

# C++ Qt 3个月实习导向学习计划

**核心目标：**掌握C++基础与Qt核心框架，完成3个递进式项目，具备Qt岗位实习能力。

**适用人群：**有C语言基础，对面向对象编程有初步认知（无Qt经验）。

**学习原则：**理论+实战7:3配比，每周任务必须落地代码，项目需完整提交到GitHub。

## 整体阶段规划

月份	核心目标	阶段定位	产出物
第1月	夯实C++基础 + Qt环境搭建与入门	筑基期	C++语法练习题、Qt基础控件Demo集
第2月	深入Qt核心机制 + 小型桌面应用开发	进阶期	本地文件管理器（Qt版）
第3月	实战Qt高级特性 + 综合项目开发	冲刺期	跨平台学生信息管理系统（带数据库）

## 第一月：C++基础强化 + Qt入门（筑基期）

### 月度大目标

- 精通C++面向对象核心语法（类、继承、多态、模板）。
- 搭建Qt开发环境（Windows/macOS），掌握Qt Creator基本操作。
- 掌握Qt核心基础控件（按钮、标签、输入框、布局），能编写简单窗口程序。

## 第1周：C++基础查漏补缺 + Qt环境搭建

时间	学习任务	实战任务	验收标准
周一	1. C++与C的差异（命名空间、引用、const） 2. 类与对象的定义与实例化	编写C++版“学生类”：包含姓名、学号属性，实现构造函数、析构函数、显示信息成员函数	代码无内存泄漏，g++编译通过
周二	1. 继承与多态（虚函数、纯虚函数、抽象类） 2. 动态内存分配（new/delete vs malloc/free）	实现“形状类”继承体系：Shape（抽象类）→ Circle、Rectangle，重写计算面积虚函数	多态调用成功，输出不同形状面积
周三	1. 模板编程（函数模板、类模板） 2. STL容器入门（vector、string）	1. 实现通用交换函数模板 <code>swap&lt;T&gt;(T&amp; a, T&amp; b)</code> 2. 用vector存储学生对象并遍历输出	模板支持int/string/自定义类，vector遍历无错误
周四	1. Qt官网下载安装（Qt 6.5 LTS） 2. Qt Creator界面熟悉（项目管理、编辑器、调试器） 3. 第一个Qt程序：Hello World窗口	1. 配置Qt开发环境，解决编译器问题（MinGW/MSVC） 2. 编写QWidget窗口，显示“Hello Qt”标签	窗口正常弹出，标签文字居中显示

时间	学习任务	实战任务	验收标准
周五	1. Qt项目文件结构（.pro、.h、.cpp、.ui） 2. 信号与槽机制入门（connect函数）	实现“按钮点击改变标签文字”功能：点击按钮，标签显示“Clicked!”	信号槽连接成功，点击按钮触发文字变化
周六	复习本周C++与Qt知识点，整理笔记	优化Hello Qt程序：添加退出按钮，点击关闭窗口	程序无闪退，功能正常
周日	休息 + 浏览Qt官方文档（Getting Started部分）	-	理解Qt的MVC设计思想

## 第2周：Qt基础控件与布局管理

时间	学习任务	实战任务	验收标准
周一	1. Qt基本控件：QLabel、QPushButton、QLineEdit 2. 控件的属性设置（文字、大小、位置）	编写登录界面：包含账号输入框、密码输入框、登录按钮	控件布局合理，密码输入框显示为圆点
周二	1. Qt布局管理器：QHBoxLayout、QVBoxLayout、QGridLayout 2. 布局嵌套使用技巧	用布局管理器重构登录界面，替代setGeometry绝对布局	窗口缩放时，控件自动调整位置和大小
周三	1. Qt信号与槽进阶（Lambda表达式写法） 2. 常用信号：clicked、textChanged	给登录界面添加功能：输入账号密码后点击登录，弹窗显示输入内容	信号槽用Lambda实现，弹窗信息准确
周四	1. Qt对话框：QMessageBox（信息、警告、错误弹窗） 2. QInputDialog输入对话框	实现“用户注册”小工具：输入用户名，弹窗确认后显示欢迎信息	支持不同类型弹窗，交互逻辑无bug
周五	1. Qt控件状态管理（enable、visible属性） 2. 事件处理入门（鼠标点击事件）	优化登录界面：未输入账号时，登录按钮置灰；输入后按钮激活	按钮状态随输入内容动态变化
周六	整合本周知识点，编写“简易计算器”(仅支持加减乘除)	计算器界面：数字按钮、运算符按钮、结果显示框	能完成基础运算，无逻辑错误
周日	上传本周所有Demo到GitHub，编写README说明	-	GitHub仓库包含5个以上Qt Demo

第3周：C++ STL进阶 + Qt容器

时间	学习任务	实战任务	验收标准
周一	1. STL容器深入：list、map、set 2. 迭代器的使用（const_iterator、reverse_iterator）	用map存储“学号-姓名”键值对，实现添加、查询、删除功能	支持根据学号查询姓名，无迭代器越界
周二	1. STL算法：sort、find、for_each 2. 函数对象与Lambda表达式结合	对vector进行排序，用for_each遍历输出	排序算法正确，遍历效率达标
周三	1. Qt容器：QList、QMap、QVector（与STL容器对比） 2. Qt容器的迭代器：QIterator	用QList存储学生成绩，实现按分数排序	Qt容器排序功能正常，与STL容器对比笔记
周四	1. Qt字符串类：QString（与std::string对比） 2. QString常用操作：拼接、分割、转换	编写字符串处理工具：将输入字符串按逗号分割，转换为大写后输出	支持中文处理，无乱码问题
周五	1. Qt日期时间类：QDate、QTime、QDateTime 2. 系统时间获取与格式化	实现“电子时钟”：显示当前系统时间，每秒刷新一次	时间显示准确，刷新无卡顿
周六	整合Qt容器与字符串，编写“通讯录管理工具”（本地内存存储）	功能：添加联系人（姓名、电话）、查询联系人、删除联系人	支持增删改查，数据存储在QMap中
周日	复习C++ STL与Qt容器差异，整理对比表格	-	明确项目中Qt容器与STL容器的选用场景

第4周：C++异常处理 + Qt窗口进阶

时间	学习任务	实战任务	验收标准
周一	1. C++异常处理：try-catch-throw 2. 自定义异常类	编写除法运算程序，捕获除数为0的异常，抛出自定义异常	异常捕获成功，错误提示友好
周二	1. Qt多窗口编程：主窗口与子窗口的切换 2. 窗口间数据传递（信号槽）	实现“主窗口-设置窗口”：点击主窗口设置按钮，弹出设置窗口；设置完成后返回主窗口显示	窗口切换无闪退，数据传递准确
周三	1. Qt菜单栏、工具栏、状态栏（QMenuBar、QToolBar、QStatusBar） 2. 资源文件（.qrc）的添加与使用	给通讯录工具添加菜单栏：文件→新建/退出；工具栏添加新建按钮	资源文件包含图标，工具栏按钮显示图标
周四	1. Qt文件操作：QFile、QTextStream 2. 文本文件的读写	实现通讯录数据的本地保存与加载：将联系人数据保存到txt文件，启动时自动加载	数据持久化成功，重启程序后数据不丢失

时间	学习任务	实战任务	验收标准
周五	1. Qt错误处理：QFile的错误判断 2. 日志输出：qDebug()的使用	给通讯录工具添加日志功能：记录数据保存/加载的时间与状态	日志输出到控制台，错误信息准确
周六	第一月月度项目整合：完善通讯录管理工具	功能清单： 1. 多窗口交互 2. 本地文件读写 3. 异常处理与日志输出	项目可独立运行，无崩溃bug
周日	第一月学习总结，整理知识盲点（如模板特化、Qt信号槽机制原理）	-	列出第二月需要重点攻克的知识

## 第二月：Qt核心机制 + 小型项目开发（进阶期）

### 月度大目标

- 1. 掌握Qt核心机制（信号槽原理、事件循环、自定义控件）。
- 2. 熟练使用Qt文件系统、网络编程基础。
- 3. 完成小型实战项目：**Qt本地文件管理器**，具备独立开发桌面应用的能力。

### 第5周：Qt信号槽深入 + 自定义控件

时间	学习任务	实战任务	验收标准
周一	1. 信号槽原理：SIGNAL/SLOT宏 vs 新语法（Qt 5+） 2. 信号槽的连接类型（AutoConnection、DirectConnection）	编写demo测试不同连接类型的执行顺序，理解同步/异步调用	输出日志清晰展示不同连接类型的差异
周二	1. 自定义信号与槽：自定义类的信号声明与槽函数实现 2. Q_OBJECT宏的作用，moc工具原理	实现“自定义学生类”：添加成绩变化信号，成绩超过90分触发奖励槽函数	自定义信号槽连接成功，moc编译无错误
周三	1. Qt自定义控件：继承QWidget重写paintEvent 2. 绘图基础：QPainter、QPen、QBrush	实现自定义圆形进度条：支持设置进度值，显示百分比	进度条随数值变化，绘制平滑无锯齿
周四	1. Qt事件过滤器：eventFilter 2. 常用事件：键盘事件、鼠标移动事件	给文件管理器添加事件过滤：鼠标悬停在文件上时，显示文件大小提示	事件过滤生效，提示信息实时显示
周五	1. Qt定时器：QTimer的使用 2. 定时任务的实现	给进度条添加动画效果：进度从0到100自动递增，每秒更新一次	动画流畅无卡顿，定时器精准触发
周六	整合自定义控件，编写“进度条工具集”：包含圆形进度条、线性进度条	两种进度条支持自定义颜色、进度值	控件可复用，能嵌入其他窗口
周日	学习Qt官方文档中“自定义控件”章节，整理最佳实践	-	明确自定义控件的开发流程

第6周：Qt文件系统 + 网络编程基础

时间	学习任务	实战任务	验收标准
周一	1. Qt文件系统深度解析：QDir、QFileInfo 2. 遍历目录下的文件与文件夹	实现“目录遍历工具”：输入路径，列出所有文件和子文件夹	支持递归遍历，显示文件大小、修改时间
周二	1. Qt二进制文件读写：QDataStream 2. 文本文件与二进制文件的区别	实现“文件加密工具”：将文本文件转换为二进制文件保存，支持解密还原	加密后的文件无法直接打开，解密后与原文件一致
周三	1. Qt网络编程基础：QTcpSocket、QTcpServer 2. TCP客户端与服务端通信流程	编写TCP回显服务器：客户端发送消息，服务器原样返回	客户端与服务端通信正常，支持多客户端连接
周四	1. Qt网络请求：QNetworkAccessManager 2. 发送HTTP GET请求获取网络数据	实现“简易天气查询工具”：调用公开天气API，显示天气信息	能解析JSON格式的返回数据，显示城市天气
周五	1. Qt JSON解析：QJsonDocument、QJsonObject 2. JSON数据的序列化与反序列化	优化天气查询工具：将查询结果保存为JSON文件，启动时加载历史记录	JSON文件读写正常，历史记录显示准确
周六	开始月度项目：Qt本地文件管理器 需求分析与架构设计	架构设计： 1. 主窗口：左侧目录树、右侧文件列表 2. 功能：文件复制、粘贴、删除、重命名	完成项目需求文档和UI草图
周日	学习Qt文件管理器相关开源项目（如Qt Creator的文件浏览器），借鉴设计思路	-	整理项目开发的技术难点（如文件拖拽）

第7周：Qt文件管理器开发（核心功能）

时间	学习任务	实战任务	验收标准
周一	1. Qt树形控件：QTreeWidget（显示目录树） 2. 列表控件：QListWidget（显示文件列表）	实现文件管理器UI：左侧目录树显示系统磁盘，右侧显示当前目录文件	UI布局合理，目录树与文件列表联动
周二	1. Qt文件拖拽功能：dragEnterEvent、dropEvent 2. MIME类型：QMimeData	实现文件拖拽：支持将文件拖拽到指定目录	拖拽功能正常，无文件丢失
周三	1. Qt右键菜单：QMenu 2. 菜单动作与信号槽连接	给文件列表添加右键菜单：包含打开、复制、删除选项	右键菜单弹出位置准确，功能正常
周四	1. Qt文件操作：复制、删除、重命名（QFile::copy、remove） 2. 进度条显示文件复制进度	实现文件复制功能：复制大文件时显示进度条	进度条实时更新，复制完成后提示

时间	学习任务	实战任务	验收标准
周五	1. Qt状态栏实时显示文件信息（大小、类型、修改时间） 2. 快捷键支持（Ctrl+C复制、Ctrl+V粘贴）	给文件管理器添加快捷键和状态栏信息显示	快捷键触发功能正常，状态栏信息准确
周六	优化文件管理器：添加文件搜索功能，支持按文件名模糊搜索	搜索功能：输入关键词，实时显示匹配文件	搜索速度快，支持中文关键词
周日	项目代码重构：提取工具类（如FileUtils），优化代码结构	-	代码注释完整，函数命名规范

## 第8周：Qt文件管理器完善 + 第二月总结

时间	学习任务	实战任务	验收标准
周一	1. Qt多线程入门：QThread的使用（避免UI卡顿） 2. 多线程与主线程的通信（信号槽）	给文件管理器添加多线程文件复制：避免复制大文件时UI卡死	UI响应流畅，进度条实时更新
周二	1. Qt程序打包：Windows下用windeployqt，macOS下用macdeployqt 2. 生成可执行文件	打包文件管理器，生成独立运行的exe/dmg文件	无Qt环境的电脑可直接运行
周三	1. Qt单元测试：QTest框架入门 2. 编写测试用例测试文件操作功能	编写单元测试：测试文件复制、删除功能的正确性	测试用例全部通过，无功能bug
周四	项目bug修复与优化：解决边界问题（如删除系统文件、路径不存在）	修复已知bug，添加错误提示	程序稳定性提升，无崩溃情况
周五	编写项目README文档：包含功能介绍、编译步骤、使用说明	README文档详细，附项目截图	文档符合开源项目规范
周六	第二月月度项目验收：上传文件管理器到GitHub，打tag v1.0	-	GitHub仓库包含完整代码、打包文件、文档
周日	第二月学习总结，梳理Qt核心知识点（信号槽、事件、多线程）	-	列出第三月综合项目需要的技术栈

## 第三月：Qt高级特性 + 综合项目开发（冲刺期）

### 月度大目标

- 掌握Qt数据库编程、多线程、界面美化核心技术。
- 完成**综合实战项目：跨平台学生信息管理系统**，具备企业级项目开发能力。
- 整理实习作品集，准备简历和面试题。

第9周：Qt数据库编程 + 多线程进阶

时间	学习任务	实战任务	验收标准
周一	1. Qt数据库模块：QSqlDatabase、QSqlQuery 2. SQLite数据库连接与操作	实现“数据库连接工具”：连接SQLite，创建学生信息表	数据库连接成功，表结构创建正确
周二	1. Qt数据库增删改查（CRUD）操作 2. 参数化查询（防止SQL注入）	编写学生信息管理的基础功能：添加、删除、修改学生信息	数据库操作无SQL注入风险，数据准确
周三	1. Qt模型视图架构：QSqlTableModel、QTableView 2. 数据与界面的分离（MVC模式）	用QTableView显示学生信息表，支持排序、筛选	表格数据与数据库实时同步
周四	1. Qt多线程进阶：继承QThread vs 移动到线程（moveToThread） 2. 多线程安全：QMutex互斥锁	实现多线程数据库查询：后台查询数据，前台显示进度	多线程查询不阻塞UI，数据安全
周五	1. Qt信号槽跨线程通信 2. 线程池：QThreadPool、QRunnable	用线程池优化批量数据导入功能：批量导入学生信息时多线程处理	导入速度提升，无数据重复或丢失
周六	综合项目需求分析：学生信息管理系统 功能规划	功能清单： 1. 学生信息CRUD 2. 成绩管理 3. 数据导入导出 4. 用户登录权限	完成详细需求文档和数据库设计
周日	学习Qt数据库相关开源项目，借鉴架构设计	-	确定项目技术栈：Qt + SQLite + MVC

第10周：学生信息管理系统开发（核心功能）

时间	学习任务	实战任务	验收标准
周一	1. Qt登录界面与权限控制 2. 数据库用户表设计（用户名、密码、权限）	实现登录功能：验证用户名密码，区分管理员/普通用户	权限控制生效，普通用户无法修改数据
周二	1. Qt表格控件高级用法：QTableView列隐藏、自定义委托 2. 单元格编辑与数据验证	实现学生信息表格：支持单元格编辑，验证学号格式	数据验证生效，非法输入无法保存
周三	1. Qt数据导入导出：Excel文件读写（使用QXlsx库） 2. CSV文件格式处理	实现数据导入导出：支持从Excel导入学生信息，导出为CSV	导入导出数据准确，格式兼容



时间	学习任务	实战任务	验收标准
周四	1. Qt图表绘制：QChart、QBarSeries（显示学生成绩统计） 2. 图表样式美化	实现成绩统计功能：用柱状图显示各班级平均分	图表显示美观，数据统计准确
周五	1. Qt打印功能：QPrinter、QPrintDialog 2. 学生信息报表打印	实现报表打印功能：打印选中的学生信息	打印预览正常，报表格式规范
周六	整合核心功能：学生信息管理 + 成绩管理 + 统计分析	系统功能完整，各模块联动正常	无功能冲突，数据流转顺畅
周日	修复项目bug，优化UI交互（如添加加载动画、错误提示）	-	程序用户体验提升

## 第11周：项目优化 + Qt界面美化 + 性能调优

时间	学习任务	实战任务	验收标准
周一	1. Qt界面美化：QSS（Qt样式表）基础语法 2. QSS与CSS的差异	用QSS美化登录界面：设置背景图片、按钮样式、输入框样式	界面美观，风格统一
周二	1. Qt样式表进阶：自定义控件样式、动态样式（hover状态） 2. QSS文件的加载与管理	编写全局QSS文件，应用到整个系统	支持换肤功能（切换不同样式表）
周三	1. Qt性能调优：减少UI重绘、优化数据库查询 2. 内存泄漏检测：使用Valgrind（Linux）/Dr. Memory（Windows）	优化数据库查询：添加索引，减少查询时间	大数据量下查询速度提升50%以上
周四	1. Qt国际化：QTranslator、多语言支持 2. 中英文切换功能实现	实现系统中英文切换：界面文字自动翻译	切换语言后无需重启程序
周五	项目文档完善：用户手册、开发手册、API文档	文档包含安装步骤、功能说明、常见问题	文档清晰易懂，适合新手使用
周六	综合项目验收：邀请同学测试，收集反馈并修复bug	系统无严重bug，功能满足需求	测试通过率100%
周日	上传学生信息管理系统到GitHub，打tag v2.0，编写详细README	-	GitHub仓库星标数≥5（可分享到技术社区）



第12周：实习求职准备 + 查漏补缺

时间	学习任务	实战任务	验收标准
周一	1. Qt面试题整理：核心机制、项目经验、技术难点 2. 常见面试题：信号槽原理、多线程实现方式	整理100道Qt面试题，编写标准答案	覆盖基础、进阶、项目三个层面
周二	1. 简历制作：突出Qt项目经验，量化成果 2. GitHub主页优化：完善README，添加项目演示视频	制作技术简历，突出两个项目的职责和成果	简历符合企业Qt开发岗位要求
周三	1. 模拟面试：针对项目中的技术难点进行复盘 2. 准备项目演示：录制学生信息管理系统操作视频	模拟面试回答流畅，能解释项目中的技术选型	演示视频清晰展示系统功能
周四	查漏补缺：针对薄弱知识点（如Qt网络编程、多线程）强化学习	编写补充Demo，巩固薄弱环节	薄弱知识点掌握程度提升
周五	求职渠道整理：实习僧、BOSS直聘、拉勾网、内推渠道	投递10家以上企业的Qt开发实习岗位	投递简历符合岗位要求，无海投
周六	复习三个月所学知识点，整理知识体系图谱	绘制Qt知识图谱，包含基础、进阶、高级三个层次	知识体系完整，逻辑清晰
周日	调整心态，准备面试，总结学习经验	-	具备自信应对Qt实习岗位面试的能力

关键保障措施

1. 代码提交：每周代码必须提交到GitHub，分支管理规范（master主分支，dev开发分支）。
2. 学习资源：
  - 官方文档：Qt 6.5 LTS 官方文档（优先阅读）。
  - 书籍：《C++ Primer》（C++基础）、《Qt 6 C++开发指南》（实战）。
  - 视频：B站《Qt 6从入门到精通》（黑马程序员）。
3. 问题解决：遇到问题优先查官方文档→Stack Overflow→Qt中文论坛，避免直接问人。
4. 时间管理：每天保证3-4小时有效学习时间，周末加倍（6-8小时），杜绝拖延。