

CAD

功能

1. 各种图形：直线、矩形、椭圆、五角星、六角形
2. 操作：移动、旋转、删除
3. 设置：设置线条的格式、设置填充的格式
4. 其他：保存文件、读取文件

有关QT

菜单栏

一般来说，只能有一个菜单栏

- 通过代码
 - 创建菜单栏：`QMenuBar *menu_bar = menuBar();`
 - 设置菜单栏：`setMenuBar(menu_bar);`
 - 创建菜单：`QMenu *file_menu = menu_bar->addMenu(tr("文件"));`
 - 创建菜单项：`QAction *newfile_action = file_menu->addAction(tr("新建"));`
 - 添加分隔栏：`file_menu->addSeparator();`
- 通过QT设计师界面
 - 直接在菜单栏上添加即可

工具栏

- 通过代码
 - 创建工具栏：`QToolBar *tool_bar = new QToolBar(this);`
 - 添加工具栏：`addToolBar(Qt::TopToolBarArea, tool_bar);`
 - 添加工具栏项目：`tool_bar->addAction(newfile_action);`
 - 添加分割线：`tool_bar->addSeparator();`
- 通过QT设计师界面
 - 直接在**Action**窗口拖拽**action**项目到工具栏上即可

状态栏

- 通过代码
 - 创建状态栏：`QStatusBar *status_bar = statusBar();`
 - 设置状态栏：`setStatusBar(status_bar);`
 - 放左侧信息：`status_bar ->addWidget(label1);`
 - 放左侧信息：`status_bar ->addPermanentWidget(label2);`

事件Event

事件是由窗口系统或者QT自身产生的，用以响应所发送的各类事情

鼠标事件

- 鼠标按下: `void mousePressEvent(QMouseEvent *event);`
- 鼠标移动: `void mouseReleaseEvent(QMouseEvent *event);`
- 鼠标释放: `void mouseMoveEvent(QMouseEvent *event);`

键盘事件

- 按键按下: `void keyPressEvent(QKeyEvent *event);`

绘制事件

`void paintEvent(QPaintEvent *event);`

QT绘图利用 QPainter 对象，例如：

```
1  void ClientWidget::paintEvent(QPaintEvent *event)
2  {
3      QPainter *painter = new QPainter(this);
4      painter->setPen(current_pen);
5      painter->setBrush(current_brush);
6      for(auto &item : list_shape) {
7          item->draw(painter);
8      }
9      delete painter;
10 }
```