软通教育

《ts模块功能实现》实验手册

|  |  |
| --- | --- |
| 专 业 | 计算机科学与技术 |
| 班 级 | 2班 |
| 姓 名 | 张渊 |
| 学 号 | 2021122155623 |
| 授课教师 | 郑少彬 |

北京软通动力教育科技有限公司

2024/08/29

1. 实验目的

1.TypeScript基础：了解其基本语法，如类、方法、访问控制修饰符，掌握类型系统和接口，通过实际代码加深理解，并在JavaScript环境中使用TypeScript。

2.面向对象编程（OOP）：掌握封装、继承和多态，学会定义类、创建对象、封装属性和方法，并利用继承扩展类功能。

3.身份验证系统：理解密码存储（哈希处理）、验证用户输入合法性、管理用户账户等基本原理和实现方式。

4.项目管理：学习如何组织代码文件、设置项目结构、配置编译选项，使用npm管理依赖关系，并编写package.json自动化构建和运行过程。

5.通过开发图书管理系统，可以深入理解TypeScript的类定义、构造函数、属性和方法，实践面向对象编程。编写代码和解决问题锻炼编程技能和调试技巧。实现查询功能时，学习优化查询效率的不同数据结构和算法，并理解其优缺点。编写清晰、简洁的代码提升可读性和可维护性，实践良好的编码习惯。

6.掌握面向对象编程的基本概念和方法，能够设计和实现包含多个类和对象的复杂系统。深入了解类与对象之间的关系，包括继承、封装和多态等特性，并通过实践提升面向对象编程的能力。实现学生、课程、成绩等基本数据结构的定义和管理，包括属性的设置和方法的实现。培养学生数据管理和组织的能力，通过构建学生成绩管理系统，学会如何有效地管理、存储和检索数据。学习并模拟基本的数据库操作，包括数据的添加、更新、删除和查询等，为日后的实际应用打下坚实基础。提高学生的问题分析和解决能力，通过面对实际需求和场景，学会如何将复杂问题分解为若干个小问题，并逐一解决。

1. 实验涉及知识点
2. 掌握Flex弹性布局
3. 掌握容器组件Column、Row
4. 掌握基础组件Text、TextInput、Button等
5. TypeScript基础：变量、类型声明、类的使用、构造函数、访问修饰符和方法声明。
6. 面向对象编程：封装、继承、对象创建和实例化。
7. 项目管理：使用Node.js和npm进行项目初始化、依赖管理和TypeScript配置。
8. 编译与构建：TypeScript编译器配置（目标版本、模块系统、严格模式）及npm脚本自动化构建。
9. 调试与日志：使用console.log()进行调试，处理错误和异常。
10. 实验准备
11. 技能要求：

建议具备基本的编程知识。

1. 实验环境要求：

* 操作系统Windows
* 安装DevEco Studio

1. 实验内容及课时分配

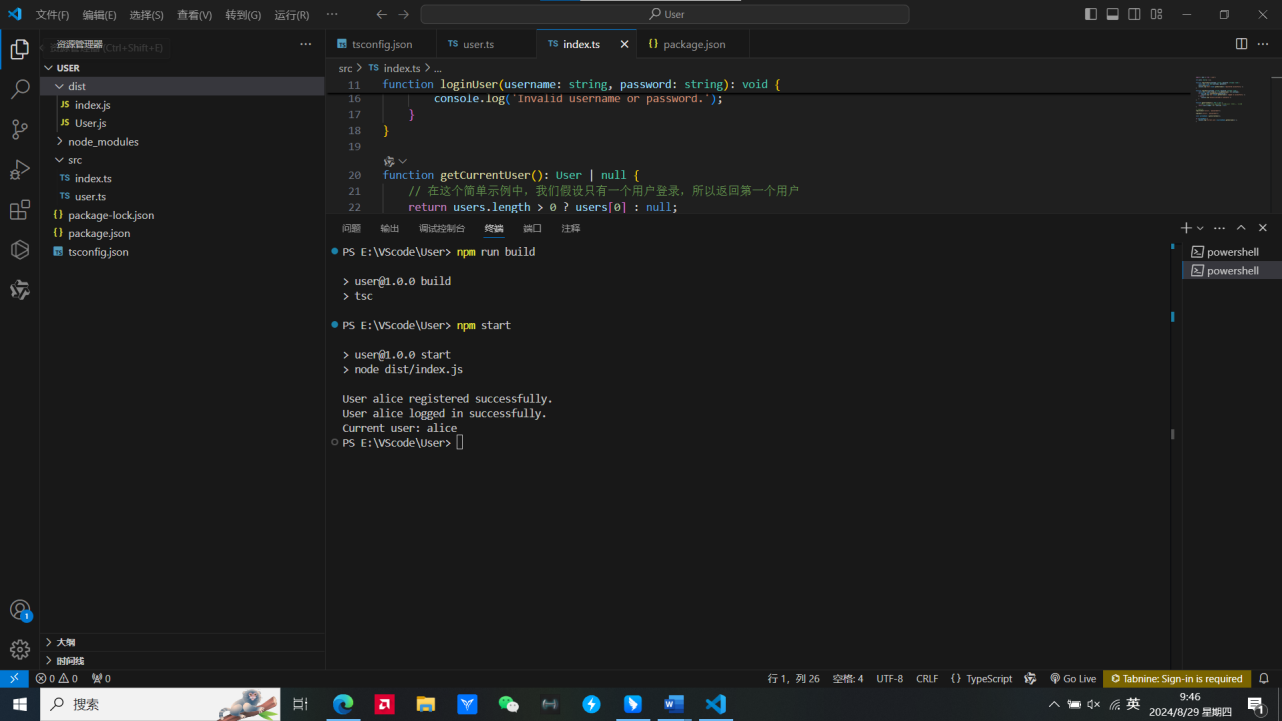
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 实验内容 | 实验课时 | 对应核心知识点 |
| 1 | 用户身份验证模块 | 2 | 1、2、3、4 |
| 2 | 图书管理系统 | 2 | 5 |
| 3 | 学生成绩管理系统 | 2 | 6 |

1. 详细实验过程

实验一

电脑萤幕的截图

描述已自动生成



**1.**初始化项目和安装依赖：创建一个新的NPM项目，并安装TypeScript、TSLint和NodeJS的类型声明。

**2.**编辑tsconfig.json文件：配置TypeScript编译选项，包括目标版本、模块系统、严格模式等。

**3.**创建User类：定义用户的基本信息，包括用户名、密码等。实现构造方法、哈希密码方法和校验密码方法。

**4.**创建index.ts文件：实现用户注册、登录和获取当前用户的方法。编写测试代码，验证功能是否正常。

**5.**编译和运行：在package.json中添加编译和运行脚本，然后执行npm run build和npm start命令，编译项目并运行。

实验概述：本实验通过一个简单的用户身份验证模块，让学生了解如何使用TypeScript编写面向对象程序，掌握OOP的基本概念（封装、继承、多态），以及如何实现用户身份验证逻辑。同时，学生还需要学会如何组织代码文件、设置项目结构、配置编译选项等。

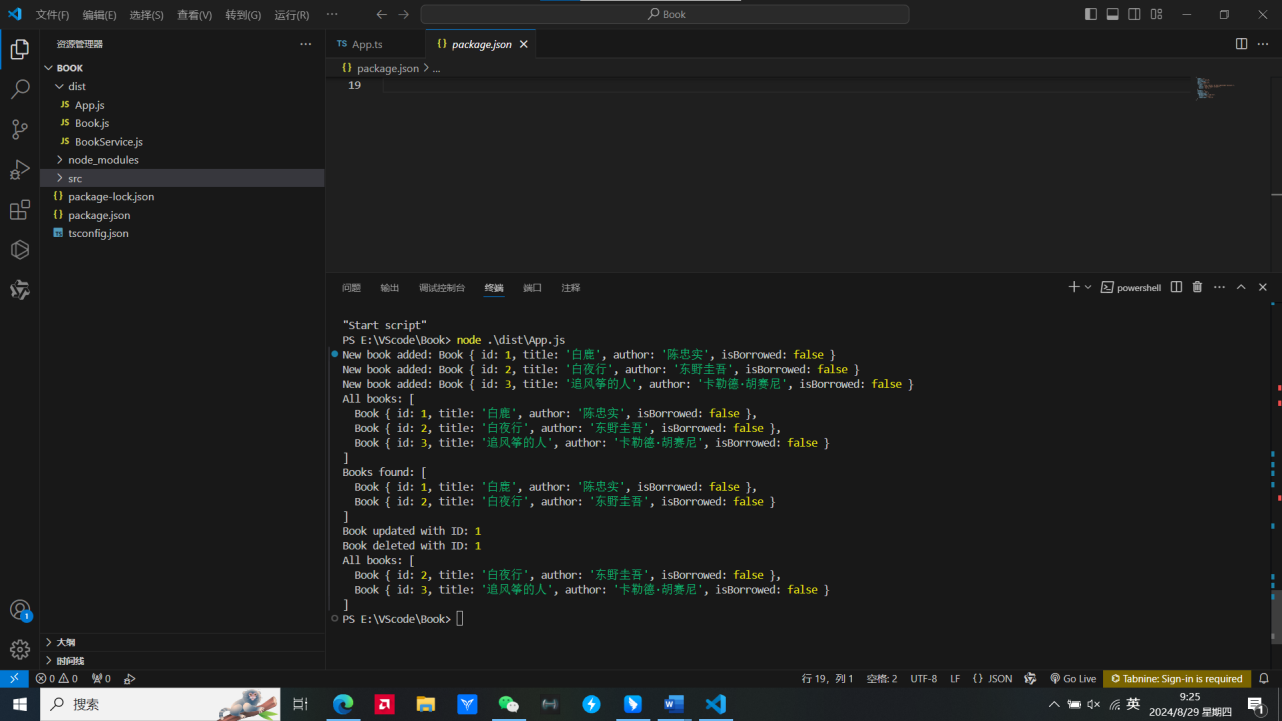
实验二

电脑萤幕的截图

描述已自动生成

电脑萤幕的截图

描述已自动生成



1. 项目初始化与依赖安装

创建新的NPM项目。安装TypeScript、TSLint和NodeJS类型声明。

1. 配置 TypeScript

在项目根目录创建tsconfig.json文件，设置编译选项。

1. 定义图书类 (Book)

创建Book.ts文件，定义Book类，包括属性和方法来管理图书信息。

1. 定义图书服务类 (BookService)

创建BookService.ts文件，定义BookService类，实现添加、删除、更新、搜索和获取所有图书的方法。

1. 使用图书服务类

在App.ts文件中，使用BookService类来调用各种图书管理方法。

1. 验证代码实现效果

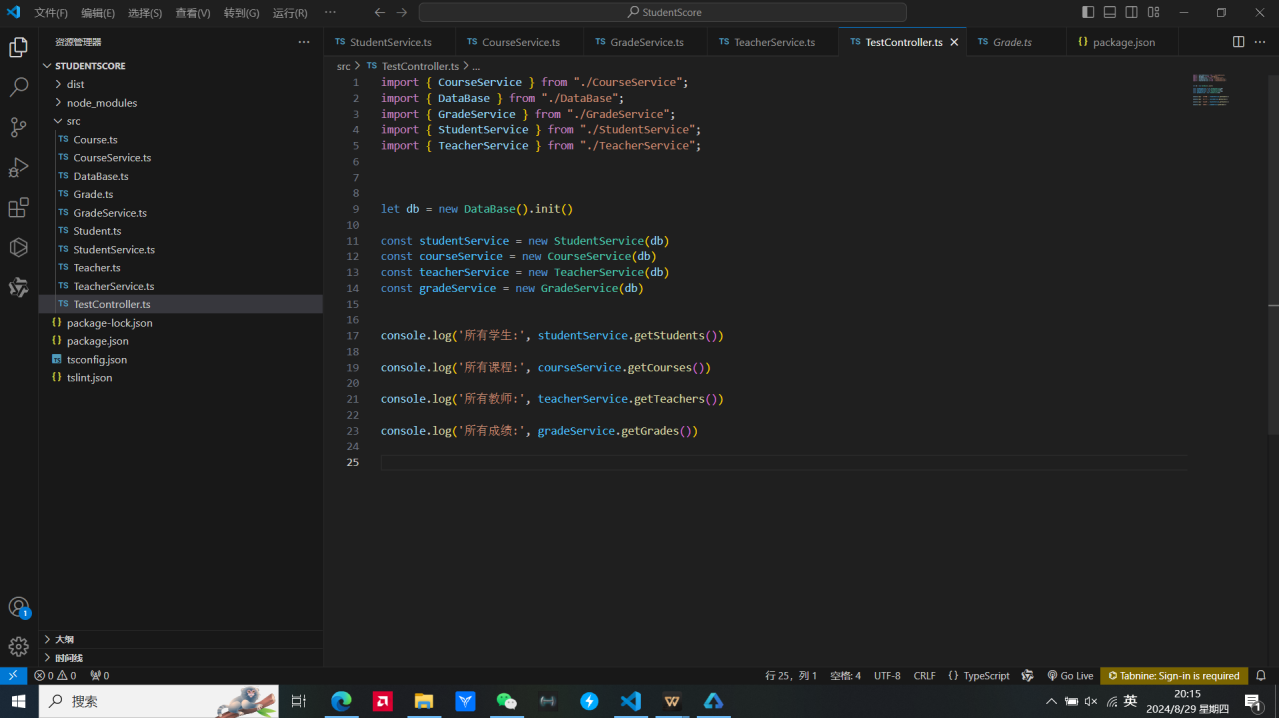
实例化App类，并调用其方法以验证图书管理系统的功能。

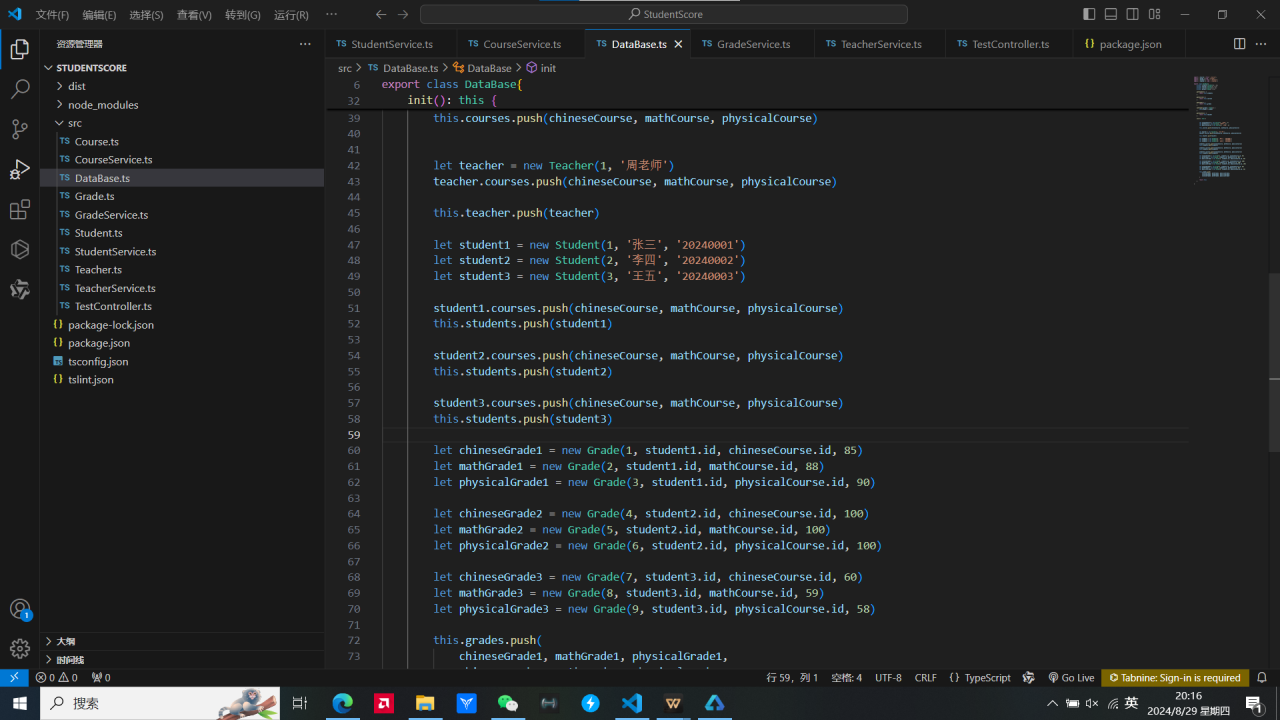
1. 编译和运行项目

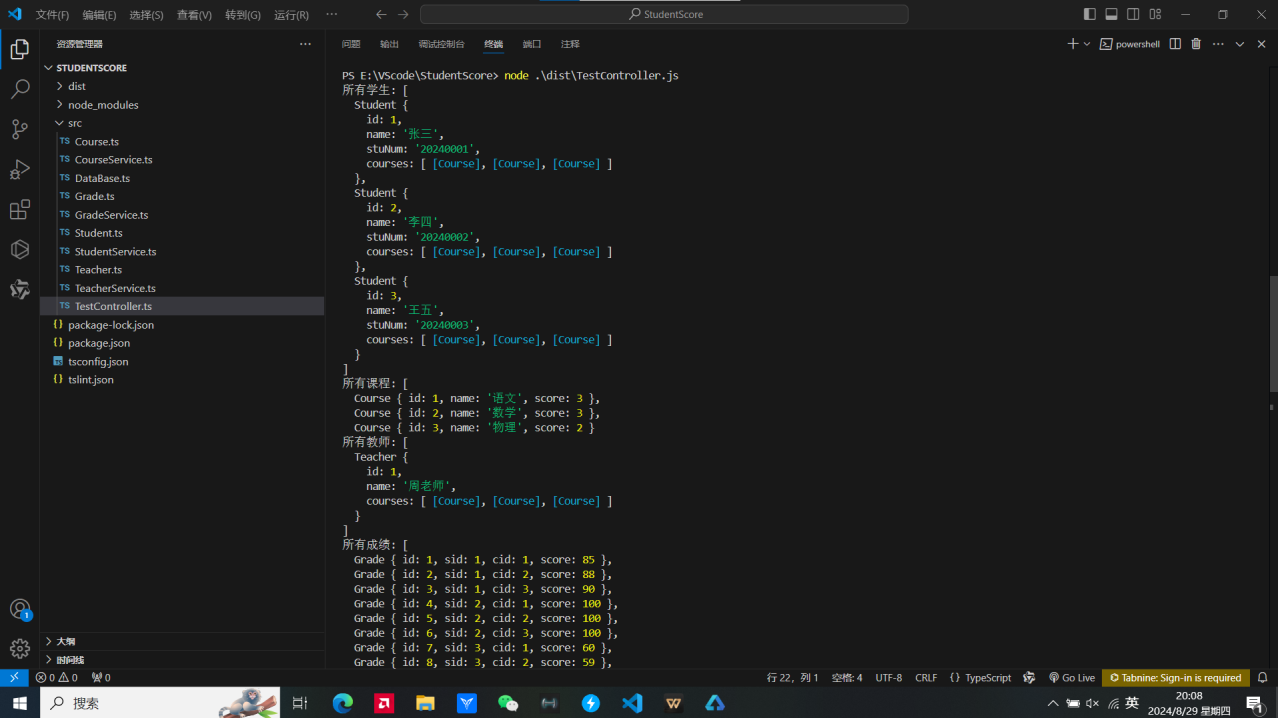
通过npm run build命令编译 TypeScript 到 JavaScript。通过npm start命令运行项目。

**创建一个新的NPM项目并安装TypeScript、TSLint和NodeJS类型声明以初始化开发环境。接着，在项目的根目录中创建一个tsconfig.json文件，用于配置TypeScript编译选项。然后，在Book.ts文件中定义一个Book类，负责管理图书信息。接下来，在BookService.ts文件中定义BookService类，实现图书的添加、删除、更新、搜索和获取所有图书等功能。之后，在App.ts文件中利用BookService类来调用这些图书管理方法。最后，通过实例化App类并调用其方法来验证系统的功能。完成以上步骤后，使用npm run build命令编译TypeScript代码，并通过npm start命令运行项目，以确保一切正常运行**

实验三







1.环境搭建与配置：

安装Node.js和npm（Node包管理器）。选择并配置文本编辑器或集成开发环境（IDE），如Visual Studio Code。创建新的NPM项目，初始化项目文件和设置TypeScript编译选项。

2.定义实体类：

设计并实现学生、课程、成绩和教师等实体类，包括其属性和方法。确保每个实体类具有清晰定义的责任，例如学生类管理学生信息，课程类管理课程数据。

3.模拟数据库操作：

实现DataBase类来模拟数据库的操作，使用数组存储和管理不同实体的数据。提供方法以添加、更新、删除和查询数据，模拟真实数据库的基本功能。

4.系统设计与实现：

根据需求分析，设计系统架构和界面流程。编码实现系统的核心功能，包括但不限于学生信息的增删改查、成绩的录入与查询等。