**Отчет по GUI**

Лабораторная работа №4

Студента гр. И-220а

Выполнил:

Курдюкова Александра

**Вариант -13**

**Задание.**

Окно (класс QWidget) приложения содержит 1 однострочный текстовый редактор (класс QLineEdit), 2 счетчика (класс QSpinBox), 3 надписи (QLabel). Надписи необходимо связать с редактором и счетчиками (метод setBuddy() – см. конспект), чтобы при наборе комбинации клавиш + передать фокус ввода редактору или счетчику. В текстовом редакторе должны автоматически отображаться числа, вводимые в “первом” счетчике, а значение данного счетчика должно соответственно обновляться и равняться числу, введенному в текстовом редакторе. Таким образом, в текстовый редактор можно вводить только целые числа. Второй счетчик задает текущую “правую” границу допустимого диапазона вводимых чисел (“левая” границапроизвольная) для редактора и первого счетчика. Для проверки на ввод данных можно использовать класс QValidator и его наследников (см. конспект). Вводимые числа в первом счетчике и в текстовом редакторе должны автоматически отображаться в заголовке окна приложения

**Код.**

**Widget.h**

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include <QLabel>

#include <QSpinBox>

#include <QLineEdit>

#include <QIntValidator>

#include <QString>

class **Widget** : public QWidget

{

Q\_OBJECT

public:

**Widget**(QWidget \*parent = nullptr);

~***Widget***();

private:

QLineEdit \*edit;

QSpinBox \*spin1;

QSpinBox \*spin2;

QLabel \*lbl1;

QLabel \*lbl2;

QLabel \*lbl3;

QIntValidator \*val;

public slots:

void **setTitle**(const QString &title);

void **setText**(int);

void **setText2**(int);

};

#endif // WIDGET\_H

Сначала мы подключаем необходимую нам библиотеку.

**Потом в privet мы объявляем поле ввода, spinbox, label, вводим класс для проверки на ввод**. **Дальше мы объявляем слоты для того что бы выполнить задания в работе**

Widget.cpp

#include "widget.h"

Widget::**Widget**(QWidget \*parent)

:QWidget(*parent*)

{

edit = new QLineEdit(this);

edit -> move(10,40);

edit -> resize(70,20);

spin1 = new QSpinBox(this);

spin1 -> move(100,40);

spin1 -> setMaximum(100);

spin1 -> resize(70,20);

spin2 = new QSpinBox(this);

spin2 -> move(175,40);

spin2 -> setMaximum(100);

spin2 -> resize(70,20);

lbl1 = new QLabel("&hello",this);

lbl1 ->setBuddy(*edit*);

lbl1 -> move(10,10);

lbl2 = new QLabel("H&i",this);

lbl2 ->setBuddy(*spin1*);

lbl2 -> move(100,10);

lbl3 =new QLabel ("H&ello", this);

lbl3 ->setBuddy (*spin2*);

lbl3 ->move(190,10);

resize(200,200);

val = new QIntValidator(1,100,this);

edit ->setValidator(val);

connect(spin1,SIGNAL(valueChanged(int)),this,SLOT(setText(int)));

connect(edit,SIGNAL(textChanged(QString)),this,SLOT(setTitle(QString)));

connect(spin2,SIGNAL(valueChanged(int)),this,SLOT(setText2(int)));

}

void Widget :: **setText**(int f)

{

QString str = QString::number(f , 'n', 0);

edit -> setText(str);

}

void Widget :: **setTitle**(const QString &title)

{

setWindowTitle(title);

}

void Widget :: **setText2**(int a)

{

spin1 -> setMinimum(a);

}

Widget::~***Widget***()

{

}

**main.cpp**

#include "widget.h"

#include <QApplication>

int **main**(int argc, char \*argv[])

{

QApplication a(*argc*, *argv*);

Widget w;

w.show();

return a.exec();

}

**Пояснение:**

edit = new QLineEdit(this);

edit -> move(10,40);

edit -> resize(70,20);

spin1 = new QSpinBox(this);

spin1 -> move(100,40);

spin1 -> setMaximum(100);

spin1 -> resize(70,20);

spin2 = new QSpinBox(this);

spin2 -> move(175,40);

spin2 -> setMaximum(100);

spin2 -> resize(70,20);

lbl1 = new QLabel("&hello",this);

lbl1 ->setBuddy(*edit*);

lbl1 -> move(10,10);

lbl2 = new QLabel("H&i",this);

lbl2 ->setBuddy(*spin1*);

lbl2 -> move(100,10);

lbl3 =new QLabel ("H&ello", this);

lbl3 ->setBuddy (*spin2*);

lbl3 ->move(190,10);

val = new QIntValidator(1,100,this);

edit ->setValidator(val);

Тут мы создаем поле для ввода, спинбоксы, лейблы и валидатор создаем.

resize(70, 20); - размер окна

->move - мы программно устанавливаем его по координатам

-> setMaximum(100) – верхняя граница для спинбокса

this – укажчик на родительский виджет

->setBuddy – перебрасывания фокуса ввода

& - для setBuddy, ставиться для того чтобы знать на какую букву нажимать букву зажав Alt, чтобы перебрасывать фокус

connect(spin1,SIGNAL(valueChanged(int)),this,SLOT(setText(int)));

connect(edit,SIGNAL(textChanged(QString)),this,SLOT(setTitle(QString)));

connect(spin2,SIGNAL(valueChanged(int)),this,SLOT(setText2(int)));

connect(spin1,SIGNAL(valueChanged(int)),this,SLOT(setText(int)));

связываю spin1 со слотом setText

connect(edit,SIGNAL(textChanged(QString)),this,SLOT(setTitle(QString)));

связываю слот с setTitle

connect(spin2,SIGNAL(valueChanged(int)),this,SLOT(setText2(int)));

связываю spin2 со слотом setText2

void Widget :: **setText**(int f)

{

QString str = QString::number(f , 'n', 0); - перевод числа в строку

edit -> setText(str);

}

void Widget :: **setTitle**(const QString &title)

{ -перевод числа в название окна

setWindowTitle(title);

}

void Widget :: **setText2**(int a)

{ - задает крайнюю границу

spin1 -> setMinimum(a);

}