

2008-QVariant 类应用--零声教育 vico 老师

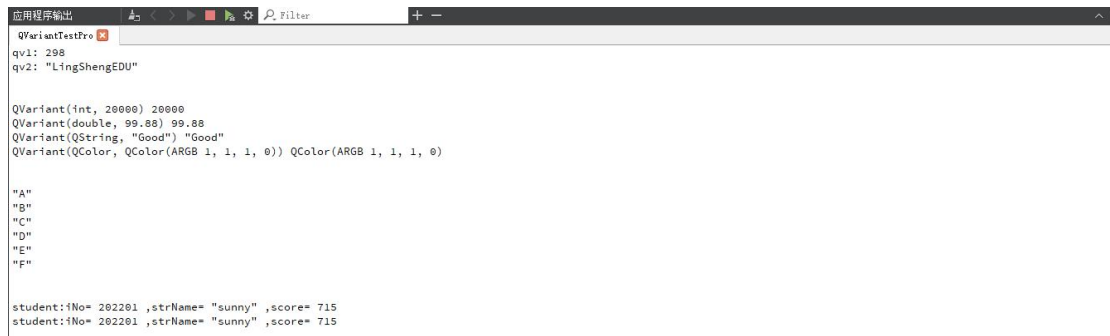
【QVariant 类应用】

QVariant 类本质为 C++联合(Union)数据类型，它可以保存很多 Qt 类型的值，包括 QBrush、QColor、QString 等等。也能够存放 Qt 的容器类型的值。

QVariant::StringList 是 Qt 定义的一个 QVariant::type 枚举类型的变量，其他常用的枚举类型变量如下表所示：

变 量	对应的类型	变 量	对应的类型
QVariant::Invalid	无效类型	QVariant::Time	QTime
QVariant::Region	QRegion	QVariant::Line	QLine
QVariant::Bitmap	QBitmap	QVariant::Palette	QPalette
QVariant::Bool	bool	QVariant::List	QList
QVariant::Brush	QBrush	QVariant::SizePolicy	QSizePolicy
QVariant::Size	QSize	QVariant::String	QString
QVariant::Char	QChar	QVariant::Map	QMap
QVariant::Color	QColor	QVariant::StringList	QStringList
QVariant::Cursor	QCursor	QVariant::Point	QPoint
QVariant::Date	QDate	QVariant::Pen	QPen
QVariant::DateTime	QDateTime	QVariant::Pixmap	QPixmap
QVariant::Double	double	QVariant::Rect	QRect
QVariant::Font	QFont	QVariant::Image	QImage
QVariant::Icon	QIcon	QVariant::UserType	用户自定义类型

【案例分析】



The screenshot shows a Qt application output window titled '应用程序输出' (Application Output). The output text is as follows:

```
QVariantTestPro
qv1: 299
qv2: "LingShengEDU"

QVariant(int, 20000) 20000
QVariant(double, 99.88) 99.88
QVariant(QString, "Good") "Good"
QVariant(QColor, QColor(ARGB 1, 1, 1, 0)) QColor(ARGB 1, 1, 1, 0)

"A"
"B"
"C"
"D"
"E"
"F"

student: iNo= 202201, strName= "sunny", score= 715
student: iNo= 202201, strName= "sunny", score= 715
```

头文件:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

// 定义学生结构体类型
struct student
{
    int iNo;
    QString strName;
    int score;
};
Q_DECLARE_METATYPE(student)

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
};
#endif // MAINWINDOW_H
```

源码文件:

```
#include "mainwindow.h"

#include <QVariant>
#include <QDebug>
#include <QColor>
```

```

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
{
    QVariant qv1(298);
    qDebug() << "qv1:" << qv1.toInt();

    QVariant qv2("LingShengEDU");
    qDebug() << "qv2:" << qv2.toString();

    QMap<QString, QVariant> qmap;
    qDebug() << endl;
    qmap["int"]=20000; // 整型
    qmap["double"]=99.88; // 浮点型
    qmap["string"]="Good"; // 字符串
    qmap["color"]=QColor(255, 255, 0); // QColor 类型

    // 输出：转换函数来处理
    qDebug() << qmap["int"] << qmap["int"].toInt();
    qDebug() << qmap["double"] << qmap["double"].toDouble();
    qDebug() << qmap["string"] << qmap["string"].toString();
    qDebug() << qmap["color"] << qmap["color"].value<QColor>();

    // 创建一个字符串列表：QStringList
    qDebug() << endl;
    QStringList qsl;
    qsl << "A" << "B" << "C" << "D" << "E" << "F";

    QVariant qvsl(qsl); // 将列表存储在一个 QVariant 变量
    if(qvsl.type()==QVariant::StringList)
    {
        QStringList qlist=qvsl.toStringList();
        for (int i=0;i<qlist.size();i++) {
            qDebug() << qlist.at(i); // 输出列表数据信息
        }
    }

    // 结构体类型和 QVariant 类配合使用
    qDebug() << endl;
    student stu;
    stu.iNo=202201;
    stu.strName="sunny";
    stu.score=715;

    // 使用静态方法保存数据

```

```

QVariant qstu=QVariant::fromValue(stu);

if(qstu.canConvert<student>()) // 判断是否可以转换操作
{
    student temp=qstu.value<student>(); // 获取数据
    student qtemp=qvariant_cast<student>(qstu); // 获取数据

    qDebug()<<"student:iNo="<<temp.iNo<<", strName="<<temp.strName<<", score="<<temp.score;

    qDebug()<<"student:iNo="<<qtemp.iNo<<", strName="<<qtemp.strName<<", score="<<qtemp.score;
}

}

MainWindow::~MainWindow()
{
}

```