# 实验一 软件标准和软件工具

## 一．实验目的

1. 熟悉软件领域相关的技术标准、知识产权、产业政策和法律法规。

2. 掌握软件工程过程中使用常用的建模工具。

## 二．实验内容

1. 简述软件主要特征。

软件的主要特征包括以下几个方面：

1.可编程性： 软件是可编程的，它可以根据需要进行修改、定制和扩展。程序员可以编写、编辑和维护软件代码，以适应不同的需求和变化的环境。

2. 灵活性：软件具有高度的灵活性，可以适应不同的任务和情境。通过修改软件代码，可以实现各种功能，从简单的文本处理到复杂的图形处理和数据分析。

3. 可重用性：软件中的代码和组件可以被多次重复使用，从而减少开发时间和成本。这有助于创建通用的软件库和框架，可以在多个项目中共享和应用。

4. 可维护性：软件可以容易地进行维护和更新，以适应新的需求、修复错误或改进性能。这有助于软件的长期可持续性。

5.可移植性：软件可以在不同的操作系统和硬件平台上运行，只需适当的配置和调整。这增加了软件的适用性和可扩展性。

6. 安全性：软件需要具备一定的安全性特征，以保护数据和系统免受潜在的威胁和攻击。

1. 概述传统软件工程和面向对象软件工程的区别。

传统软件工程：

1. 基于过程：传统软件工程强调开发过程，通常采用线性和顺序的方法，例如瀑布模型，开发活动被严格分阶段执行。

2. 模块化：传统软件工程通常使用模块化的方法，但模块之间的关系可能不够清晰。通常，功能被划分为不同的模块或组件。

3. 重点在数据和功能上：传统方法通常侧重于数据和功能，而不够强调对象和它们之间的关系。

4. 可维护性：传统软件工程中的代码通常比较难维护，因为模块之间的耦合可能较高。

5. 弱重用性：传统方法中的代码重用性相对较低，因为模块之间的独立性不够高。

面向对象软件工程：

1. 基于对象：面向对象软件工程强调对象，开发过程围绕对象的创建、定义和交互展开。它更加面向对象和面向问题领域。

2. 封装和继承：面向对象方法使用封装和继承等概念，以提高代码的可重用性和可维护性。

3. 关注问题领域：面向对象方法更注重问题领域的建模，使用类和对象来表示问题领域的实体和关系。

4. 模块化和松耦合： 面向对象方法鼓励模块化和松耦合，通过定义清晰的接口和封装，减少模块之间的依赖关系。

5. 高重用性：面向对象方法更有利于代码的重用，因为类和对象可以在不同项目中重复使用，提高了开发效率。

1. 以表格形式比较软件过程模型的基本思想，优势和选择条件。

|  |  |  |  |
| --- | --- | --- | --- |
| **软件过程模型** | **基本思想** | **优势** | **选择条件** |
| 瀑布模型 | 阶段化，线性顺序的开发过程，每个阶段有明确定义的输入和输出。 | 1.易于理解和管理。 2.适用于小型项目。 3.有明确的阶段和文档。 | 需求稳定。  项目风险较低。 |
| 快速原型模型 | 迭代开发，通过快速构建原型来理解和定义需求。 | 1.更好地理解和捕获需求。  2.需求变化频繁。  3.提高用户满意度。 | 需求不明确或频繁变化。快速开发需求验证。 |
| 敏捷模型 | 迭代和增量开发，注重团队协作和快速交付可用软件。 | 1.适应需求变化。  2.提高用户满意度。  3.高度灵活。 | 项目需求不断变化。有独立的、协作的团队。 |
| 喷泉模型 | 强调风险管理，包括风险分析、风险规划和风险缓解。 | 1.有效的风险管理。  2.适用于大型和复杂项目。 | 项目存在潜在的高风险。 |
| 螺旋模型 | 迭代开发，包括风险评估、需求分析、设计、实施和测试。 | 1.高度适应性。2.适合大型、高风险项目。 | 复杂项目。需要长期支持和维护。 |
| 增量模型 | 逐步构建系统，逐渐添加新功能。提前交付可用功能。容易管理和测试各个增量。 | 1.需求不完全明确。  2.提高用户满意度，快速交付可用软件。  3.适用于中大型项目。 | 项目可划分为可独立开发的部分。  项目要求快速迭代和交付。  需要验证系统的一部分。 |

1. 搜索和阅读相关软件工程国家标准，回答以下问题。
2. 请说明常用的软件工程国家标准有哪些？

1. ISO 9001：2015：这是国际标准化组织（ISO）发布的标准，用于质量管理体系，包括软件开发领域。它强调质量管理、流程改进和持续改进。

2. ISO/IEC 12207：这是ISO和国际电工委员会（IEC）联合发布的标准，定义了软件生命周期过程的框架，包括开发、维护和管理等方面。

3. ISO/IEC 15288：这是另一个由ISO和IEC联合发布的标准，涵盖了系统工程和软件工程的基本过程和方法。

4. IEEE 12207：由美国电气和电子工程师协会（IEEE）发布，类似于ISO/IEC 12207，定义了软件生命周期过程。

5. IEEE 830：该标准定义了软件需求规格说明的格式和内容，有助于清晰、一致地记录和描述软件需求。

1. 有哪些软件工程文档撰写的国家标准，请举例。

1. IEEE 830-1998：这是由美国电气和电子工程师协会（IEEE）发布的标准，定义了软件需求规格说明文档的格式和内容。它包括了规范需求的结构、标识、描述和验证方法。

2. IEEE 1016-1998：这个标准也由IEEE发布，定义了软件设计文档的内容和结构，包括模块、接口、数据结构和算法的描述。

3. ISO/IEC 25062：这个国际标准定义了软件用户文档的要求和测试方法，包括用户手册、在线帮助和培训材料等。

4. ISO/IEC 9126：该标准定义了软件质量特性和度量，其中包括质量文档的撰写和评估。

5. ISO/IEC 20000-4：这个国际标准定义了IT服务管理文档的要求，其中包括服务级别协议（SLA）和服务操作手册等。

1. 软件可行性分析一般包含哪些内容？

软件可行性分析包含以下主要内容：

1. 经济可行性：

成本估算：估计软件开发、维护和运营的成本，包括硬件、软件、人力资源和其他相关费用。

收益估算：评估软件项目可能带来的经济收益，如增加的销售、成本节约或效率提高。

成本-收益分析：比较项目成本和潜在收益，以确定项目的经济可行性。

2. 技术可行性：

技术要求：评估所需的技术、硬件和软件资源是否可用或可获得。

技术能力：考察开发团队是否具备实施项目所需的技术和知识。

技术风险：识别潜在的技术障碍或挑战，如缺乏必要的技术或工具。

3. 操作可行性：

系统集成：考虑新软件系统与现有系统的集成和交互。

维护和支持：评估软件的日常维护、支持和更新的可行性。

培训需求：确定是否需要培训用户和维护人员，以确保系统的顺利运行。

4. 法律和合规性可行性：

法律要求：考察项目是否符合法律法规，如数据隐私法律、知识产权法和行业标准。

合规性要求：确定项目是否符合组织内部政策和合规性要求。

5. 日程可行性：

项目时间表：制定项目时间表，包括开发、测试、上线和支持的时间表。

项目风险：评估项目的时间敏感性，以确定是否有足够的时间来完成项目。

6. 市场可行性：

市场需求：分析目标市场，确定是否存在足够的需求来支持软件产品。

竞争分析：研究竞争对手和市场竞争状况，评估进入市场的机会。

7. 环境可行性：

生态影响：考察项目对环境的潜在影响，如资源消耗、废弃物处理等。

社会和文化因素：评估项目对社会和文化方面的影响，包括社会接受度和文化因素。

1. 需求规格说明书一般包含哪些内容？

需求规格说明书（Requirements Specification Document）是软件工程中的关键文档，它描述了软件系统的需求和规格，为开发团队提供了详细的指南，以确保软件系统满足用户和系统需求。一般来说，需求规格说明书通常包含以下内容：

1. 引言：

项目的背景和目标。

文档的目的和范围。

2. 总体描述：

产品的功能和目标。

用户需求和期望。

系统的整体结构和组件。

非功能需求，如性能、安全性和可维护性。

3. 特定需求：

功能需求：详细描述系统的功能，包括用例、功能性操作和数据处理

接口需求：定义系统与其他系统、硬件或外部服务的接口。

性能需求：规定系统的性能指标，如响应时间、吞吐量和容量。

安全需求：说明系统的安全特性和要求，如身份验证和授权。

可用性需求：定义系统的可用性要求，如系统的可靠性和可维护性。

数据需求：描述数据的结构、存储和处理要求。

约束条件：包括技术和法律约束，如标准、法规和知识产权。

4. 验收标准：

描述如何验证系统是否满足需求。

包括测试用例、验收测试计划和验收标准。

5. 附录和附加信息：任何其他相关信息，如图表、图形、表格、参考文献或附加材料。

1. 概要设计和详细设计说明书的区别和联系。

概要设计：

1.深度：概要设计是设计的高层次阶段，侧重于系统的整体结构和模块之间的关系。它通常不包括具体的代码或细节。

2.关注点：概要设计主要关注系统的整体架构、模块划分、接口设计和基本算法的选择。它定义了系统的大致结构和模块之间的通信方式。

3.抽象级别：概要设计通常是抽象的，它描述了系统的逻辑视图，但不涉及实际的代码实现。

4.目标：目标是确定系统的整体设计，以确保满足功能需求并为详细设计提供基础。

详细设计：

1.深度： 详细设计是设计的低层次阶段，它更详细地描述系统的各个模块和组件，包括数据结构、算法、接口、类、函数等具体细节。

2.关注点：详细设计侧重于每个模块的内部实现，包括数据结构的定义、算法的精细规划、函数和方法的定义以及数据流和控制流的详细描述。

3.具体性：详细设计是具体的，它通常包括伪代码、类图、时序图、数据结构定义和其他实际的设计元素。

4.目标：目标是为开发人员提供足够的详细信息，以便实际编写和测试代码。

联系和关系：

1. 连续性：概要设计和详细设计是相互关联的。概要设计提供了详细设计的基础，定义了系统的整体结构，详细设计则在此基础上进行，为每个模块提供具体设计和实现。

2. 迭代性：在实际软件开发中，设计是一个迭代的过程。概要设计可能需要多次修改和改进，以满足需求，然后详细设计在不断优化和扩展。

3. 沟通：这两个阶段的设计文档用于沟通和协作。概要设计用于与利益相关者讨论系统的整体架构，详细设计则用于向开发人员传达具体实施细节。

综上所述，概要设计和详细设计是软件设计过程中的两个阶段，各自关注不同层次和细节，但它们紧密联系，一起构成了软件系统的全面设计。

1. 搜索并阅读相关软件法律法规，回答一下问题。
2. 我国关于软件知识产权的相关法律法规有哪些？

在中国，与软件知识产权相关的主要法律法规包括：

1. 中华人民共和国著作权法：该法规定了著作权的保护范围，包括计算机软件。著作权法详细规定了软件的著作权保护，包括著作权的产生、内容、继承、许可和终止等。

2. 计算机软件保护条例：该法规进一步详细说明了计算机软件的保护，包括软件的注册、许可、侵权行为的处理和行政处罚等。

3. 中华人民共和国专利法：虽然不是传统的软件著作权，但专利法规定了计算机软件的专利保护。软件可以根据专利法要求进行专利申请，以获得技术保护。

4. 中华人民共和国商标法：商标法规定了与软件相关的商标的保护，尤其是在涉及软件产品的商标注册和保护方面。

5. 中华人民共和国反不正当竞争法：该法规定了禁止不正当竞争行为，包括与软件相关的不正当竞争行为。

6. 网络安全法：网络安全法规定了涉及计算机网络和信息系统安全的法规，包括对软件安全的保护和管理。

7. 电子商务法：该法规定了涉及电子商务和在线软件销售的法规。

1. 试举例5个与软件相关的法律法规。

1. 中华人民共和国著作权法：著作权法规定了著作权的范围，包括计算机软件作品的著作权保护，以及软件开发者的权利和义务。

2. 计算机软件保护条例：这一法规详细说明了计算机软件的保护，包括软件的注册、许可、侵权行为的处理和行政处罚等。

3. 中华人民共和国专利法：尽管不是传统的软件著作权，专利法规定了计算机软件的专利保护。软件可以根据专利法要求进行专利申请，以获得技术保护。

4. 中华人民共和国商标法：商标法规定了与软件相关的商标的保护，尤其是在涉及软件产品的商标注册和保护方面。

5. 网络安全法：这一法规规定了涉及计算机网络和信息系统安全的法规，包括对软件安全的保护和管理。

1. 请详细阅读《中华人民共和国著作权法》，并结合其他文献学习，回答以下问题。

【案例分析】某一个开发者接受到甲方的委托开发一款软件，经过需求分析、设计、实现等过程后软件被成功研制出来，甲方也支付了对应的软件开发费用，并在合同中承诺未来收入的10%将作为维护费用支付给乙方。接下来甲方决定将软件推广给同行来实现互利互惠的策略。乙方不同意，那么软件的归属权究竟属于谁？

答：根据《中华人民共和国著作权法》的相关规定，软件的知识产权通常属于软件开发者，除非在合同中有明确的约定。

在上述案例中，乙方是软件的开发者，根据常见的软件开发合同惯例，软件开发者通常在开发过程中保留了软件的知识产权。然而，在合同中有明确的条款规定，甲方支付了软件开发费用，并且在合同中承诺将未来收入的10%作为维护费用支付给乙方。这意味着乙方可能会在合同中放弃了一部分知识产权，或者对软件的使用和维护权利作出了一些限制。

在这种情况下，软件的归属权可能受合同的具体约定影响。如果合同中没有明确规定软件知识产权的归属和使用权，那么根据著作权法，软件开发者通常会保留软件的知识产权。然而，由于在这个案例中有涉及费用支付和未来维护费用的承诺，所以合同的具体内容将起决定性作用。

1. 参考常用的建模工具使用指南，教材中第4章4.3.1节所有基本用例图，选择其中一个需求分析建模工具（例如visio，procession等）绘制。
2. 观看网络教学平台上关于Docker、GitHub、JUnitTest等软件工具短视频，要逐步学会使用，并且在软件工程课程设计中使用上述工具进行软件项目的测试、部署及项目版本管理。
3. 请选择一个软件工具最新版本，制作图像、声音清晰的6-8分钟该工具使用视频。（选做加分题，平时课堂成绩可以加分，在期末前可直接通过学习通邮箱提交）

## 三．实验报告

1. 要求提交 word 版本的实验报告内容（必须提交第1-6题，第8题选做），参照实验报告模板。
2. 总结本次实验体会。

**实验体会：**

在本次实验中，我得出了以下一些体会和总结：

1. 软件工程：了解了软件工程的基本概念和过程，包括传统和面向对象软件工程的区别，软件过程模型的基本思想和优势，以及增量模型等内容。

2. 软件工程文档：了解了软件工程中的不同文档类型，如需求规格说明书、概要设计和详细设计说明书等，以及它们的内容和作用。

3. 软件知识产权：学习了软件知识产权的基本概念，包括著作权、专利和商标等，以及与软件相关的法律法规，如《中华人民共和国著作权法》和《计算机软件保护条例》。

4. 软件知识产权案例分析：通过案例分析，了解了在软件开发合同中，合同的具体约定对软件知识产权归属和使用权有重要影响，需要明确约定以防止潜在争议。

5.法律法规的重要性：了解了遵守软件相关的法律法规对于软件开发者和企业的合法性和合规性非常重要，法律专业人士的咨询和法律文件的详细审查都至关重要。