WILEY

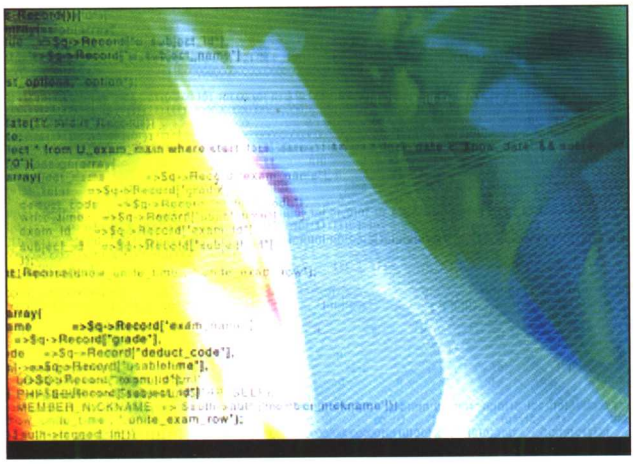
ACCREDITED symbian DEVELOPER

移动终端软件开发系列丛书

# symbian

# 认证开发工程师考试指南

## —— symbian OS基础

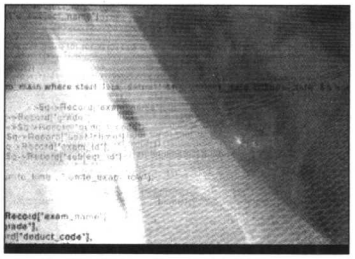


ASD考试官方指南

〔美〕Jo Stichbury 和 Mark Jacobs 著
谭利平 越敏 邹大山 潘洁 李伟 译

人民邮电出版社
POSTS & TELECOM PRESS

移动终端软件开发系列丛书



# symbian

## 认证开发工程师考试指南

### ——symbian OS基础

〔美〕Jo Stichbury 和 Mark Jacobs 著
谭利平 越敏 邹大山 潘洁 李伟 译

人民邮电出版社
北 京

**图书在版编目（CIP）数据**

Symbian 认证开发工程师考试指南：Symbian OS 基础 /（美）斯蒂克伯里（Stichbury, J.），（美）雅各布斯（Jacobs, M.）著；谭利平等译. —北京：人民邮电出版社，2008.2
（移动终端软件开发系列丛书）
ISBN 978-7-115-17133-7

Ⅰ. S… Ⅱ. ①斯…②雅…③谭… Ⅲ. C 语言—程序设计—应用—移动通信—携带电话机—工程技术人员—资格考试—自学参考资料 Ⅳ.TN929.53 TP312

中国版本图书馆 CIP 数据核字（2007）第 170511 号

## 内 容 提 要

在目前几种主流的智能手机操作系统中，Symbian OS 占据 60%以上的市场份额，是绝对的市场领先者。随着 Symbian 公司进入中国，越来越多的软件开发商、程序开发爱好者加入基于 Symbian OS 的产业链中。

Symbian 认证开发工程师（Accredited Symbian Developer, ASD）是一个面向专业智能手机软件开发者的工业标准资格认证。ASD 认证考试检验了应试者对 Symbian C++软件开发的基本方面的理解，为专业的 Symbian C++开发者提供了机会，让应试者向潜在的用人单位展示他们的知识和能力。此认证还让用人单位充满信心，来招聘那些充分了解 Symbian C++开发的人员。这类考试还可以用来找出一个团队的技术缺陷，进而减少相应的培训开支。

本书共分 16 章，循序渐进地讲解了 Symbian OS 基础知识，指出了考试要点和注意事项，并给出了参考书目和网上参考资料。通过此书，开发者可以深刻理解关于 Symbian OS 的重要概念，发现 Symbian OS 中自己不熟悉的方面，从而牢固掌握基础知识，并以此为基础探索更为专业的领域。

移动终端软件开发系列丛书

**Symbian 认证开发工程师考试指南——Symbian OS 基础**

◆ 著 [美] Jo Stichbury 和 Mark Jacobs
译 谭利平 越 敏 邹大山 潘 洁 李 伟
责任编辑 王建军

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092 1/16
印张：14.25
字数：384 千字 2008 年 2 月第 1 版
印数：1 – 4 000 册 2008 年 2 月河北第 1 次印刷
著作权合同登记号 图字：01-2007-4746

ISBN 978-7-115-17133-7/TN

定价：35.00 元

读者服务热线：(010)67113894 印装质量热线：(010)67129223
反盗版热线：(010)67171154

# 关于作者

## Jo Stichbury

Jo Stichbury 毕业于剑桥大学玛格达琳学院，并在那里获得了 Stothert Bye 奖学金。她曾获得自然科学的硕士学位和有机钼化合物化学的博士学位。从 1997 年开始她就一直在 Symbian 的“基础、连接和安全团队”为 Symbian 产业生态系统而工作。她也曾在 Advansys、索尼爱立信和诺基亚工作过。Jo 在 2005 年成为一位 Symbian 认证开发工程师，并于 2006 年获得“诺基亚论坛杰出发展者”称号。

Jo 也是 Symbian OS Explained: Effective C++ Programming for Smartphones 这本书的惟一作者，该书于 2004 年由 Symbian 出版社出版。在该书出版前不久，Jo 就离开英国去了加拿大，最后在温哥华，她与 Mark 合著了这本关于 Symbian 认证开发工程师考试的新书。

## Mark Jacobs

Mark 于 1980 年年初编写了他的第一个计算机程序。为了证明 Modula-2 是更好的程序设计语言，他在 1987 年编写了自己的第一个 C++程序，并且自那时起就喜欢上跟 C++语言的两极关系。

在 2000 年 1 月，他加入了 Symbian，当时 Symbian OS v6.0 开发已经进入最后的几个月，这完全不同于 NASA 主持开发的 Motorhead 的后期阶段系统。Mark 在 Symbian 的大多数时间是作为一个系统架构师。在 2004 年，他离开伦敦去了温哥华，并在 2005 年年末创办了 Meme Education。

他在赫特福德大学获得计算机科学的理学士学位，并且是一个 Symbian 认证开发工程师。他和 Jo 一起生活，有两只暹罗猫和一套帕摩尼咖啡器具。

# 作者致谢

首先，非常感谢 Symbian 出版社的 Coach Freddie（Sven）Gjertsen。Freddie 是个很不错的人：倾听我们诉说，帮助我们，并且默默无闻地做事。

本书能够顺利完成不得不提到 Phil Northam，是他来到温哥华的繁华大街 Piccadilly 的酒吧宴请我们，并鼓励 Jo 再写一本书。也要感谢 Drew Kenerly 和 John Wiley 的所有同仁使事情变得更加容易，还要感谢 Symbian 出版社的 McNabb。

感谢 Symbian 出版社的作者：Richard Harrison、Steve Babin、Craig Heath、 Jane Sales（及 Boris 和 Mishka）、Michael Jipping 和所有为 Symbian 出版社编写优秀著作的同仁，我们在写这本书时借鉴了这些好书。

我们还要感谢为这本书的正确性作出巨大贡献的审稿者。

Jo 要感谢支持和理解她的同事，使她能够抽出更多的时间编写这本书，尤其要感谢 Van Ly、Jon Bruce、Bill Bonney、Amonn Phillip 和 Kevin Chan。感谢以上诸位的耐心，是他们的帮助使我领会并弥补遗漏的地方，正常的编写工作才得以继续进行。

特别要感谢 Majinate 公司的 Ian Weston，他的诚实、建议和“做该做的事”的方法对我们来说是极其宝贵的。我们要向 Symsource 公司的好人们颔首表示尊重，当然也包括那些编写试题的人。谢谢，伙计们——即使知道答案，那些试题也让人很头痛。

本书是用 Pages 2 编写的，因此要感谢苹果电脑公司为我们提供的选择。

愿 James、Yvonne、Viv、Val、Nicky、Clive 和 Scott 快乐、幸福。

# 前　言

手机功能在不断地扩展，手机软件平台的性能和复杂度也随之不断增加。此外，手机用户开始更为广泛地使用移动应用程序和服务，因此对于质量的要求也随之提高。对于手机软件提供者来说，不管它是设备制造商、独立软件工作室还是程序开发爱好者，高质量的产品才是在今天以及未来取得成功的关键。

对于软件公司来说，聘请的软件工程师是否能够真正了解其程序开发所基于的操作系统，是至关重要的。仅仅通过面试就判定其是否具有如上技术能力却并非易事。Symbian 认证开发工程师（Accredited Symbian Developer，ASD）徽标则是证明个人专业能力的一种方式，招聘者可以通过它来判定应聘者是否是熟练的 Symbian C++开发人员。

如果某个人还不是 Symbian 认证开发工程师，ASD 考试将能提供其个人能力的真实信息。同时，Symbian 认证开发工程师计划还能够使面试过程更加容易。

作为开发人员如果希望提高自己的 Symbian C++专业能力，并成为 Symbian 认证开发工程师，则您可以从很多信息源获取值得学习的东西。例如，www.forum.nokia.com 和 www.Symbian.com/developer 等开发者网站，以上网站的相关讨论区以及诺基亚论坛和 Symbian 提供支持的实践培训。

书籍一直是使用最广泛的掌握程序设计技能的一种方法，哪怕是今天，它也占有一席之地。Symbian 出版社已经出版了大量的可供选择的书籍，涵盖了 Symbian 软件开发的各个方面。例如由 Jo Stichbury 编写、Symbian 出版社出版的 Symbian OS Explained，已经在世界范围内被开发者和理论学习者（作为移动软件程序设计课程基础）阅读。我敢肯定的是，她的第二本书同样会给阅读者带来不可估量的宝贵知识。可以简单地假设：如果您能够读懂本书，您一定能够获得丰富的知识，从而成为一位 Symbian 认证开发工程师。本书深入研究 ASD 考试中包含的话题，并详细地解释这些话题，从而确保读者可以通过 ASD 考试。

诺基亚论坛开发者关系副总裁——Lee Epting 女士

我与 Symbian OS 打交道已有 8 个年头了，首先是在爱立信移动通信公司，然后是 Symbian AB 公司，现在是 UIQ Technology 公司。在这段时间中，我亲历了 Symbian 社区的发展，Symbian OS 现已成为业界领先的手机操作系统，并占有最大的市场份额。随着这种发展而来的便是对具备丰富的 Symbian C++和系统 API 知识的软件开发者的需求。

本书将带领开发者进行全面的探索，深刻理解关于 Symbian OS 的重要概念，并为 Symbian 认证开发工程师考试做好准备。通过帮助开发者发现 Symbian OS 中他们不熟悉的方面，使开发者得以牢固掌握基础知识，并以此为基础探索更为专业的领域。

UIQ Technology 公司同其他手机制造商、独立软件工作室一样，定期地招募软件工程师。当然，我们期待能招聘到 Symbian OS 专家、能够编写可在手机中稳定运行 24×7h 的高质量程序并优化内存使用和功耗的优秀开发者。

但摆在所有招聘者面前的困难是，任何人都可以自称是一个 Symbian OS 专家。本书和 ASD 计划将能很好地帮助开发者成为 Symbian OS 专家，同时也为 Symbian 社区中的公司招募到合格的工程师提供帮助。在 UIQ Technology 公司，我们认可 ASD 计划并支持那些旨在帮助社区中的开发者提升专业能力并达到专业要求的活动。我们的开发者计划通过提供相关内容和支持，旨在涵盖 ASD 考试的所有范围。由于公司更期望招募到有经验的 Symbian OS 开发者，通过 ASD 认证的开发者在面对雇主时有更多优势。同时，基于 ASD 认证的评测也是帮助企业选择雇员的好方式。

本书作者 Jo 和 Mark 都曾在 Symbian 公司工作。之后 Jo 先后到索尼爱立信和诺基亚继续她的事业，并编写了 Symbian OS Explained。Mark 曾参与过 ASD 考试的制定并在加拿大经营一家在北美范围内支持 ASD 的公司。他们丰富的开发经验、渊博的 Symbian OS 知识加之长久的教学实践，使得他们成为编写本书的最佳人选。我有理由相信本书定会成为经典之作。

当你翻开本书的任何一页都能理解其中的内容时，你已具有处理绝大多数 Symbian OS 项目的能力了。

UIQ Technology 公司市场营销副总裁——Elisabet Melin 女士

# 介　绍

**Ian Weston, Majinate**

基于 Symbian 平台的软件开发是一门精细的学问，它需要解决在其他开发环境中不会遇到的一系列问题。Symbian OS 是专为手机开发的操作系统，因而使用了不同于标准桌面开发或服务器开发的开发概念。这些概念的设计用于：

- 提供有效的功耗管理。
- 确保快速启动时间。
- 允许手机在不重启的条件下持续运行。
- 节约和回收内存。
- 允许在不重启操作系统条件下进行软件安装。
- 使用网络设备的持久连接性。
- 确保网络连接失败时电话的稳定性。

## Symbian 认证开发工程师计划是什么？

与其他软件开发相比，移动软件开发中机遇和障碍的权衡有显著的区别。一个重视和理解 Symbian OS 概念的软件开发者将具有很强的专业能力。Symbian 认证开发工程师（ASD）资格使得专业 Symbian 开发者可以通过业界认可的专业技能证书，来证明他们对 Symbian 操作系统软件开发的理解和知识水平。ASD 计划的目标是保证 Symbian 产业生态系统拥有较高教育层次的开发者，并使开发者们在产业生态系统中凸显自己。

ASD 计划公布了“ASD 课程”，即一系列需要理解并用于 Symbian 操作系统软件开发的知识。该计划，尤其是该课程为持续的专业发展提供了框架，并与 Symbian OS 同步演进。课程随着操作系统新旧特性的替代而不断更新。这种更新以一种精心计划的方式进行，会全面地考虑 Symbian OS 未来在手机中的使用，也会考虑该系统长期的演化。

本课程确保对应试者进行客观的测试，测试内容是现有的资料。考试能在一定程度上区分能力水平，经过 Symbian 公司独立的确认和评审，授予达到一定水平的应试者“Symbian 认证开发工程师”的身份。

ASD 基础设施为公正、安全的测试专业能力和课程理解提供了极好的工具。通过一系列测试所表现出的可量化能力提升，培训经理便可以根据详细的熟练程度分析制定适当的培训投入并精确地评估投资回报。也就是说，这意味着老板能更好地为员工分配合适的任务和角色。全面测试还可以生成应试者在整个课程中的相对强项和弱项报告，应试者以此获得面试预备单。这在面试筛选中是很有价值的，可以更有效地利用面试时的面谈时间，从而避免了进行较长时间的技术面试。

# Symbian 认证开发工程师考试详细介绍

考试基础设施是 Majinate 公司提供的基于 Web 的考试系统，通过访问 www.majinate.com/takeexam 站点便可以在全球任何有因特网连接的地方参加考试。

该考试的问题测试应试者关于 ASD 每门课程领域的知识水平。这些问题在难度上由简单（需要很少的实践经验和有限的书本学习）到相当难（需要多年的实践经验和对本操作系统原理透彻的理解）。在任何考试中，考虑到该科目中前面问题的回答情况，使用自适应机制决定下一个问题的难度。这确保每位应试者面对的测试问题能达到他或她的理解限度。本考试没有一个预定的问题数目或持续时间；尽管如此，大多数应试者应该在 70 min 内完成考试并且在每一问题上花费大约 90 s 的时间。

考试中的每个问题有 5 个答案，这些答案中最多有 3 个是正确的。应试者应该标记出尽可能多的正确答案并避免标记出错误答案（设置来干扰警惕性不高的应试者）。只有在最快的时间内选择了所有的正确答案，才能在该问题上取得最多的分数。负值分数判给那些答题缓慢、没有选择正确答案或者选择了错误答案的应试者。跳过的问题（应试者没有选择任何答案）酌情给分。只选择了一个答案并选错了，可能导致比前一种情况获得更低的分数。因此，在决定如何回答一个问题时，应试者应该设法至少选择一个正确答案。

# 考试结果和控制

考试结束的时候，答题结果被离线分析并与 Symbian 公司确定的及格分数线进行比较。在 48 h 内，应试者会收到一封附带考试结果的邮件。通过考试的应试者将被告知他们的证书编号和证书递送时间。没有通过所要求的标准的应试者会被指出知识薄弱的环节，为他们进一步学习提出建议。该考试没有停歇期，应试者可以在任何时间内自由地再次参加考试。应试者再次考试后的简报、分析和结果不会受他之前考试的影响。

为了确定应试者必须达到何种水平才能被承认是 ASD，初始的及格分数线通过考察大量的 Symbian 公司和 Symbian 合作伙伴公司的工程师获得。及格分数线每年都会被评审，以此保证 ASD 计划维持在一个高标准上。

考试系统中有内置的安全特性来监测类似于作弊的非常规行为，系统可以根据情况而反应。为了最优化考试的可访问性，考试能以两种模式进行：那些不能或不愿意去由 Majinate 在全球范围内的任何一家合作伙伴公司进行监督的测试中心参加考试的应试者，可以参加自我监控模式的考试。除了考试环境以外，自我监控的应试者同受监控的应试者接受一样的测验并且收取一份标记出本考试未在监控下举行的证书。Majinate 不能确定持有这样一份证书的人员的确是在不借助外援的情况下单独进行考试的。对于监控下的应试者，考试的公平性受监控器保障，考试证书将显示应试者的详细情况。

潜在的雇主和面试考官能根据需求运行一个简单的在线测试，并能立即获得测试结果。尽管这种测试不等同于真正的考试，但它能确认接受测试的人是否真的达到了证书所评定的等级。

# 考试要点

## 1 C++语言基础

### 1.1 类型

- 理解 C++拥有种类繁多的数据类型，如整型、算术型（arithmetic）
- 知道 typedef 关键字是为已有数据类型定义别名，而非定义新的数据类型
- 知道枚举类型是用户定义的数值集合
- 详细说明 const 类型数据相较于数值宏定义（#define）的优点
- 理解 C++引用类型的使用和属性
- 详细说明指针与引用的区别
- 理解指针运算的语义
- 知道指针操作以及空指针（NULL）值的用途
- 区分 const 型指针和指向 const 型数据的指针

### 1.2 声明

- 了解类型和定义声明的使用与属性
- 知道 extern 关键字的使用
- 理解并说明变量的初始化及其作用域
- 理解 C++循环语句（while，for，do）的语法、行为以及用途
- 详细说明 continue 与 break 关键字在循环中的行为和作用
- 详细说明 C++条件语句（if，switch）的语法和行为

### 1.3 表达式与运算符

- 详细说明单目运算符与双目运算符表达式的语法和意义
- 理解运算符的优先级与结合性的区别
- 知道通用运算符的分类，包括逻辑运算符、前缀运算符以及后缀运算符
- 举例说明通用运算符优先级和结合性的使用规则

### 1.4 函数

- 掌握函数原型的语法
- 列举 inline 关键字的用途
- 掌握向函数传递默认参数和未指定数目参数的规则
- 知道参数的值传递、数组传递和指针传递等方法
- 理解函数的作用域，理解函数的引用返回和值返回

- 详细说明指向函数的指针语法
- 知道用指向函数的指针作为回调参数

## 1.5 动态内存分配

- 理解使用 new 与 delete 操作进行 C++自由存储分配的范围
- 知道 new 操作的语法与用途

## 1.6 工具链基础

- 掌握工具和 C++工具链（例如编译器、链接器）的功能
- 知道编译中的词法和语法分析阶段
- 能够描述 C++预处理器的作用，并指出一些常用指令
- 掌握 inline 函数在 C++中的作用
- 了解怎样使用 extern 关键字

# 2 类与对象

## 2.1 作用域与 C++面向对象程序设计支持

- 理解代码块和命名空间的作用域和生命期属性
- 掌握 C++支持的数据抽象
- 详细说明 C++对象在面向对象程序设计中的特征
- 了解类声明的语法
- 说明对象和类的区别
- 区分基本数据结构（struct 关键字）与类

## 2.2 构造函数与析构函数

- 掌握类及其成员变量的构造与析构顺序
- 知道构造函数的隐式调用（重载与模式匹配），知道关键字 explicit 在构造函数声明中的作用
- 知道编译器为用户自定义类自动生成哪些东西
- 掌握拷贝构造函数（包括参数传递）的使用和用途
- 理解赋值和初始化的区别
- 说明一个类需要哪些成员函数来安全地拥有一个指针数据

## 2.3 类成员

- 描述对类成员进行 private、protected 以及 public 型的访问控制
- 声明指向类成员的指针并详细说明其语法
- 辨别嵌套类及其作用域和生命空间
- 详细说明嵌套类的访问规则和生命期
- 掌握 friend（友元）函数的语法和作用域
- 掌握成员函数和成员数据的寻址方式

- 掌握作用域运算符的用途
- 掌握 this 指针的作用
- 详细说明 static 型类成员的性质

# 3 类的设计与继承

## 3.1 类之间的关系

- 理解继承的优点与用途
- 详细说明复合、聚合以及继承之间的区别
- 列举说明面向对象继承关系

## 3.2 继承

- 能够定义 public 继承
- 掌握使用作用域运算符访问派生类层次结构
- 给定一个类派生层次结构，详细说明其访问规则（包括其友元类）
- 描述可访问域规则，描述 public 继承、protected 继承以及 private 继承的用途
- 详细说明类派生层次结构中，各个构造函数与析构函数的调用顺序

## 3.3 动态多态——虚函数

- 详细说明 C++的面向对象重用机制
- 能够说明 C++支持的多态性
- 掌握覆盖与重载的区别与用途
- 掌握使用覆盖方式来改变派生类的行为
- 掌握调用重载函数的规则和匹配标准
- 辨别运算符重载的典型用法与行为
- 描述虚函数表的作用，说明其带来的约束与开销
- 详细说明虚函数的使用方法，使用时的取舍
- 掌握怎样在 C++中实现虚基类
- 说明接口继承与实现继承的区别
- 知道并掌握与多继承相关的问题
- 说明在用户自定义类中使用 static_cast 运算符的要求

## 3.4 静态多态与模板

- 详细说明一个简单函数模板的语法
- 能够列举函数模板的优点（例如同宏定义比较）
- 掌握类模板的语法和继承规则
- 理解模板类型/类声明的语法与意义
- 知道模板原型声明和模板声明的模式匹配特性及其使用方法
- 掌握 Symbian OS 瘦模板同主流 C++模板在实现方法和用途上的区别

# 4 Symbian OS 中类型和声明

## 4.1 基本类型

- 知道 Symbian OS 基本类型怎样同固有的 C++类型相关联
- 理解 Symbian OS 基本类型的使用应始终优先于固有的 C++类型（bool，int，float 等），因为它们是独立于编译器的

## 4.2 T 类

- 知道 T 类的用途，知道它能（不能）拥有哪些成员数据，知道它不能有析构函数
- 知道 T 类可能拥有哪些类型的函数
- 理解 T 类可以在堆或栈中创建
- 掌握 T 类可以替代传统的 C/C++结构体（struct）
- 知道 T 前缀还可以用于定义枚举类型

## 4.3 C 类

- 知道 C 类总是派生自 CBase
- 知道 C 类的用途及其他可能拥有何种类型的数据
- 理解 C 类必须在堆上实例化
- 知道 C 类分两步构造，并且其数据成员在堆上分配空间时初始化为零
- 理解 C 类的析构是通过定义在 CBase 中的虚析构函数完成

## 4.4 R 类

- 知道 R 类的用途是拥有一个资源
- 掌握 R 类可以在堆或堆栈上实例化
- 理解 R 类的构造与初始化是分开进行的
- 理解 R 类的清除与析构是分开进行的，理解在析构前忘记调用 Close()或 Reset()函数的后果

## 4.5 M 类

- 知道 M 类的用途是定义接口
- 掌握使用 M 类进行多继承，掌握继承自 C 类和 M 类的派生顺序
- 知道 M 类不能包含成员数据，不能有构造函数
- 知道 M 类可能包含的函数类型以及能够定义函数实现的适当情形
- 理解 M 类不能实例化

## 4.6 静态类

- 知道静态类不能有字母前缀
- 静态类不能被实例化，因为它只含有静态函数

## 4.7 Symbian OS 类设计注意事项

- 知道创建一个新类的注意事项，以及怎样根据这些事项来选择 Symbian OS 类

### 4.8 Symbian OS 命名规范的重要性

- 理解通过使用类前缀，可以使希望使用该类的人清楚地知道怎样实例化、使用以及销毁该类才是安全的
- 知道命名习惯可以强迫类设计者考虑第 4.7 节中的注意事项，在决定了类的基本行为之后，能够专注于类的作用，了解能够安全退出的构造函数、析构函数以及所有权问题已被解决

## 5 异常退出与清除栈

### 5.1 异常退出：Symbian OS 的轻量级异常

- 了解 Symbian OS v9 以前版本不支持标准 C++异常（try/catch/throw），但使用了轻量级的 TRAP 和异常退出作为替代
- 知道异常退出是 Symbian 错误处理的基本部分，用于整个系统
- 理解异常退出与 C 语言中声明的 setjmp/longjmp 之间的相似处
- 识记可能造成异常退出的典型系统函数，包括 User::LeaveXXX()函数和 new(ELeave)
- 能够列举造成异常退出的典型情况（例如，没有足够的内存用于堆分配）
- 理解 new(Leave)是用来保证在没有发生异常退出情况下，指针返回的值总是有效的

### 5.2 使用异常退出函数

- 知道对于可能发生异常退出的函数的名字是以“L”作为后缀的（例如，InitializeL()）
- 能够认出哪些函数不是异常退出安全的，哪些是异常退出安全的
- 理解异常退出是用于错误处理的，很少有函数既返回一个错误代码又能异常退出
- 理解在构造函数和析构函数中不应发生异常退出的原因

### 5.3 异常退出与严重错误

- 掌握异常退出与严重错误的区别
- 识记由断言失败产生的严重错误，断言在程序开发中用于标记程序的错误
- 识记异常退出不应该用来改变正常代码的执行逻辑

### 5.4 TRAP

- 识记 TRAP 捕获特性
- 清楚为了效率应该尽可能少地使用 TRAP

### 5.5 清除栈

- 了解如何使用清除栈使程序异常退出安全，即在异常退出时内存不泄漏
- 知道，即使 CleanupStack::PushL( )发生异常退出也不会泄漏内存
- 了解从清除栈上移除条目的次序，并且知道如何使用 CleanupStack::PopAndDestroy( ) 和 CleanupStack::Pop( )函数
- 识记清除栈的正确使用方法与不正确的使用方法
- 理解将一个并非从 CBase 基类派生的 C 类放入清除栈产生的后果

- 知道对于 C、R、M 及 T 类的对象如何使用 CleanupStack::PushL( )和 CleanupXXXPushL( ) 函数，并且清楚对于 C++数组如何使用 CleanupArrayDeletePushL( )
- 理解 Symbian OS 函数中以“C”和“D”为后缀的函数的含义

### 5.6 内存泄漏检测

- 识记使用__UHEAP_MARK 和 __UHEAP_MARKEND 两个宏来检测内存泄漏的方法

## 6 两阶段构造与对象析构

### 6.1 两阶段构造

- 知道为什么代码不应该在构造函数内部异常退出
- 识记两阶段构造函数，避免创建无确切状态的对象
- 理解构造函数和第二段 ConstructL( )方法在类中给定的私有或保护的成员说明列表，使用两阶段构造函数，防止无意中使用这些成员
- 知道如何实现两阶段构造，并且清楚如何使用初始化两阶段方法构造一个基类对象
- 了解使用了两阶段构造的典型的 Symbian OS 类（C 类）

### 6.2 对象析构

- 知道在析构函数中删除一个指针后将它置为 NULL 是既无效率又无必要的
- 理解析构函数在对指针解引用前必须检查该指针是不是 NULL，如果只是调用 delete 该指针就不需要检查

## 7 描述符

### 7.1 Symbian OS 描述符特性

- 知道 Symbian OS 描述符可以容纳文本和二进制数据
- 知道描述符可能是窄字符（8 位）、宽字符（16 位）或宽度无关字符（实际上是 16 位字符，因为 Symbian OS 支持 Unicode 字符集）
- 理解描述符不会动态的扩展所引用的数据空间，因此调用描述符的某个方法时如果数据空间不足以存储待保存的数据就会产生严重错误

### 7.2 Symbian OS 描述符类

- 了解 TDesC、TDes、TBufC、TBuf、TPtrC、TPtr、RBuf 及 HBufC 描述符类的特性
- 理解描述符基类 TDesC 和 TDes 实现了所有通用描述符操作代码，而派生描述符只是添加了它们特有的构造和赋值代码
- 识别 TDesC 和 TDes 类中修改方法的正确用法与错误用法
- 了解并没有 HBuf 类，不过 RBuf 可以作为一个可修改的动态分配的描述符使用

## 7.3 描述符类的继承层次

- 了解描述符类的继承层次
- 理解描述符类继承模型的内存效率和影响

## 7.4 描述符 API 的使用

- 理解描述符基类 TDesC 和 TDes 不能实例化
- 理解描述符方法 Size()、Length()和 MaxLength()之间的差异
- 理解描述符方法 Copy()和 Set()之间的差异，并且掌握如何使用它们正确地赋值

## 7.5 作为函数参数的描述符

- 理解将描述符作为参数时应当使用引用，无论该描述符是常量还是可以在函数中被修改

## 7.6 动态描述符类的正确使用

- 识别实例化一个 HBufC 堆缓冲区描述符对象的正确技巧和方法
- 识记并阐述如何使用新描述符类 RBuf

## 7.7 描述符的低效使用

- 知道 TFileName 对象不能滥用，因为每个 TFileName 对象都会消耗大量的栈空间
- 掌握何时直接对 HBufC 对象解引用，何时调用 Des() 获取可修改描述符（TDes&）

## 7.8 字面描述符

- 知道如何操作字面描述符，并且清楚使用_L 声明字面描述符的方法已被废弃
- 识别使用_L 与使用_LIT 声明字面描述符的不同，并且知道使用_L 声明字面描述符的缺点

## 7.9 描述符转换

- 知道如何使用描述符方法 Copy()或 CnvUtfConverter 类在 8 位描述符和 16 位描述符之间作转换
- 识记如何将文件中数据读取到 8 位描述符，然后不使用填充再将数据“转换”成 16 位。反之，又如何操作
- 知道如何使用 TLex 类将一个描述符转换成一个数，以及如何使用 TDes::Num() 将一个数转换成一个描述符

# 8 动态数组

## 8.1 Symbian OS 中的动态数组

- 举例说明对 Symbian OS 动态数组（CArrayX 和 RArray 类）基础的理解
- 掌握 Symbian OS 动态数组中关于内存分配（平坦的或分段的）、对象存储（在数组内或在别处）、对象长度（固定的或可变的）以及对象所有权的不同类型

- 识记使用分段缓冲区数组类而不使用平坦数组类的适当情况

## 8.2 RArray，RPointerArray 或 CArrayX

- 了解使用 RarrayX 代替 CArrayX 的原因，并且清楚最好选择使用 CArrayX 类的例外情况

## 8.3 数组粒度

- 了解数组的粒度和容量的含义
- 知道如何选择数组的粒度以适应数组的预期使用

## 8.4 数组的排序与查找

- 举例说明如何在动态数组中进行排序和查找
- 识记 RArray、RPointerArray 和 CArrayX 类都可支持排序，尽管使用 CArrayX 类排序效率不高

## 8.5 TFixedArray

- 识记当不需要动态数组时，应该优先使用 TFixedArray 代替 C++数组，因为该类支持数组越界检查（仅在调试代码中检查或在调试代码和发布代码中都检查）

# 9 活动对象

## 9.1 Symbian OS 中的事件驱动多任务

- 举例说明对同步请求和异步请求之间区别的理解，并且能够区分两者的典型例子
- 识记活动对象的典型用法，即在不阻塞线程的条件下允许请求异步任务
- 理解使用多线程的多任务与使用多个活动对象的多任务之间的区别，并且知道在 Symbian OS 代码中使用后者的原因

## 9.2 CActive 类

- 理解活动对象的优先级的重要性
- 识记活动对象时间处理方法（RunL( )）是非抢占式的
- 了解活动对象的继承特性，并且知道哪些函数需要实现和覆盖
- 知道如何构造、使用和销毁一个活动对象

## 9.3 活动调度器

- 掌握活动调度器的使用和特性
- 知道至少有一个活动对象有一个未完成的请求时，才调用 CActiveScheduler::Start( )
- 识记线程处理事件失败的典型原因可能是活动调度器没有启动或提前被停止
- 理解 CActiveScheduler 类可以被派生，以及构造派生一个活动调度器类的原因

## 9.4 取消未完成的请求

- 理解当异步请求正常完成，和由于调用 Cancel( )而取消时，活动对象在代码中执行的路径是不同的