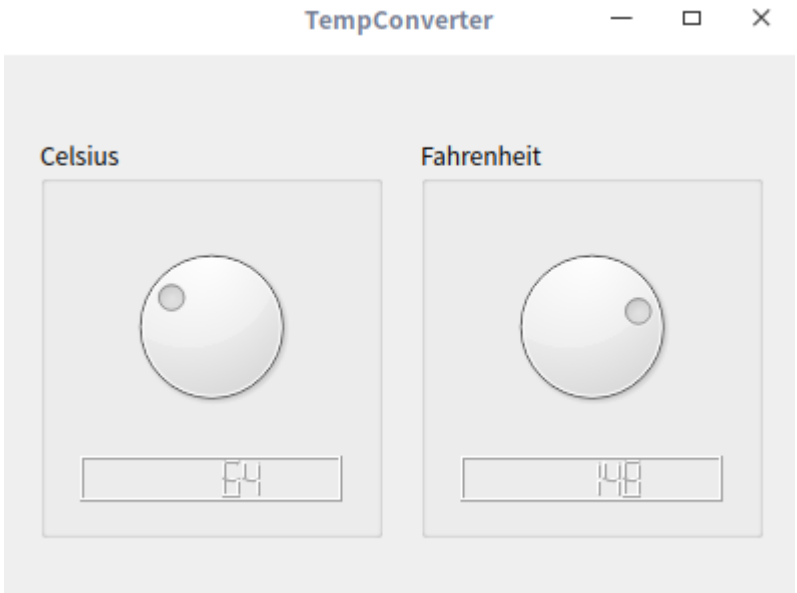


D4T2文档

1.实验任务: 使用QT完成摄氏度-华氏度换算程序

2.运行截图



3.代码分析:

代码由 `tempconverter.h` `main.cpp` `tempconverter.cpp` 组成

以下对 `tempconverter.h` 和 `tempconverter.cpp` 代码进行分析

`tempconverter.h`

```
#ifndef TEMPCONVERTER_H
#define TEMPCONVERTER_H

#include <QWidget>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QDial>
#include <QtWidgets/QGroupBox>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLCDNumber>
```

```

#include <QtWidgets/QWidget>

namespace Ui {
class TempConverter;
}

class TempConverter : public QWidget           //继承自QWidget类
{
    Q_OBJECT

public:
    explicit TempConverter(QWidget *parent = 0);
    void setUi(QWidget *TempConverter);       //该函数用于设置UI

    //声明了UI界面的主要控件
    QGroupBox *groupBox;
    QLCDNumber *cLcd;
    QDial *cDial;
    QGroupBox *groupBox_2;
    QLCDNumber *fLcd;
    QDial *fDial;
    int tempFahrenheit() const;

public slots:           //槽函数
    void setTempCelsius(int);           //用于设置摄氏温度
    void setTempFahrenheit(int);        //用于设置华氏温度
signals:               //信号
    void tempCelsiusChanged(int);
    void tempFahrenheitChanged(int);

private:
    int m_tempCelsius=0;                //初始摄氏度设为0
    int m_tempFahrenheit;
};

#endif // TEMP_CONVERTER_H

```

tempconverter.cpp

```

#include "tempconverter.h"
#include<QGroupBox>
TempConverter::TempConverter(QWidget *parent) :
    QWidget(parent)
{
    setUi(this);
    m_tempFahrenheit=m_tempCelsius*(9.0/5.0)+32;

    //初始化两个LCD和Dial控件的value
    cLcd->display(m_tempCelsius);
    fLcd->display(tempFahrenheit());
}

```

```

cDial->setValue(m_tempCelsius);
fDial->setValue(tempFahrenheit());

//讲信号与槽函数进行关联
connect(cDial, SIGNAL(valueChanged(int)), this, SLOT(setTempCelsius(int)));
connect(cDial, SIGNAL(valueChanged(int)), cLcd, SLOT(display(int)));
connect(this, SIGNAL(tempCelsiusChanged(int)), cDial, SLOT(setValue(int)));

connect(fDial, SIGNAL(valueChanged(int)), this, SLOT(setTempFahrenheit(int)));
connect(fDial, SIGNAL(valueChanged(int)), fLcd, SLOT(display(int)));
connect(this, SIGNAL(tempFahrenheitChanged(int)), fDial, SLOT(setValue(int)));
}

void TempConverter::setUi(QWidget *TempConverter)           //用于布置UI界面
{
    if (TempConverter->objectName().isEmpty())
        TempConverter->setObjectName(QStringLiteral("TempConverter"));
    TempConverter->resize(399, 274);

    groupBox = new QGroupBox(TempConverter);
    groupBox->setObjectName(QStringLiteral("groupBox"));
    groupBox->setGeometry(QRect(20, 40, 171, 201));

    //cLcd对象用于显示摄氏度的LCD
    cLcd = new QLCDNumber(groupBox);
    cLcd->setObjectName(QStringLiteral("lcdNumber"));
    cLcd->setGeometry(QRect(20, 160, 131, 23));

    //cDial用于显示摄氏度的表盘
    cDial = new QDial(groupBox);
    cDial->setObjectName(QStringLiteral("dial"));
    cDial->setGeometry(QRect(30, 50, 111, 91));
    cDial->setMaximum(200);

    groupBox_2 = new QGroupBox(TempConverter);
    groupBox_2->setObjectName(QStringLiteral("groupBox_2"));
    groupBox_2->setGeometry(QRect(210, 40, 171, 201));

    //fLcd用于显示华氏度的LCD
    fLcd = new QLCDNumber(groupBox_2);
    fLcd->setObjectName(QStringLiteral("lcdNumber_2"));
    fLcd->setGeometry(QRect(20, 160, 131, 23));

    //fDial用于显示华氏度的表盘
    fDial = new QDial(groupBox_2);
    fDial->setObjectName(QStringLiteral("dial_2"));
    fDial->setGeometry(QRect(30, 50, 111, 91));
    fDial->setMaximum(200);

    //设置文本
    TempConverter->setWindowTitle(QApplication::translate("TempConverter",
"TempConverter", Q_NULLPTR));
    groupBox->setTitle(QApplication::translate("TempConverter", "Celsius", Q_NULLPTR));

```

```

        groupBox_2->setTitle(QApplication::translate("TempConverter", "Fahrenheit",
        Q_NULLPTR));

        QMetaObject::connectSlotsByName(TempConverter);
    }

    //槽函数实现:用于重置华氏度并发射温度更改信号
    void TempConverter::setTempFahrenheit(int tempFahrenheit)
    {
        if(m_tempFahrenheit == tempFahrenheit)
            return;
        m_tempFahrenheit=tempFahrenheit;
        m_tempCelsius = (5.0/9.0)*(tempFahrenheit-32);
        setTempCelsius(m_tempCelsius);
        emit tempCelsiusChanged(m_tempCelsius);
        emit tempFahrenheitChanged(m_tempFahrenheit);
    }

    //槽函数实现:用于重置摄氏度并发射温度更改信号
    void TempConverter::setTempCelsius(int tempCelsius)
    {
        if(m_tempCelsius == tempCelsius)
            return;
        m_tempCelsius = tempCelsius;
        m_tempFahrenheit=tempCelsius*(9.0/5.0)+32;
        emit tempCelsiusChanged(m_tempCelsius);
        emit tempFahrenheitChanged(m_tempFahrenheit);
    }

    int TempConverter::tempFahrenheit() const
    {
        return (m_tempCelsius*(9.0/5.0)+32);
    }

```

分析:

本实验中主窗体继承于QWidget类,使用了QDial,QGroupBox,QLCDNumber类,

主要实现了三种信号-槽的连接

- 1.cDial对象的值更改信号 `valueChanged(int)` 与窗体的设置温度参数槽 `setTempCelsius(int)` 连接,用于实现当转盘转动时更新TempConverter类私有属性温度的值
- 2.cDial对象的值更改信号 `valueChanged(int)` 与LCD对象的更新显示槽函数 `display(int)` 连接,用于实现当转盘转动时更新LCD显示的示数
- 3.主窗体的温度参数改变信号 `tempCelsiusChanged(int)` 与cDial对象的值重置函数连接 `setValue(int)` ,用于实现当华氏度改变时对应摄氏温度改变,从而更新转盘和LCD的状态

4.实验反思

可考虑使用QT Designer辅助设计UI,所见即所得,清晰明了,易于使用.

设计程序时尽量考虑将逻辑和UI解耦,模块化有助于程序的修改和复用.