基于C#的抽题系统开发

吉林大学计算机学院

21210613 马周原

55210916 周宇恒

21210713 李光赫

19210215 邹坤成


需求分析
01
02
实现细节
系统演示
03
04
组员分工

需求分析

- **读取题库：**题库加载（文本、图片、公式）、更改题库路径
- **显示题目：**放大缩小、特殊字符、超文本（如图片、公式）

- **抽题机制：**随机抽题（不重复）
- **状态维护：**状态保存、状态加载、状态重置。

- **UI设计：**美观性、文字提示

读取题库

```
dataset/
├── 题库1/
│   ├── img/
│   │   └── xxx.png
│   └── problems.txt
└── 题库2/
```

图：题库结构

需求：软件应能在多个包含纯文本、图片、公式以及组合格式题目的题库中进行随机抽题操作

实现：所有题库都位于一个可指定的目录下（dataset），软件启动时自动加载该目录下所有题库中的文本文件（problems.txt）作为题目。

图片存储的相对路径

LaTex公式

```
[IMAGE:./img/1.png] 设函数$f(x)$在区间$(-1, 1)$上有定义，且$\lim _{x \to 0}=0$，则（）
```

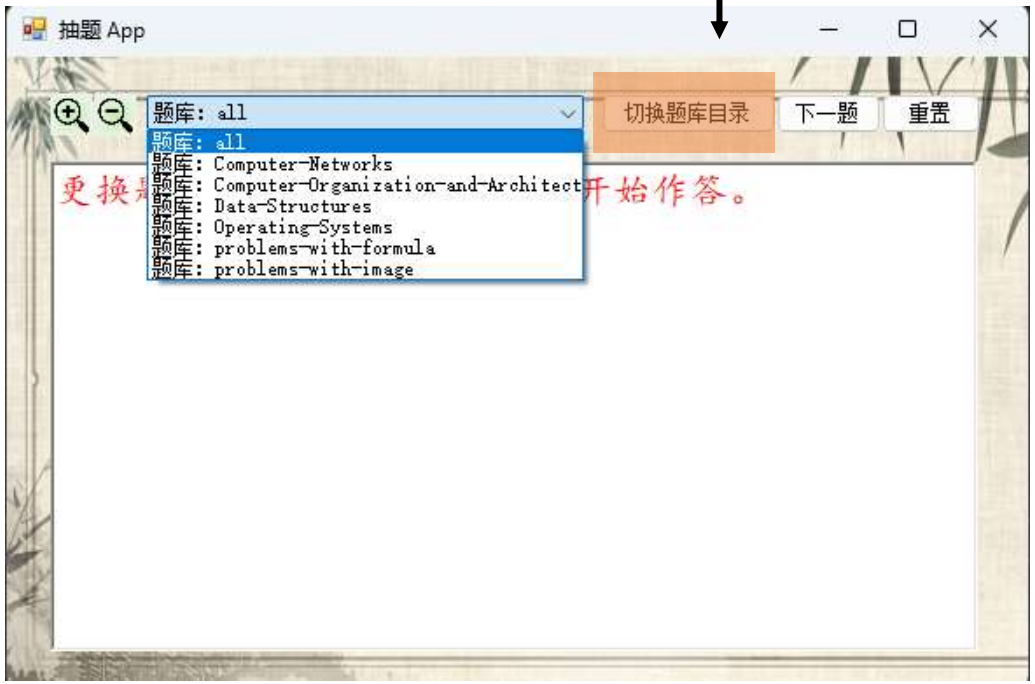
图：problems.txt中某一道题

读取题库

```
internal class DataLoader
{
    // 字典用于存储每个文件夹中的题目
    private Dictionary<string, string[]> problemData;
    private List<string> folders;
    private int problemCount;
    int seeds;
    private string myDataFolderPath;
    Dictionary<string, int> suf_count_problemset;
    Dictionary<string, int> used_for_problemset;
    public string allfolder;

    // 构造函数，传入data文件夹路径
    public DataLoader(string dataFolderPath)...

    // 读取data文件夹下所有子题库中的problems.txt文件
    private void LoadData(string dataFolderPath)...

    private void WriteData()...

    // 对外接口，获取指定文件夹下指定索引的题目
    private int GetProblemIndex(string folder, int prob)...
    private int GetNumberIndex(string folder, int prob)...
    public void ReSet()...
    public string Get(string folder)...
    public string[] GetFolders()...
    public int GetProblemCount(string folder)...
}
```

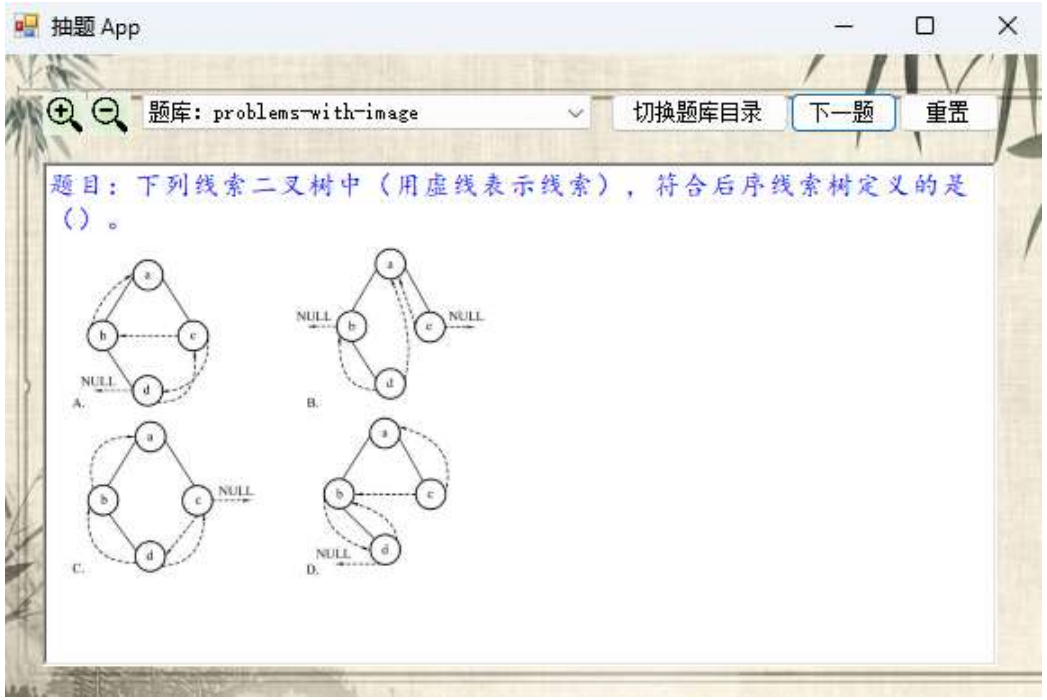
图：DataLoader类

题库读取默认路径为./data，允许用户修改题库路径。
读取的题库会被DataLoader类组织，并对外提供调用接口



图：选择题库

文本/图片显示

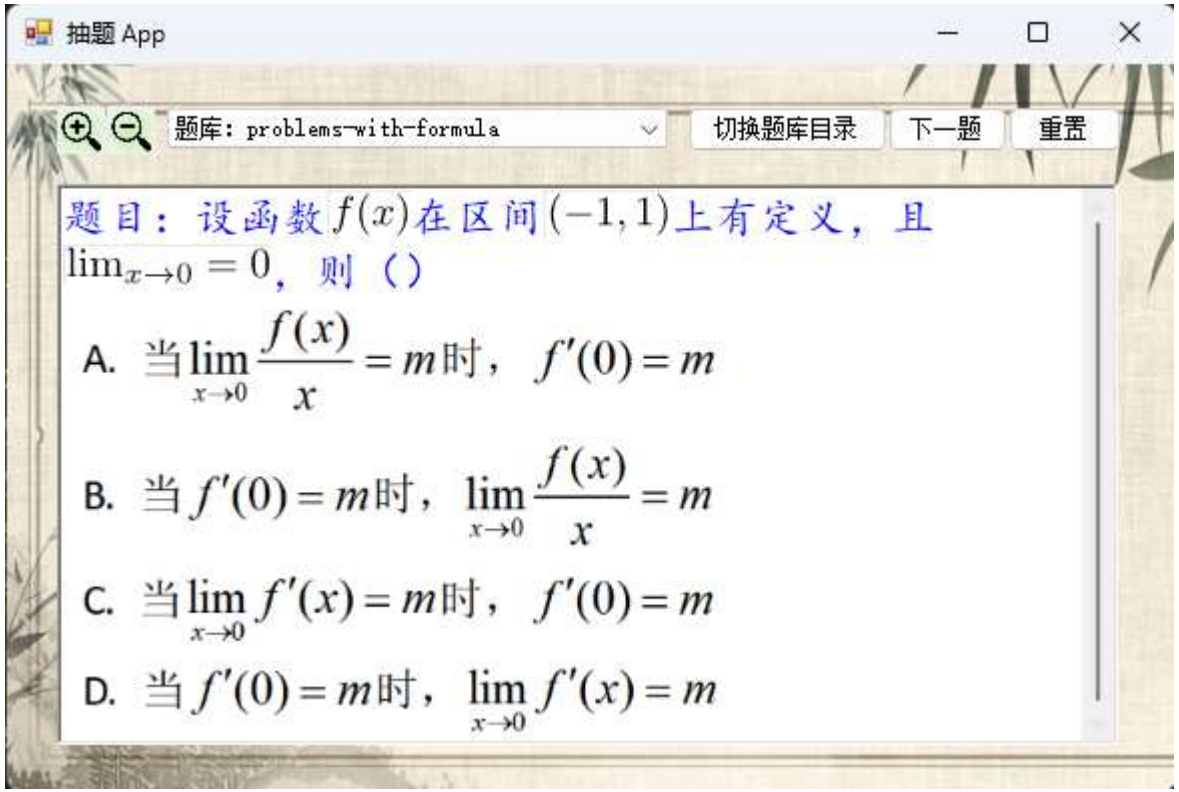


图：题目被显示到RichTextBox控件

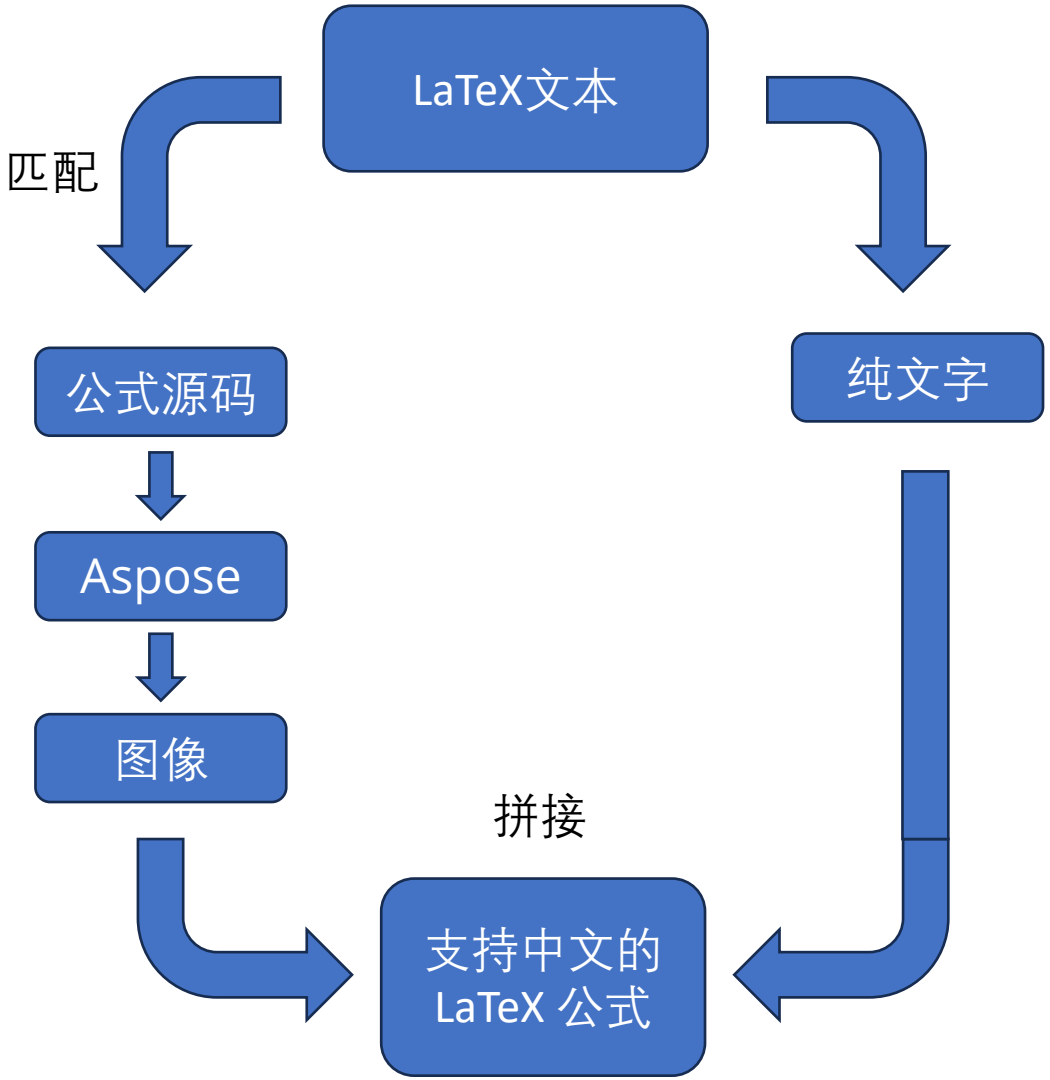
文本显示：使用RichTextBox的AppendText方法

图片显示：若题目中包含图片，则根据相对路径读取，使用 Clipboard.SetDataObject() 方法将图片数据复制到剪贴板上，并Paste到RichTextBox上。

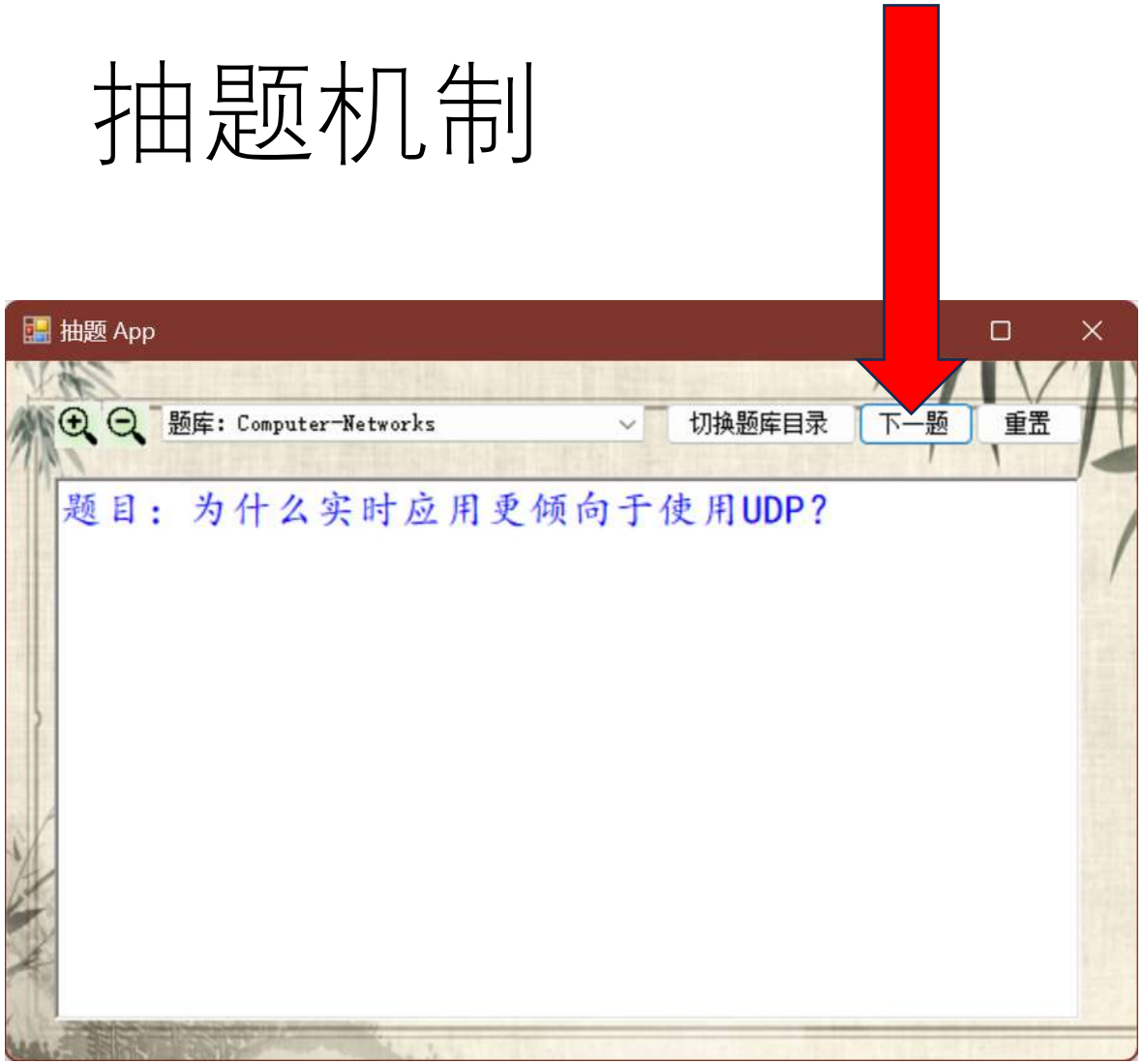
LaTeX 公式显示

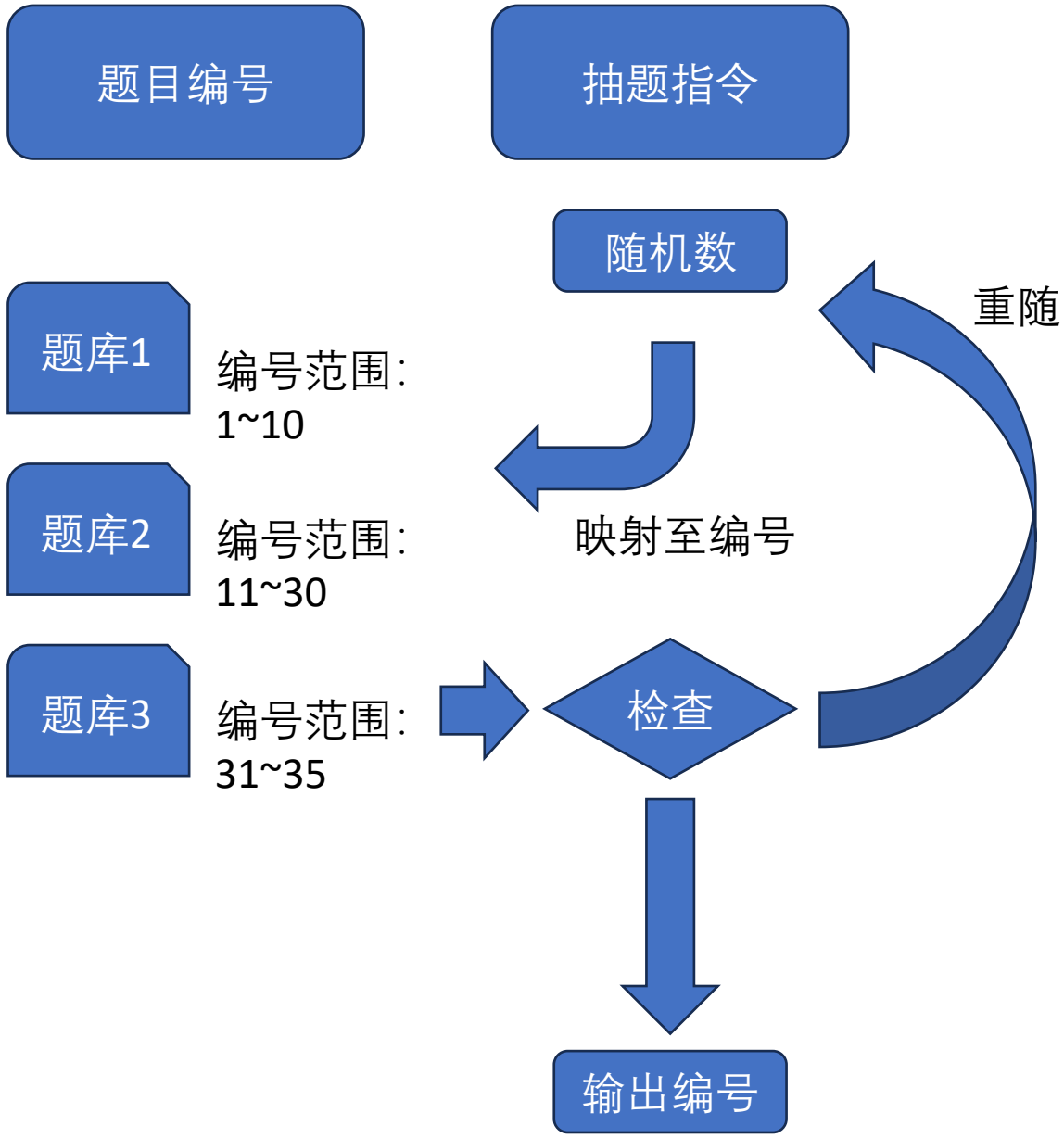


图：穿插在文本里的LaTeX公式渲染



抽题机制


抽题 App
题库：Computer-Networks
切换题库目录
下一题
重置
题目：为什么实时应用更倾向于使用UDP?


题目编号
抽题指令
随机数
重随
题库1
编号范围:
1~10
题库2
编号范围:
11~30
题库3
编号范围:
31~35
映射至编号
检查
输出编号

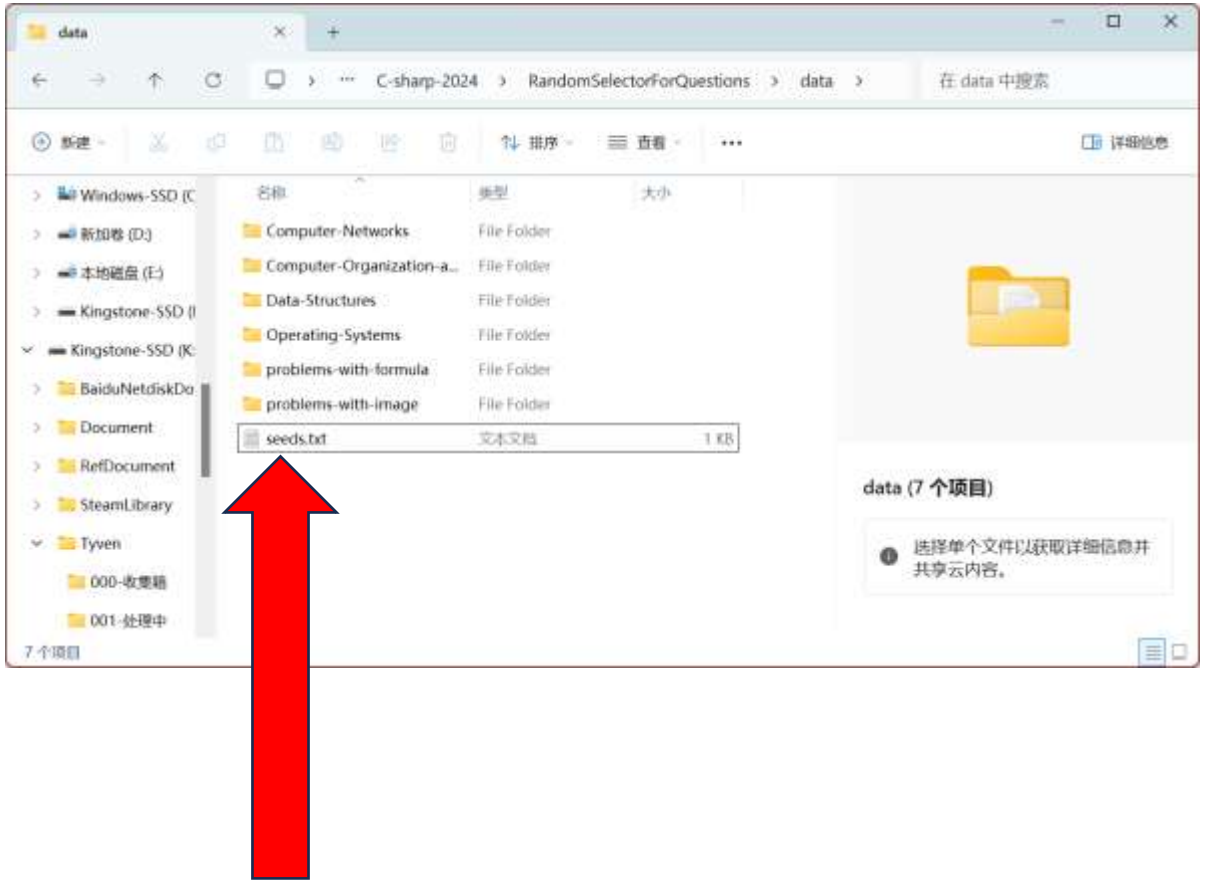
状态维护-随机数发生器

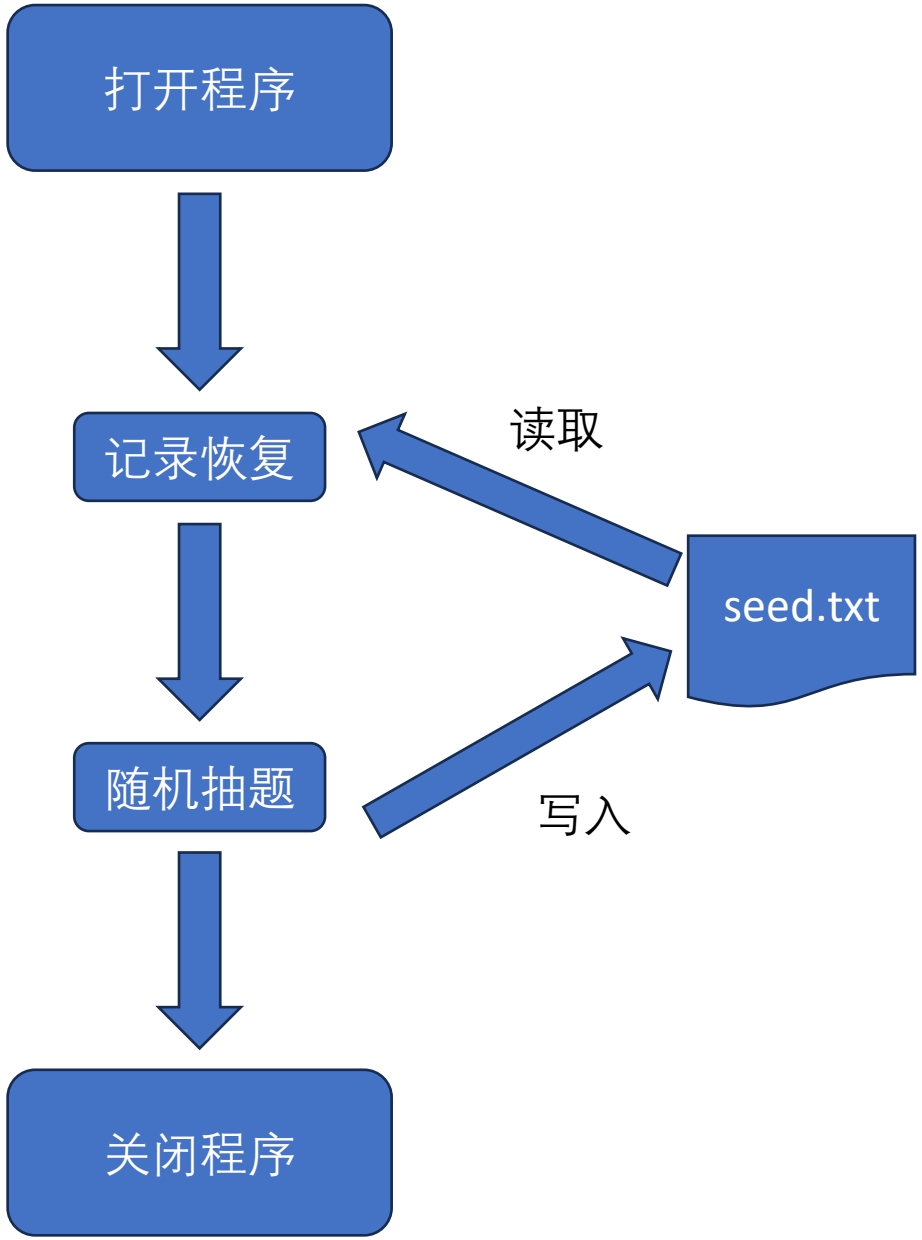
Kobayashi提出的满周期2^{31}的混合同余发生器：

$$x_n = (314159269x_{n-1} + 453806245)(\text{mod } 2^{31})$$

```
namespace RandomSelectorForQuestions
{
    9 个引用
    internal class RandomNumberGenerator
    {
        private static RandomNumberGenerator instance;
        1 个引用
        private RandomNumberGenerator() { }
        private long seed, tot, x;
        1 个引用
        static public void init() { instance = new RandomNumberGenerator(); }
        2 个引用
        static public void setSeed(long seed) { instance.seed = seed; reset(); }
        1 个引用
        static public void reset() { instance.x = instance.seed; instance.tot = 0; }
        0 个引用
        static public long getSeed() { return instance.tot; }
        3 个引用
        static public int getNext() {
            instance.x = ( instance.x * 314159269 + 453806245 ) % 2147483648;
            instance.tot++;
            return (int)instance.x;
        }
    }
}
```

状态维护-状态保存


data
C-sharp-2024
RandomSelectorForQuestions
data
在 data 中搜索
新建
排序
查看
详细信息
Windows-SSD (C
新加卷 (D:)
本地磁盘 (E:)
Kingstone-SSD (I
Kingstone-SSD (K:
BaiduNetdiskDo
Document
RefDocument
SteamLibrary
Tyven
000-收集箱
001-处理中
7 个项目
名称
类型
大小
Computer-Networks
Computer-Organization-a...
Data-Structures
Operating-Systems
problems-with-formula
problems-with-image
seeds.txt
文本文档
1 KB
File Folder
data (7 个项目)
选择单个文件以获取详细信息并共享云内容。


打开程序
记录恢复
随机抽题
关闭程序
读取
写入
seed.txt

状态维护-确定性抽题过程

题库1

编号范围:
1~10

题库2

编号范围:
11~30

题库3

编号范围:
31~35

状态维护-确定性抽题过程

生成全排列

31, 3, 13, 2, 33, 17, 11, 5, 4, 6…

题库1

编号范围:
1~10

题库2

编号范围:
11~30

题库3

编号范围:
31~35

状态维护-确定性抽题过程

分解子序列

31, 3, 13, 2, 33, 17, 11, 5, 4, 6…

题库1

编号范围:
1~10

, 3, , 2, , , , 5, 4, 6…

题库2

编号范围:
11~30

, , 13, , , 17, 11, , , , …

题库3

编号范围:
31~35

31, , , , 33, , , , , , …

状态维护-确定性抽题过程

题库2抽题1

31, 3, 13, 2, 33, 17, 11, 5, 4, 6…

题库1

编号范围:
1~10

, 3, , 2, , , , 5, 4, 6…

题库2

编号范围:
11~30

, , 13, , , 17, 11, , , , …

题库3

编号范围:
31~35

31, , , , 33, , , , , , …

状态维护-确定性抽题过程

题库2抽题2

31, 3, 13, 2, 33, 17, 11, 5, 4, 6…

题库1
编号范围:
1~10

, 3, , 2, , , , 5, 4, 6…

题库2
编号范围:
11~30

, , 13, , , 17, 11, , , , …

题库3
编号范围:
31~35

31, , , , 33, , , , , , …

状态维护-确定性抽题过程

题库3抽题1

31, 3, 13, 2, 33, 17, 11, 5, 4, 6…

题库1

编号范围:
1~10

, 3, , 2, , , , 5, 4, 6…

题库2

编号范围:
11~30

, , 13, , , 17, 11, , , , …

题库3

编号范围:
31~35

31, , , , 33, , , , , , …

状态维护-确定性抽题过程

全题库抽题1

31, 3, 13, 2, 33, 17, 11, 5, 4, 6…

题库1

编号范围:
1~10

, 3, , 2, , , , 5, 4, 6…

题库2

编号范围:
11~30

, , 13, , , 17, 11, , , , …

题库3

编号范围:
31~35

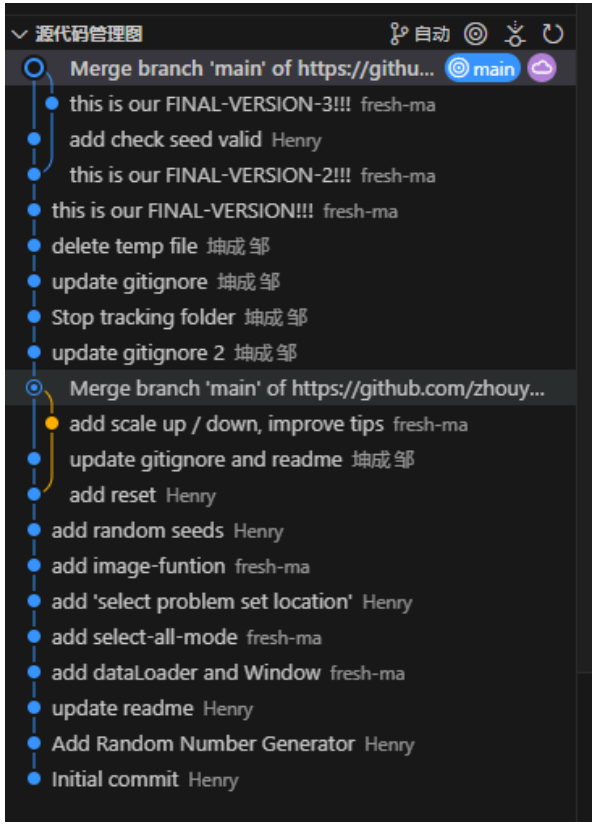
31, , , , 33, , , , , , …

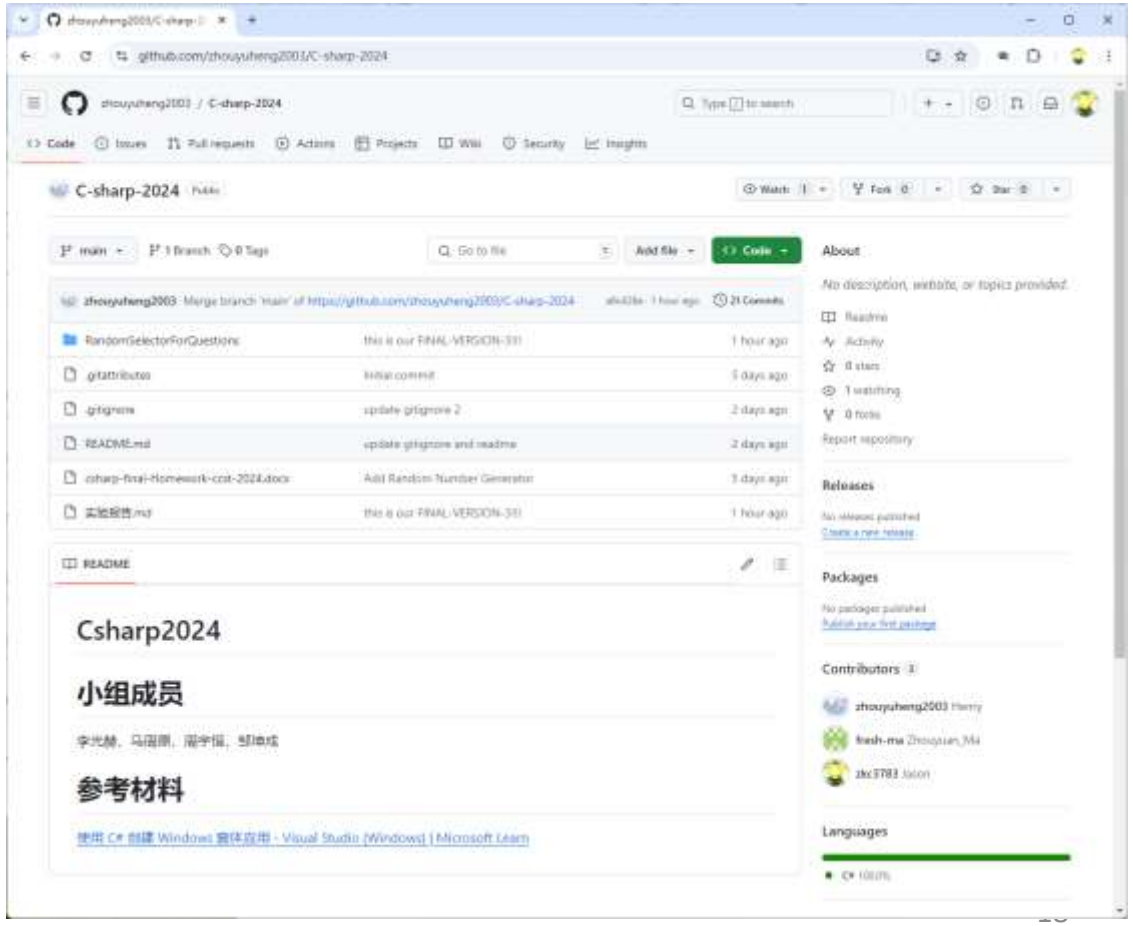
代码管理

本项目位于Github（项目地址：https://github.com/zhouyuheng2003/C-sharp-2024），

使用Visual Studio配合Git进行多人协作和代码同步，大幅提升了开发效率

将代码，测试数据，资源进行同步。忽略无关临时文件





系统演示

组员分工

贡献	李光赫 21210713	邹坤成 19210215	周宇恒 55210916	马周原 21210613
UI设计	✔	✔	✔	✔
抽题机制		✔	✔	
图片/公式	✔			✔
题库设计/加载				✔
状态维护			✔	
报告撰写	✔	✔	✔	✔
贡献百分比	**25%**	**25%**	**25%**	**25%**

THANK YOU

吉林大学计算机学院

马周原