

**数据结构实验报告**

**实验报告题目：** **顺序表基本算法的实现**

**学生姓名**：  **刘凌杰**

**专 业： 软件工程**

**学 号**： 201720180411

**指导教师： 邹 国 华**

**2018年 10 月 20 日**

1. **实验内容：**

* **判断栈是否满**
* **栈的内存申请（构造函数）**
* **栈的内存释放（析构函数）**
* **判断栈是否空**
* **获取栈顶元素**
* **出栈**
* **入栈**

**二. 实验算法实现：**

**类：**

typedef int Status;

typedef int DataType;

//栈类

class Stack

{

private:

int top;

DataType \*data;

public:

Stack()

{

this->data=(DataType\*)malloc(SIZE\*sizeof(DataType));

if(!this->data)

{

cout<<"内存申请失败！"<<endl;

return ;

}

this->top=-1;

}

Stack(int n)

{

this->data=(DataType\*)malloc(n\*sizeof(DataType));

if(!this->data)

{

cout<<"内存申请失败！"<<endl;

return ;

}

this->top=-1;

}

~Stack()

{

free(this->data);

this->data=NULL;

this->top=-1;

}

bool emptyStack(); //判断栈是否为空

bool fullStack(); //判断栈是否满

DataType topStack(); //获取栈顶元素

Status inputStack(DataType e); //入栈

Status popStack(DataType e); //出栈

};

**1.判断栈是否满**

//判断栈是否满

bool Stack::fullStack()

{

return this->top==SIZE-1;

}

**2.判断栈是否空**

//判断栈是否为空

bool Stack::emptyStack()

{

return this->top==-1;

}

**3.获取栈顶元素**

//获取栈顶元素

DataType Stack::topStack()

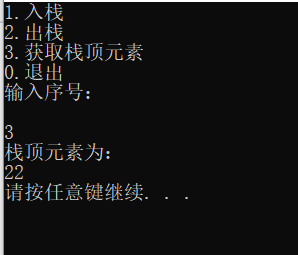
{

if(this->emptyStack())

return ERROR;

return this->data[this->top];

}

****

**4.出栈**

//入栈

Status Stack::inputStack(DataType e)

{

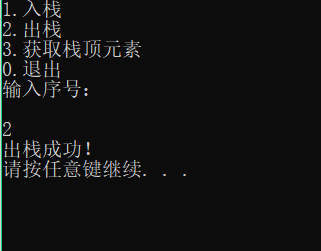
if(this->fullStack())

return ERROR;

this->data[++this->top]=e;

return OK;

}

****

**5.入栈**

//出栈

Status Stack::popStack(DataType e)

{

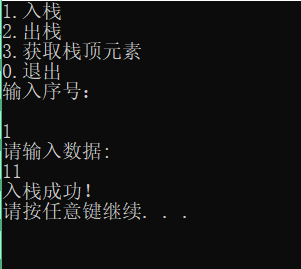
if(this->emptyStack())

return ERROR;

e=this->data[this->top--];

return OK;

}

****

**6.main函数**

int main()

{

Stack s;

DataType data;

char c;

while(1)

{

cout<<"1.入栈"<<endl;

cout<<"2.出栈"<<endl;

cout<<"3.获取栈顶元素"<<endl;

cout<<"0.退出"<<endl;

cout<<"输入序号："<<endl<<endl;

cin>>c;

switch(c)

{

case'1':

cout<<"请输入数据:"<<endl;

cin>>data;

if(s.inputStack(data)==0)

cout<<"入栈成功！"<<endl;

else

cout<<"入栈失败！"<<endl;

system("pause");

system("cls");

break;

case'2':

if(s.popStack(data)==0)

cout<<"出栈成功！"<<endl;

else

cout<<"出栈失败！"<<endl;

system("pause");

system("cls");

break;

case'3':

cout<<"栈顶元素为："<<endl;

cout<<s.topStack()<<endl;

system("pause");

system("cls");

break;

case'0':

exit(0);

default:

cout<<"输入序号错误！"<<endl;

system("cls");

}

}

return 0;

}

**三．实验总结：**

经过一段时间的学习，栈算法的实现基本掌握了，就是代码打得还不太熟练，还有待提高，有些功能不太熟要问同学或查资料。以后在这方面我会多花点时间，让自己更加熟练。