

2025 高中信息技术课标

一、信息技术基础

1. 信息技术概述

信息技术概述

(1) 信息技术，顾名思义，是指与信息处理、存储、传输和利用相关的各种技术。随着社会的发展和科技的进步，信息技术已经成为现代社会不可或缺的一部分。从简单的文字处理到复杂的网络通信，从传统的计算器到现代的智能设备，信息技术的发展极大地丰富了人们的生活，提高了工作效率，推动了社会经济的快速发展。

(2) 信息技术的发展可以分为几个阶段。在早期的计算机时代，信息技术主要集中在计算机硬件和软件的开发上，如计算机的 CPU、内存、操作系统等。随着互联网的普及，信息技术进入了网络时代，人们开始通过网络进行信息交流、资源共享和远程协作。近年来，随着人工智能、大数据、云计算等新兴技术的兴起，信息技术进入了智能化时代，人工智能助手、智能穿戴设备、自动驾驶汽车等应用层出不穷，为人们的生活带来了前所未有的便利。

(3)

信息技术的发展不仅带来了便利，也带来了一系列挑战。如何确保信息安全、保护个人隐私、应对网络攻击等问题日益突出。同时，信息技术的发展也对社会产生了深远的影响，包括就业结构的变化、教育方式的革新、文化传承的演变等。因此，了解信息技术的发展历程、现状和趋势，对于每个人来说都具有重要的意义。在未来的发展中，信息技术将继续推动社会进步，同时也需要我们共同面对和解决由此带来的各种问题。

2. 计算机硬件与软件

计算机硬件与软件

(1) 计算机硬件是计算机系统的物理组成部分，包括处理器、内存、存储设备、输入输出设备等。处理器作为计算机的大脑，负责执行指令、处理数据和进行计算；内存则用于存储当前运行的程序和数据；存储设备包括硬盘、固态硬盘等，用于长期保存数据；输入输出设备如键盘、鼠标、显示器等，是用户与计算机进行交互的桥梁。硬件的配置直接影响计算机的性能和运行速度。

(2) 软件是计算机的灵魂，是计算机硬件执行任务的指令集合。软件可以分为系统软件和应用软件两大类。系统软件包括操作系统、编译器、数据库管理系统等，负责管理和控制计算机硬件资源，为应用软件提供运行平台。应用软件则根据用户需求，提供各种功能和服务，如办公软件、图形设计软件、游戏软件等。软件的开发需要遵循一定的编程语

言和开发工具，以满足不同用户的需求。

(3)

随着技术的不断进步，计算机硬件和软件都在不断发展和创新。硬件方面，从早期的 286、386 处理器到现在的多核处理器，从机械硬盘到固态硬盘，计算机的性能得到了极大提升。软件方面，操作系统越来越智能化、易用化，应用软件的功能也越来越丰富、高效。此外，随着互联网的普及，云计算、大数据、人工智能等新技术也逐渐融入计算机硬件与软件的发展中，为我们的生活带来了更多可能性。计算机硬件与软件的紧密融合，共同推动了信息时代的到来。

3. 网络基础

网络基础

(1) 网络基础是信息技术领域的重要部分，它涉及到计算机网络的构建、运行和维护。计算机网络通过物理介质和通信协议，将分布在不同地理位置的计算机连接起来，实现信息的高速传输和资源共享。网络基础包括网络拓扑结构、网络协议、网络设备等多个方面。在网络拓扑结构中，常见的有星型、环型、总线型和网状等，每种结构都有其特点和适用场景。网络协议则是网络通信的规则，如 TCP/IP 协议，它定义了数据在网络中的传输方式。

(2) 网络设备是网络通信的关键组成部分，包括路由器、交换机、集线器、网关等。路由器负责在不同网络之间转发数据包，交换机则用于在同一网络内进行数据包的交换，集线器则是一种较为简单的网络设备，用于连接多个设备。网关则是一种连接不同类型网络的设备，如将局域网连接到互

联网。这些设备共同构成了网络的基础设施，保障了网络的高效运行。

(3) 网络安全是网络基础中的重要环节，它涉及到如何保护网络免受恶意攻击、数据泄露和非法访问。网络安全措施包括防火墙、入侵检测系统、加密技术等。防火墙用于监控和控制进出网络的流量，防止未授权访问；入侵检测系统用于检测网络中的异常行为，及时发现并阻止攻击；加密技术则用于保护数据在传输过程中的安全，防止数据被窃取。随着网络技术的不断发展，网络安全问题日益突出，因此加强网络安全意识，提高网络安全防护能力，对于维护网络稳定 and 用户利益至关重要。

4. 信息安全意识

信息安全意识

(1) 在信息时代，信息安全意识已经成为每个人都需要具备的基本素养。随着网络技术的普及和互联网的深入应用，个人信息、企业数据、国家机密等面临着前所未有的安全风险。信息安全意识要求我们时刻保持警惕，对可能存在的安全威胁有所了解，并采取相应的预防措施。这包括对网络钓鱼、恶意软件、社交工程等常见攻击手段的认识，以及对个人隐私保护、数据加密、安全认证等基本安全措施的掌握。

(2)

信息安全意识的培养需要从多个层面进行。首先，个人应当了解网络安全的基本知识，包括如何识别可疑链接、如何设置强密码、如何安全地使用公共 Wi-Fi 等。其次，企业和组织应当建立健全的信息安全管理制度，对员工进行定期的安全培训，确保每个人都能够遵守安全规范。此外，国家层面也需要制定相应的法律法规，加强对网络安全的监管，打击网络犯罪活动。

(3) 信息安全意识不仅仅是一种自我保护的行为，它还涉及到社会责任和道德伦理。在信息共享和交流的过程中，我们应当尊重他人的隐私和数据安全，不传播和滥用个人信息。同时，对于发现的安全漏洞和风险，应当及时报告并采取措施进行修复，以防止潜在的安全事故发生。通过提高信息安全意识，我们可以共同构建一个安全、可靠、和谐的网络环境。

二、计算机操作与应用

1. 操作系统基础

操作系统基础

(1) 操作系统是计算机系统的核心软件，它负责管理和控制计算机硬件资源，提供用户与计算机之间的交互界面，并确保计算机系统的稳定运行。操作系统的主要功能包括进程管理、内存管理、文件系统管理、设备管理和用户接口等。在进程管理方面，操作系统负责创建、调度和终止进程，确保多个进程能够高效地共享 CPU 资源。内存管理则负责分配

和回收内存空间，保证程序运行时所需的内存资源。文件系统管理涉及文件的创建、读取、修改和删除等操作，为用户提供数据存储和访问的便利。设备管理负责控制和管理计算机的各种硬件设备，如打印机、扫描仪等。

(2)

操作系统的发展经历了多个阶段，从早期的单用户、单任务操作系统，如 MS-DOS，到多用户、多任务操作系统，如 Windows 和 Unix。现代操作系统不仅具备强大的多任务处理能力，还提供了图形用户界面（GUI），使得用户能够更加直观地操作计算机。操作系统还支持网络功能，允许用户通过网络进行远程登录、文件传输和资源共享。此外，随着云计算的兴起，操作系统也在不断适应新的技术和应用需求，如虚拟化、容器化等。

(3) 操作系统的性能和稳定性是评价其优劣的重要指标。性能方面，操作系统需要具备高效的 CPU 调度算法、内存分配策略和文件系统设计，以确保系统资源的合理利用和程序的快速响应。稳定性则要求操作系统在处理各种异常情况时能够保持正常运行，如硬件故障、软件错误和用户误操作等。为了提高操作系统的性能和稳定性，厂商通常会进行持续的优化和更新，引入新的技术和管理方法，以适应不断变化的技术环境和用户需求。

2. 文字处理软件应用

文字处理软件应用

(1) 文字处理软件是现代办公中不可或缺的工具，它为用户提供了创建、编辑、格式化和打印文档的功能。从简单的文档排版到复杂的报告撰写，文字处理软件能够满足各种文字编辑需求。常见的文字处理软件包括 Microsoft Word、WPS

Office 等，它们提供了丰富的文本编辑功能，如字体、字号、段落格式、页眉页脚设置等，使得文档的编辑和排版更加便捷。

(2) 文字处理软件的应用场景十分广泛。在学术领域，研究人员可以使用文字处理软件撰写论文、报告；在商业领域，企业员工可以利用文字处理软件制作合同、报告、宣传册等；在教育领域，教师和学生可以通过文字处理软件编写教案、作业、论文等。此外，文字处理软件还支持多人协作编辑，方便团队共同完成文档工作。

(3) 随着技术的不断发展，文字处理软件的功能也在不断丰富。例如，许多现代文字处理软件支持云存储，用户可以将文档保存在云端，随时随地访问和编辑。此外，文字处理软件还具备语音输入、翻译、OCR（光学字符识别）等功能，极大地提高了工作效率。同时，随着移动设备的普及，许多文字处理软件也推出了移动版，用户可以在手机或平板电脑上随时随地处理文档，极大地提高了办公的灵活性。

3. 电子表格软件应用

电子表格软件应用

(1) 电子表格软件是现代办公中用于数据管理和分析的重要工具。它通过二维表格的形式，帮助用户组织和处理数据，进行计算、分析和可视化。常见的电子表格软件包括 Microsoft Excel、WPS 表格等，它们具备强大的数据处理和分析功能，广泛应用于财务、统计、科研、教育等多个领域。

(2)

电子表格软件的应用场景十分广泛。在财务管理中，它可以用于制作资产负债表、利润表、现金流量表等财务报表，帮助财务人员快速处理和分析财务数据。在统计分析领域，电子表格软件可以用于数据录入、清洗、转换和分析，为科研人员提供数据支持。在教育领域，教师和学生可以利用电子表格软件进行数据记录、成绩管理、教学计划制定等。此外，电子表格软件还支持数据可视化，通过图表和图形展示数据趋势，使数据更加直观易懂。

(3) 随着电子表格软件功能的不断完善，它们在数据处理和分析方面的能力也得到了显著提升。例如，Excel 提供了丰富的函数和公式，可以用于复杂的计算和数据分析；WPS 表格则支持宏录制和自动化操作，提高工作效率。此外，许多电子表格软件还具备数据连接功能，可以连接数据库、外部数据源，实现数据的实时更新和同步。随着大数据和云计算的发展，电子表格软件也在不断适应新的技术和应用需求，如数据挖掘、机器学习等，为用户提供了更加全面的数据处理和分析解决方案。

4. 演示文稿制作

演示文稿制作

(1) 演示文稿是一种视觉辅助工具，广泛应用于学术报告、商务演讲、产品介绍等场合。它通过图文并茂的方式，将信息直观地展示给观众，有助于增强演讲效果和记忆点。演示文稿制作软件如 Microsoft PowerPoint、WPS 演示等，

提供了丰富的模板、动画效果和图表工具，帮助用户轻松创建专业级的演示文稿。

(2) 演示文稿的制作过程包括选题、内容策划、设计制作和演讲准备等环节。选题是演示文稿的基础，需要根据演讲目的和受众特点进行选择。内容策划则是对演示文稿的整体构思，包括结构安排、信息组织和重点突出。设计制作阶段，用户可以根据需要选择合适的模板，并利用软件提供的功能进行个性化设计，如添加图片、图表、动画等。演讲准备则是对演示文稿的预演和优化，确保演讲过程中流畅自然。

(3) 演示文稿的成功与否，不仅取决于内容的质量，还与视觉效果和演讲技巧密切相关。在视觉效果方面，应注重色彩的搭配、字体的选择和版式的布局，使演示文稿既美观又易于阅读。在演讲技巧方面，演讲者应熟悉演示文稿的内容，掌握好演讲节奏，与观众进行互动，以提高演讲的吸引力和说服力。此外，随着技术的发展，演示文稿制作软件也不断推出新的功能，如远程演示、协作编辑等，为用户提供了更加便捷和高效的制作体验。

三、程序设计基础

1. 程序设计基本概念

程序设计基本概念

(1)

程序设计是计算机科学的核心领域之一，它涉及到编写指令集，使计算机能够执行特定任务。程序设计的基本概念包括算法、数据结构、控制结构、编程语言和调试等。算法是解决问题的步骤集合，是程序设计的灵魂。数据结构则是组织和管理数据的方式，如数组、链表、树和图等。控制结构决定了程序执行的流程，包括顺序、选择和循环等。编程语言是用于编写程序的工具，不同的语言有不同的语法和特点。调试则是程序开发过程中的重要环节，用于识别和修复程序中的错误。

(2) 程序设计的过程可以分为需求分析、设计、编码、测试和维护等阶段。需求分析阶段，开发者需要与用户沟通，明确程序的功能和性能要求。设计阶段，开发者根据需求分析的结果，设计程序的架构和模块。编码阶段，开发者使用编程语言将设计转化为实际的代码。测试阶段，开发者对程序进行测试，确保其符合预期功能。维护阶段则是对已发布的程序进行更新和修复，以适应新的需求和技术变化。

(3) 程序设计不仅是一门技术，更是一种思维方式。它要求开发者具备逻辑思维、问题解决能力和创新精神。在程序设计过程中，开发者需要不断学习新的技术和方法，以提高编程效率和程序质量。随着软件工程的不断发展，敏捷开发、DevOps 等新的开发模式也应运而生，它们强调快速迭代、持续集成和自动化测试，以适应快速变化的市场需求。程序设计的基本概念和技能对于培养开发者的综合素质和职业

发展具有重要意义。

2. 编程语言基础

编程语言基础

(1)

编程语言是用于编写计算机程序的语言，它是人与计算机之间的沟通桥梁。不同的编程语言有着不同的语法和特点，但它们都遵循一定的基本概念，如变量、数据类型、控制结构、函数和对象等。变量用于存储数据，数据类型决定了数据的存储方式和操作方法。控制结构包括条件语句和循环语句，用于控制程序的执行流程。函数是代码的模块化，它可以将复杂的任务分解为可重用的单元。对象则是面向对象编程中的基本概念，它将数据和操作数据的方法封装在一起。

(2) 编程语言可以分为多种类型，包括过程式语言、面向对象语言、函数式语言和逻辑语言等。过程式语言如 C 语言，强调程序的执行流程，通过函数和循环实现复杂逻辑。面向对象语言如 Java 和 Python，强调数据和方法的封装，通过类和对象实现模块化和重用。函数式语言如 Haskell，强调函数的不可变性，通过高阶函数和递归实现编程。逻辑语言如 Prolog，强调逻辑推理，通过模式匹配和规则实现编程。

(3)

学习编程语言的基础，首先要掌握其语法和基本概念，然后通过编写简单的程序来实践。编程语言的入门通常从基础语法开始，如变量声明、数据类型、运算符、控制结构等。随着学习深入，开发者需要掌握更高级的概念，如面向对象设计、异常处理、文件操作等。此外，了解编程语言的生态系统和库也是非常重要的，因为它们提供了丰富的功能和工具，可以帮助开发者更高效地开发程序。不同的编程语言适用于不同的应用场景，因此选择合适的编程语言对于开发成功的产品至关重要。

3. 算法与数据结构

算法与数据结构

(1) 算法与数据结构是计算机科学中的核心概念，它们是解决问题的关键。算法是一系列解决问题的步骤，它指导计算机如何高效地处理数据。数据结构则是组织数据的方式，它决定了数据的存储、访问和操作效率。算法与数据结构的学习对于提高编程能力和解决复杂问题至关重要。常见的算法包括排序、搜索、图论算法等，而数据结构包括数组、链表、栈、队列、树、图等。

(2) 算法的效率通常通过时间复杂度和空间复杂度来衡量。时间复杂度表示算法执行所需的时间与输入规模的关系，而空间复杂度表示算法执行过程中所需内存的大小。在设计算法时，需要考虑时间复杂度和空间复杂度的平衡，以实现最优的性能。数据结构的选择也会影响算法的效率，例

如，使用哈希表进行快速查找，使用平衡树进行高效插入和删除操作。

(3)

算法与数据结构的学习不仅需要掌握理论知识，还需要大量的实践。通过编写和调试代码，可以加深对算法和数据结构的理解。在实际应用中，开发者需要根据具体问题选择合适的算法和数据结构，以实现高效的数据处理。此外，随着技术的发展，新的算法和数据结构不断涌现，如分布式算法、并行算法、内存优化数据结构等，这些新技术为解决大规模数据处理问题提供了新的思路和方法。因此，持续学习和掌握算法与数据结构是计算机科学领域的重要任务。

4. 程序调试与测试

程序调试与测试

(1) 程序调试与测试是软件开发过程中的关键环节，它们确保了软件产品的质量和可靠性。程序调试是指识别和修复程序中的错误（bug）的过程。调试的目的是使程序按照预期的方式运行，解决软件在开发过程中可能出现的问题。调试工具和技术包括断点设置、单步执行、查看变量值、分析程序流程等。有效的调试技巧可以帮助开发者更快地定位问题并找到解决方案。

(2) 程序测试则是在软件开发生命周期的不同阶段，通过运行程序来验证其行为是否符合预期。测试的主要目的是发现软件中的缺陷和错误，确保软件的功能、性能和稳定性。测试方法包括单元测试、集成测试、系统测试、验收测试等。单元测试是对程序中最小的可测试部分进行测试；集成测试则测试各模块组合在一起时的交互；系统测试是对整个软件

系统进行的全面测试；验收测试则是在软件交付用户前进行的测试，以确保软件满足用户需求。

(3)

程序调试与测试是一个迭代的过程。在软件开发过程中，可能会不断发现新的错误和缺陷，因此需要反复进行调试和测试。为了提高调试和测试的效率，开发者通常会采用一些最佳实践，如编写可测试的代码、使用自动化测试工具、维护详细的测试文档等。自动化测试工具可以帮助开发者快速执行大量的测试用例，提高测试的覆盖率和效率。此外，持续集成和持续部署（CI/CD）流程的引入，使得测试过程更加自动化和高效，有助于缩短软件开发周期，提高软件质量。

四、数据库基础

1. 数据库基本概念

数据库基本概念

(1) 数据库是用于存储、管理和检索数据的系统。它由数据、数据库管理系统（DBMS）、应用程序和用户组成。数据库中的数据以结构化的形式存储，便于查询和管理。数据库管理系统负责处理数据的存储、检索、更新和删除等操作，同时确保数据的一致性和完整性。数据库的应用领域广泛，包括企业资源规划（ERP）、客户关系管理（CRM）、电子商务、科学研究和政府管理等。

(2) 数据库的基本概念包括数据库模式、表、记录、字段、索引和视图等。数据库模式是数据库的逻辑结构，定义了数据库的组织方式和数据类型。表是数据库中存储数据的基本单位，由记录和字段组成。记录是表中的行，代表一个

具体的数据实体；字段是记录中的列，代表数据的属性。索引是一种数据结构，用于加速数据的检索；视图则是虚拟表，它基于一个或多个表的数据生成，可以简化复杂的查询操作。

(3)

数据库设计是数据库开发的重要环节，它涉及到如何组织和结构化数据，以满足特定应用的需求。数据库设计包括概念设计、逻辑设计和物理设计。概念设计阶段，开发者通过实体-关系模型（ER 模型）来描述现实世界的实体及其关系；逻辑设计阶段，将概念设计转换为具体的数据库模式，如关系模型；物理设计阶段，则关注数据库在物理存储层面的实现，如索引策略、存储分配等。良好的数据库设计可以提高数据的一致性、完整性和可扩展性，同时降低维护成本。随着技术的发展，数据库技术也在不断演进，如 NoSQL 数据库的兴起，为处理大规模非结构化数据提供了新的解决方案。

2. 关系型数据库

关系型数据库

(1) 关系型数据库是一种基于关系模型的数据库管理系统，它使用表格（也称为关系）来存储数据。在关系型数据库中，数据以行和列的形式组织，每一行代表一个记录，每一列代表一个字段。这种结构使得数据易于理解和管理。关系型数据库的核心是关系代数，它提供了一套操作关系（即表格）的数学规则，包括选择、投影、连接和并集等操作。

(2)

关系型数据库的主要特点包括数据的一致性、完整性、事务性和可扩展性。数据一致性确保了数据库中数据的准确性和可靠性；完整性约束（如主键、外键、唯一性约束等）用于防止数据不一致；事务性保证了一系列操作要么全部完成，要么全部不做，从而维护数据库的稳定性和可靠性；可扩展性则允许数据库随着数据量的增长而扩展。

(3) 常见的商业关系型数据库管理系统包括 Oracle、Microsoft SQL Server、IBM DB2 和 MySQL 等。这些系统提供了丰富的功能和工具，支持复杂的数据查询、事务处理和用户管理。关系型数据库的查询语言 SQL（Structured Query Language）是数据库操作的标准语言，它允许用户执行各种数据操作，如创建表、插入数据、更新数据、删除数据以及执行复杂的查询。随着云计算的普及，许多关系型数据库管理系统也提供了云服务，使得数据库的部署和管理更加灵活和高效。

3. 数据库设计

数据库设计

(1) 数据库设计是数据库开发过程中的关键步骤，它涉及到如何将现实世界中的实体和关系转化为数据库中的表和关系。一个良好的数据库设计能够提高数据的一致性、完整性和可维护性，同时降低系统的复杂性和维护成本。数据库设计通常分为概念设计、逻辑设计和物理设计三个阶段。概念设计阶段，开发者通过实体-关系模型（ER 模型）来抽

象现实世界的业务需求，定义实体、属性和关系。逻辑设计阶段，将概念模型转换为具体的数据库模式，如关系模型，并定义表结构、字段类型和约束。物理设计阶段，则关注数据库在物理存储层面的实现，如索引策略、存储分配等。

(2) 在数据库设计过程中，需要考虑多个因素，包括业务需求、数据完整性、性能优化和安全性。业务需求决定了数据库需要存储哪些数据以及如何组织这些数据。数据完整性确保了数据库中数据的准确性和一致性，通常通过主键、外键、唯一性约束和检查约束来实现。性能优化涉及数据库的查询性能、索引策略和数据分区等，以提高数据检索和处理的速度。安全性则包括用户认证、访问控制和数据加密等，以保护数据库免受未授权访问和攻击。

(3) 数据库设计是一个迭代和逐步完善的过程。在设计初期，可能需要与业务分析师和用户进行多次沟通，以充分理解业务需求和用户场景。随着设计的深入，开发者可能会发现一些潜在的问题，如数据冗余、更新异常等，这时需要对设计进行调整和优化。在实际应用中，数据库设计还需要考虑扩展性，以便在未来能够适应业务增长和数据量的增加。因此，数据库设计不仅仅是技术层面的工作，更是业务理解和系统规划的重要环节。

4. SQL 语言基础

SQL 语言基础

(1) SQL (Structured Query

Language) 是一种用于数据库管理的标准查询语言, 它广泛应用于各种关系型数据库系统中。SQL 语言包括数据定义语言 (DDL)、数据操纵语言 (DML)、数据控制语言 (DCL) 和数据查询语言 (DQL) 四个部分。DDL 用于定义数据库结构, 如创建表、修改表结构、删除表等; DML 用于插入、更新、删除数据; DCL 用于控制数据库的访问权限; DQL 则是用于查询数据库中的数据, 如 SELECT 语句。

(2) SELECT 语句是 SQL 语言中最常用的查询语句, 它允许用户从数据库中选择特定的数据。SELECT 语句的基本语法包括 SELECT 关键字后跟字段列表, FROM 关键字后跟表名, 以及 WHERE 关键字后跟条件表达式。例如, ``SELECT name, age FROM users WHERE age > 18`` 这条语句会从 users 表中选出年龄大于 18 岁的用户的姓名和年龄。此外, SQL 还支持 JOIN 操作, 用于连接多个表中的数据, 以及 GROUP BY 和 ORDER BY 子句, 用于对结果进行分组和排序。

(3) SQL 语言提供了一系列内置函数和聚合函数, 用于处理和汇总数据。内置函数包括字符串函数、日期和时间函数、数值函数和转换函数等, 它们可以用于格式化数据、计算日期差、执行数学运算等。聚合函数如 SUM、AVG、COUNT、MAX 和 MIN 等, 用于对一组数据进行汇总和计算, 例如, ``SELECT SUM(salary) FROM employees`` 会计算 employees 表中所有员工的薪资总和。掌握这些函数对于编写复杂的 SQL 查询和进行数据统计分析至关重要。随着 SQL 语言的不

断发展，许多数据库系统还支持窗口函数、CTE（公用表表达式）等高级特性，进一步丰富了 SQL 语言的功能和应用范围。

五、网络编程基础

1. 网络编程基本概念

网络编程基本概念

(1) 网络编程是利用编程语言和网络协议开发网络应用程序的过程。它涉及到如何通过计算机网络的物理层、数据链路层、网络层、传输层和应用层进行数据传输和通信。网络编程的基本概念包括网络协议、数据传输模型、网络编程接口和网络编程框架等。网络协议定义了数据在网络中的传输规则，如 TCP/IP 协议族。数据传输模型包括客户端-服务器模型和 peer-to-peer（对等网络）模型。网络编程接口（如 Winsock、libcurl）提供了网络编程的 API，而网络编程框架（如 Node.js、Django）则简化了网络应用程序的开发。

(2) 网络编程的核心是理解网络协议和数据传输过程。TCP（传输控制协议）和 UDP（用户数据报协议）是两种最常用的传输层协议。TCP 提供可靠的、面向连接的服务，适用于需要保证数据完整性和顺序的应用程序；UDP 则提供不可靠的、无连接的服务，适用于对实时性要求高、对数据完整性要求不严格的应用程序。在网络编程中，开发者需要处理网络编程接口提供的 API，如创建套接字、绑定地址、监听连接、发送和接收数据等。

(3)

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/187060025003010055>