**顺序栈的操作**

1. 定义
   1. 只能在栈顶进行操作的特殊线性表
   2. 因此，我们话不多说，直接进行coding
2. 代码

#include <iostream>

#include <stdio.h>

#include <iomanip>

#include <string>

#include <assert.h>

#include <time.h>

#include <string>

template <typename T>

class SQStack{

T \*data, \*top\_; //\*top\_ 代表的是长度，我们用top\_ - data, 我们data指向的是栈底元素

int capacity; // 容量

public:

//初始化函数

SQStack(int cap = 5){

data = new T[cap];

if(!data) throw "space error";

top\_ = data;

capacity = cap;

}

//入栈函数

bool push(T e){

//注意下标是否越界

if(top\_ - data >= capacity)

relloc();

\*top\_++ = e; //完赋值，地址++就行了

}

//出栈函数

bool pop(){

--top\_;

}

//获取长度

int getSize(){

return top\_ - data;

}

//下标函数获取值,实现两个

T operator[](int i)const{

return data[i];

}

T& operator[](int i){

return data[i];

}

//更新值操作

bool setVal(int i, T e){

data[i-1] = e;

}

private:

//增加内存函数

void relloc(){

T \*p = new T[2\*capacity]; // 增加内存空间

for(int i = 0; i < capacity; i++){

p[i] = data[i];

delete[] data;

data = p;

}

}

};

template <typename T>

void printStack(SQStack<T> &L){

T e;

for(int i = 0; i < L.getSize(); i++){

e = L[i];

std::cout<<e<<" ";

}

std::cout<<std::endl;

}

using std::string;

struct student{

string name;

int age;

student(string n = "no", int a = 0):name(n),age(a){}

};

//进行一下输出的重载

std::ostream& operator<<(std::ostream &os, student &s){

os<<s.name<<":"<<s.age;

}

int main()

{

SQStack<char> sqstack;

sqstack.push('A');

sqstack.push('B');

sqstack.push('C');

printStack(sqstack);

sqstack.setVal(2, 'D');

printStack(sqstack);

sqstack.pop();

printStack(sqstack);

std::cout<<"----------------------"<<std::endl;

SQStack<student> students;

students.push(student("weilin", 23));

printStack(students);

}