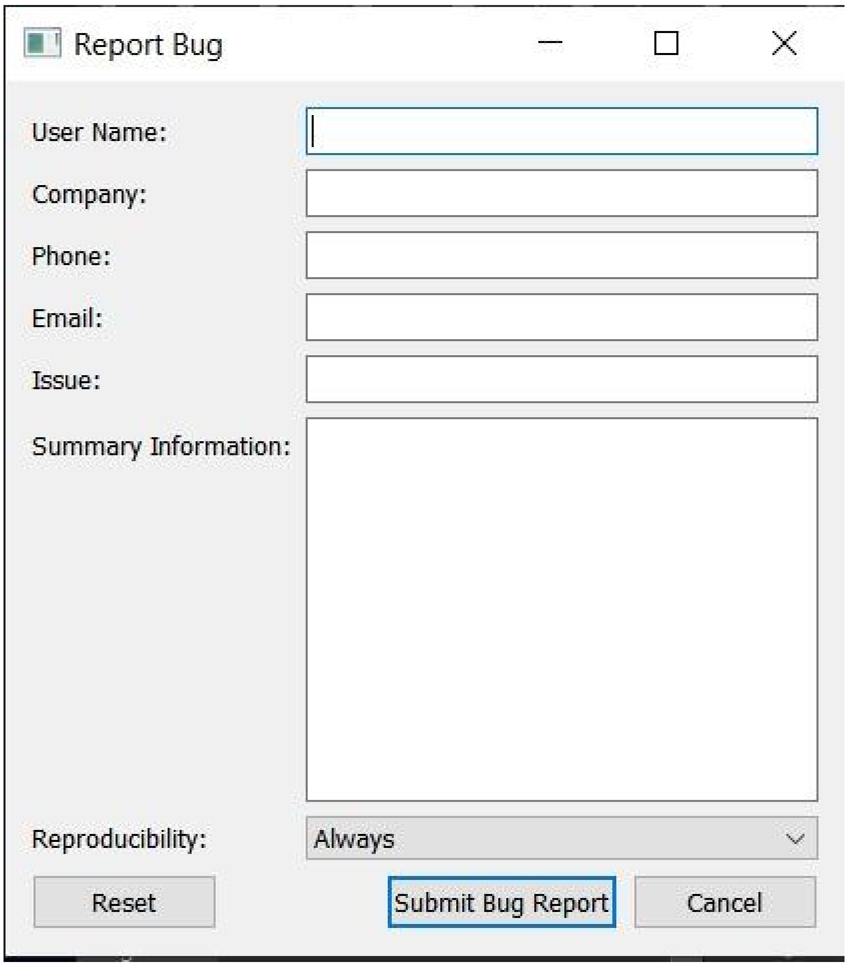
bugrep.cpp:

```
void bugreport::createWidgets() {
 nameEdit = new QLineEdit;
 companyEdit = new QLineEdit;
 phoneEdit = new QLineEdit;
 emailEdit = new QLineEdit;
 problemEdit = new QLineEdit;
 summaryEdit = new QTextEdit;
 reproducibilityCombo = new QComboBox;
 reproducibilityCombo->addItem(tr("Always"));
 reproducibilityCombo->addItem(tr("Sometimes"));
 reproducibilityCombo->addItem(tr("Rarely"));
 buttonBox = new QDialogButtonBox;
 buttonBox->addButton(tr("Submit Bug Report"),
QDialogButtonBox::AcceptRole);
 buttonBox->addButton(tr("Cancel"), QDialogButtonBox::RejectRole);
 buttonBox->addButton(QDialogButtonBox::Reset);
}
void bugreport::positionWidgets() {
 QFormLayout *layout = new QFormLayout;
 layout->addRow(tr("User Name:"), nameEdit);
 layout->addRow(tr("Company:"), companyEdit);
 layout->addRow(tr("Phone:"), phoneEdit);
 layout->addRow(tr("Email:"), emailEdit);
 layout->addRow(tr("Issue:"), problemEdit);
 layout->addRow(tr("Summary Information:"), summaryEdit);
 layout->addRow(tr("Reproducibility:"), reproducibilityCombo);
 QVBoxLayout *mainLayout = new QVBoxLayout;
 mainLayout->addLayout(layout);
 mainLayout->addWidget(buttonBox);
 setLayout(mainLayout);
}
bugreport::bugreport(QWidget *parent) : QWidget(parent) {
 createWidgets();
 positionWidgets();

 setWindowTitle(tr("Report Bug"));
}
```

main.cpp :

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    auto b =new bugreport;
    b->show();
    return a.exec();
}
```

example :



Dialog to report a form.

Grid Layout :

`calculator.h:`

```
class calculator : public QWidget
{
public:

    explicit calculator(QWidget *parent = nullptr);

    void creatingWdgets();
    void positionWidgets();
    void makeConnections();
private:

    QPushButton *buttons[10];
    QPushButton *bEnter;
    QLCDNumber *lcd;
    QVBoxLayout *mainLayout;
    QGridLayout *grid;
};
};
```

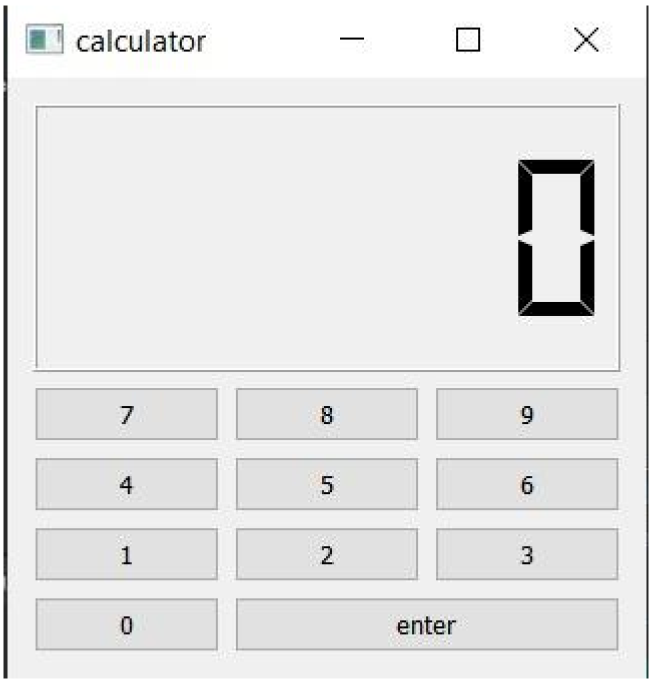
`calculator.cpp:`

```
void calculator ::creatingWdgets(){

        for(int i=0;i<10;i++){
           QString s = QString::number(9-i);
            buttons[i]=new QPushButton(s);
        }
        bEnter =new QPushButton("enter");

        lcd = new QLCDNumber();
        lcd->setSegmentStyle(QLCDNumber::Flat);

}
void calculator :: positionWidgets(){

     mainLayout = new QVBoxLayout();
     grid = new QGridLayout();
         int k = 0;
     for(int i=1;i<4;i++){
         for(int j=0;j<3;j++){
          grid->addWidget(buttons[k],i,2-j);
          k++;
          lcd->setMinimumHeight(80);
          lcd->setDigitCount(6);
         }
     }
      grid->addWidget(buttons[9],4,0);
      grid->addWidget(bEnter,4,1,1,2);
      mainLayout->addWidget(lcd);
      mainLayout->addLayout(grid);
      resize(300,300);
      setLayout(mainLayout);

}

calculator::calculator(QWidget* parent):QWidget(parent)
{
creatingWdgets();
positionWidgets();
}
void calculator :: makeConnections(){

}
```

- `main.cpp:`

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    auto calc = new calculator();
    calc->show();
    return a.exec();
}
```

example :



Calculator using the Grid Layout.

Made by:
Haytam El Ouarrat
Hassou Aymane