**数据结构课程实践1-迷宫设计文档**

1. **需求分析**

* **总需求：输入或生成一个由二维数据表示的迷宫，实现自动寻路并且输出对应的路径**

**具体功能如下**

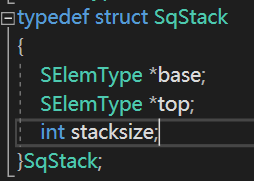
1. 可以自动随机生成迷宫或者选择由用户以TXT文本的形式导入迷宫
2. 实现走迷宫路径的动态显示，实时显示走迷宫的状态，让走迷宫可视化
3. 当迷宫存在多路径时可以输出全部路径（路径较多故不加动态显示）
4. 当存在多条路径时，可以输出同等级的最优解
5. **具体需求实现设计**
6. 实现随机生成迷宫以及文件导入
7. 起点规定为（1，0），终点规定为（M-2，N-1）
8. 在运行时输入迷宫的长宽后显示选择界面，用户可以自行选择自行生成迷宫或者导入根目录下的input.txt文件
9. 实现走迷宫的动态显示，让走迷宫可视化
10. 在用户输入迷宫后可以选择动态显示1条路模式进入动态显示功能
11. 用实心方块■表示墙 空心方块□表示路 X表示已知的不通的路
12. 用箭头表示实时走迷宫的方向
13. 最后再输出具体走过的路径供用户参考

（3&4） 当迷宫存在多条路径时输出全部路径或最短路径

1. 用户可以在输入迷宫后可以选择输出多条路径或者输出最短路径
2. 选择多条路径后 显示所有的走法
3. 选择最短路径后 显示所有最短的走法
4. **概要设计**
5. 数据结构设计

**这次采用的基本数据结构是以顺序表为基础的堆栈 实现了以下功能**

1. 基本的结构体设计



1. 获取栈顶元素

Status GetTop(SqStack \*s,SElemType \*)

1. 构造一个空栈

Status InitStack(SqStack \*s)

1. 元素出栈

Status Pop(SqStack \*s,SElemType \*)

1. 元素入栈

Status Push(SqStack (\*s),SElemType e)

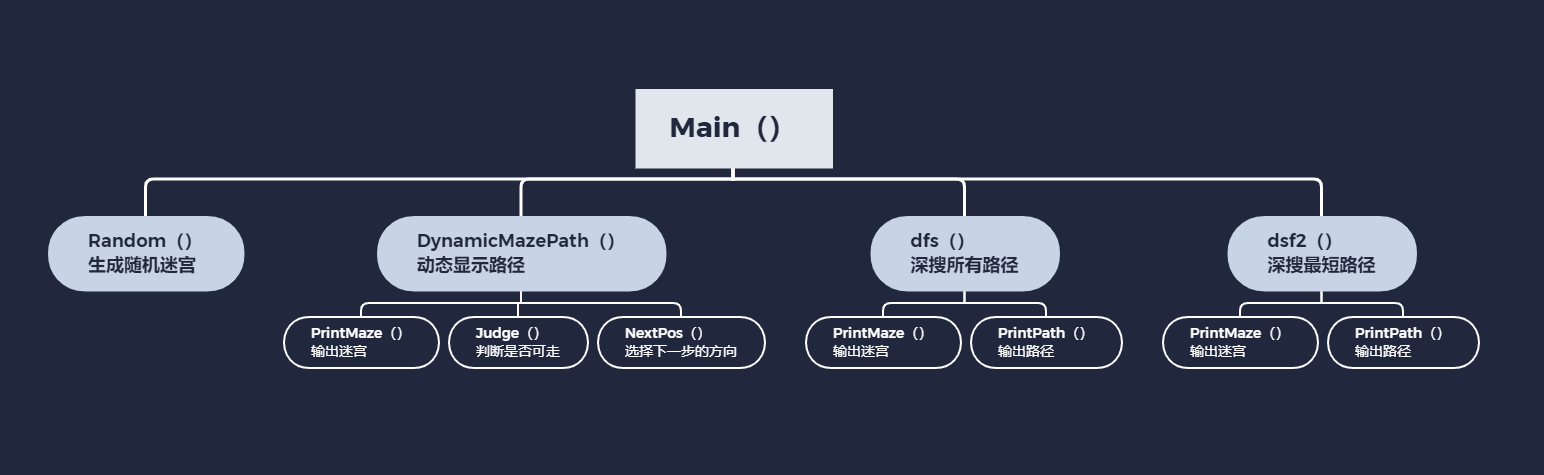
1. 判断栈是否为空

Status StackEmpty(SqStack s)

1. 清空栈中所有元素

void StackClear(SqStack \*s)

1. 模块设计
2. 模块结构图



1. 函数说明

项目中包含了以下函数

1. void Random()//随机生成迷宫的函数
2. Status Judge(PosType e) //判断是否可走
3. PosType NextPos(PosType e,int dir) //下一步
4. void PrintPath(SqStack \*s)//输出路径
5. void PrintMaze()//打印迷宫
6. void dfs(PosType start,PosType end,SqStack \*s) //深度搜索所有路径
7. void dfs2(PosType start,PosType end,SqStack \*s,int len)

//深度搜索最短路径

1. Status DynamicMazePath(PosType start,PosType end,SqStack \*s)

//动态显示路径

1. **详细设计**
2. **程序流程**

