## 附

南京工程学院

**实 训 报 告**

课程名称 **多媒体编程基础**

实训项目名称 实训1：控件与布局管理器应用

——用户信息界面设计

实训学生班级 数嵌 172

实训学生姓名 朱广锋

学　　　　号 202170638

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **实训目的**   1.掌握 Qt Creator 的基本使用方法  2.掌握常用Qt控件的应用方法  4.掌握布局管理器的应用，自行设计界面布局  **二、实训环境及开发工具：**  PC机、Qt5.14（或其它版本）  **三、实训要求及内容：**  1. 自行安装Qt 5.14（或其它版本）软件，并搭建好 Qt 开发环境。  2. 学习 Qt Creator 设计模式的基本使用方法。  3. 学习教材第2章关于控件的内容（P.38-52）  4. 学习教材第3章内容，参考提供的案例，完成一个显示学生信息的  对话框界面，信息内容必须含有：学生**姓名、班级、学号、性别**（其  他可以自由扩展）  5. 建议重点参考教材案例：CH304和CH305 **修改用户资料**  **6. 创建方法不必完全照搬教材，建议使用ui设计器+手动添加代码**  **四、程序设计思路（30分）**   1. 从数据结构组织分析，学生信息主要包括姓名、学号、性别、班级等，其数据类型则为字符串、布尔、整型等，可以简单的统一用字符串来保存，Qt与C++标准均提供了字符串类，为了和Qt控件更好的兼容，这里采用QString作为字符串类。 2. 在布局上，可以参考一些实际的学生信息管理系统，如奥兰等，采取左侧为学生列表，右侧显示学生信息的布局，由于学生信息较多，采取表格方式的布局。使用QSplitter来分割主窗口，GridLayoutTemplate(QGridLayout派生类)、QVBoxLayout、QHBoxLayout作为布局控件，QHBoxLayout来水平布局头像位置，QVBoxLayout对经历和备注进行垂直布局，GridLayoutTemplate以表格的形式布局学生的主要信息。 3. 学生列表采用QListWidget控件，文字提示信息使用QLabel控件，姓名、班级等单行信息采用QLineEdit控件，经历和备注的内容可能很长，采用QTextEdit控件，性别采用下拉框QComboBox控件。   **五、设计方法及代码（30分）**  GridLayoutTemplate派生自QGridLayout，增加pushWidgets函数，来简化添加一行控件的过程。   |  | | --- | | void GridLayoutTemplate::pushWidgets(std::initializer\_list<QWidget\*>widgetList)  {  int nCol = 0;  for (auto& widget : widgetList)  QGridLayout::addWidget(widget, curRow, nCol++);  curRow++;  } |   StudentInfoWidget构造函数初始化子控件，设定布局，由于子控件全部为成员对象，所以可以在构造列表内初始化。   |  | | --- | | StudentInfoWidget::StudentInfoWidget(QWidget\* parent)  : QWidget(parent)  , stuNameLabel("姓名：")  , stuNumberLabel("学号：")  …// 略  , avatarLabel("头像")  {  genderComboBox.addItem("女");  genderComboBox.addItem("男");  QPixmap icon("student.png");  avatarIconLabel.setPixmap(icon);  avatarIconLabel.resize(icon.width(), icon.height());  // 添加子控件  leftLayout.pushWidgets({ &stuNameLabel,&stuNameEdit });  …// 略  leftLayout.pushWidgets({ &nationalityLabel,&nationalityTextEdit });  leftLayout.pushWidgets({ &politicalOrientationLabel,&politicalOrientationTextEdit });  leftLayout.pushWidgets({ &postcodeLabel,&postcodeTextEdit });  leftLayout.pushWidgets({ &enrollmentTimeLabel,&enrollmentTimeTextEdit });  // 设定缩放  leftLayout.setColumnStretch(0, 1);  leftLayout.setColumnStretch(1, 3);  // 设定头像位置布局  topRightLayout.setSpacing(20);  topRightLayout.addWidget(&avatarLabel);  topRightLayout.addWidget(&avatarIconLabel);  // 设定右侧信息布局  rightLayout.setMargin(10);  rightLayout.addLayout(&topRightLayout);  rightLayout.addWidget(&experienceLabel);  rightLayout.addWidget(&experienceTextEdit);  rightLayout.addWidget(&commentsLabel);  rightLayout.addWidget(&commentsTextEdit);  // 设置主布局  auto mainLayout = new QGridLayout(this);  mainLayout->setMargin(15);  mainLayout->setSpacing(10);  mainLayout->addLayout(&leftLayout, 0, 0);  mainLayout->addLayout(&rightLayout,0,1);  mainLayout->setSizeConstraint(QLayout::SetFixedSize); |   stickStudent设定控件内容，以便在切换时更换学生信息。   |  | | --- | | // 设置学生信息  void StudentInfoWidget::stickStudent(const StudentInfo& stu)  {  stuNameEdit.setText(stu.stuName);  …// 略  commentsTextEdit.setText(stu.comments);  QPixmap icon(stu.photo.isEmpty() ? "student.png" : stu.photo);  avatarIconLabel.setPixmap(icon);  avatarIconLabel.resize(icon.width(), icon.height());  } |   students数组初始化如下。   |  | | --- | | StudentInfo students[]{  {  .stuName = "刘一"  , .stuNumber = "202200601"  …// 略  },…// 略  }; |   主布局框的设置，此处连接槽函数，由左侧学生列表控件list发出currentRowChanged信号，由StudentInfoWidget接收，并使用stickStudent设定新的学生信息。使用了lambda表达式。   |  | | --- | | QSplitter splitterMain(Qt::Horizontal);  splitterMain.setOpaqueResize(true);  QListWidget list(&splitterMain);  StudentInfoWidget StudentInfoWidget(&splitterMain);  QObject::connect(&list, &QListWidget::currentRowChanged, &StudentInfoWidget, [&](int n) {StudentInfoWidget.stickStudent(students[n]); });  list.insertItem(0, "刘一");  …// 略  list.insertItem(9, "郑十");  StudentInfoWidget.stickStudent(students[0]);  // 设置主布局框即水平分割窗的标题  splitterMain.setWindowTitle("学生信息管理");  splitterMain.setMinimumSize(splitterMain.minimumSize());  splitterMain.setMaximumSize(splitterMain.maximumSize());  splitterMain.show(); |   **六、实训结果及说明（30分）**  程序运行结果如图所示，点击左侧列表更换信息显示不同学生的信息。  **七、实训思考（10分）**   1. 由于项目是使用Visual Studio编写的，在起初运行时会发现编码问题导致界面乱码。   后在StudentInfoWidget类定义文件添加宏指令来改变编码为uft-8解决了问题。   |  | | --- | | #ifdef Q\_OS\_WIN  #pragma execution\_character\_set("utf-8")  #endif |  1. 在StudentInfoWidget进行布局后发现只显示其中一个布局。   检查发现对布局初始化时设定了父窗口为this，即StudentInfoWidget类对象指针，设置为nullptr（默认）解决。   1. 改进：   ① 信息窗口应该默认是无法修改的，可以增加一个将编辑框控件替换为Label或者设为只读的Widget，或者增加设定编辑框只读属性的按钮来控制编辑和显示状态。  ② 头像的位置不太适合，可以调整布局，放在名字附近。 |