

Постановка задачи

Создать класс для объекта стек. Стек хранит целые числа. Имеет характеристики: наименование (строка, не более 10 символов) и размер (целое). Размер стека больше или равно 1. Функционал стека:

- добавить элемент и вернуть признак успеха (логическое);
- извлечь элемент и вернуть признак успеха (логическое);
- получить имя стека (строка); - получить размер стека (целое);
- получить текущее количество элементов в стеке (целое).

В классе определить параметризованный конструктор, которому передается имя стека и размер. При переполнении стека очередной элемент не добавлять и определяется соответствующий признак успеха.

В основной программе реализовать алгоритм:

1. Ввести имя и размер для первого стека.
2. Создать объект первого стека.
3. Ввести имя и размер для второго стека.
4. Создать объект второго стека.
5. В цикле:
 - 5.1. Считывать очередное значение элемента.
 - 5.2. Добавлять элемент в первый стек, при переполнении завершить цикл.
 - 5.3. Добавлять элемент во второй стек, при переполнении завершить цикл.
6. Построчно вывести содержимое стеков.

Описание входных данных

Первая строка: «имя стека 1»«размер стека»

Вторая строка: «имя стека 2»«размер стека»

Третья строка: Последовательность целых чисел, разделенных пробелами, в количестве не менее чем размер одного из стеков + 1.

Описание выходных данных

Первая строка:

«имя стека 1»«размер»

Вторая строка:

«имя стека 2»«размер»

Третья строка:

«имя стека 1»«имя стека 2» Каждое имя стека в

третьей строке занимает поле длины 15 позиции и прижата к левому краю.

Четвертая строка и далее построчно, вывести все элементы стеков:

«значение элемента стека 1»«значение элемента стека 2» Вывод значений элементов стеков производится последовательным извлечением.

Каждое значение занимает поле из 15 позиции и прижата к правому краю.

Метод решения

библиотека `iostream` библиотека `string` собственный класс `Sk`,

реализующий такую структуру как стек Атрибуты:

- `int max_size` - максимально возможный размер стека
- `int size` - нынешний размер стека
- `int * m` - указатель на массив для хранения элементов стека
- `string name` - имя стека

Методы:

- `Sk(string name, int size)` - конструктор, принимающий в аргументах значение имени стека и её максимальный размер
- `~Sk()` - деструктор стека для очистки памяти
- `bool add(int num)` - добавляет элемент `num` в стек, если это возможно
- `bool pop()` - удаляет последний элемент стека, если это возможно
- `int get(int index)` - возвращает элемент стека с данным индексом
- `string get_name()` - возвращает имя стека
- `int get_max_size()` - возвращает максимальный размер стека доступного для данного объекта

- `int get_size()` - возвращает нынешний размер стека

Описание алгоритма

Функция: `main`

Функционал: Функция, из которой программа начинает свою работу

Параметры: отсутствуют

Возвращаемое значение: `int` - код работы

№	Предикат	Действия	№ перехода	Комментарий
1		Вызов функции <code>program</code>	∅	

Функция: `program`

Функционал: Основной код программы тут

Параметры: отсутствуют

Возвращаемое значение: отсутствует

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация переменной <code>pp</code> строкового типа	2	
2		Инициализация переменной <code>n</code> целочисленного типа	3	
3		Считывание переменных <code>pp</code> и <code>n</code>	4	
4		Инициализация объекта <code>s1</code> класса <code>Sk</code> с аргументами <code>pp</code> и <code>n</code>	5	
5		Считывание переменных <code>pp</code> и <code>n</code>	6	
6		Инициализация объекта <code>s2</code> класса <code>Sk</code> с аргументами <code>pp</code> и <code>n</code>	7	
7	Всегда	Считывание <code>n</code>	8	
	Никогда		9	

8	Метод add объекта s1 с аргументом n возвращает ложь		10	
	Метод add объекта s1 с аргументом n возвращает правду		9	
9	Метод add объекта s2 с аргументом n возвращает ложь		10	
	Метод add объекта s2 с аргументом n возвращает правду		7	
10		Вывод имени очереди s1 и её размера	11	
11		Вывод имени очереди s2 и её размера	12	
12		Вывод имен очередей s1 и s2 по левому краю	13	
13	Сумма размеров стеков не равна нулю	Вывод последнего элемента каждого из стеков	14	
	Сумма размеров стеков равна нулю		∅	
14		Удаление последних элементов стеков методом pop	15	
15	Сумма размеров стеков не равна нулю	Вывод переноса строки	13	
	Сумма размеров стеков равна нулю		13	

Класс объекта: Sk

Модификатор доступа: public

Метод: Sk

Функционал: Конструктор класса

Параметры: string name - имя стека, int size - максимальный размер стека

Возвращаемое значение: отсутствует

№	Предикат	Действия	№ перехода	Комментарий
---	----------	----------	------------	-------------

1		Присваивает закрытому свойству name значение аргумента метода name	2	
2		Присваивает закрытому свойству max_size значение аргумента метода size	3	
3		Инициализация массива закрытого свойства размера max_size	4	
4		Присвоение текущему размеру массива значения 0	∅	

Класс объекта: Sk

Модификатор доступа: public

Метод: ~Sk

Функционал: Деструктор класса

Параметры: отсутствуют

Возвращаемое значение: отсутствует

№	Предикат	Действия	№ перехода	Комментарий
1		Очистка памяти, выделенной для закрытого свойства m	∅	

Класс объекта: Sk

Модификатор доступа: public

Метод: add

Функционал: Добавляет в стек значение, если это возможно

Параметры: int num - добавляемое значение

Возвращаемое значение: bool - удалось ли добавить

№	Предикат	Действия	№ перехода	Комментарий
1	Закрытое свойство size равно закрытому свойству max_size	Вернуть false	∅	
	Закрытое свойство size не равно закрытому свойству max_size	Присвоить элементу массива закрытого свойства m по индексу size значение num	2	

2		Увеличить закрытое свойство size на один, вернуть true	∅	
---	--	--	---	--

Класс объекта: Sk

Модификатор доступа: public

Метод: pop

Функционал: Удаляет верхний элемент из стека, если это возможно

Параметры: отсутствуют

Возвращаемое значение: отсутствует

№	Предикат	Действия	№ перехода	Комментарий
1	Закрытое свойство size равно нулю	Вернуть false	∅	
	Закрытое свойство size не равно нулю	Уменьшить это свойство на один и вернуть true	∅	

Класс объекта: Sk

Модификатор доступа: public

Метод: get

Функционал: Вернет значение элемента по данному индексу

Параметры: int index - индекс элемента, который нужен

Возвращаемое значение: int - элемент по индексу index

№	Предикат	Действия	№ перехода	Комментарий
1		Вернуть элемент массива закрытого свойства m с индексом index	∅	

Класс объекта: Sk

Модификатор доступа: public

Метод: get_name

Функционал: Возвращает имя стека

Параметры: отсутствуют

Возвращаемое значение: string - имя стека

№	Предикат	Действия	№ перехода	Комментарий
1		Вернуть значение закрытого свойства name	∅	

Класс объекта: Sk

Модификатор доступа: public

Метод: get_max_size

Функционал: Возвращает значение максимального размера стека
Параметры: отсутствуют

Возвращаемое значение: int - значение максимально возможного размера стека у данного объекта

№	Предикат	Действия	№ перехода	Комментарий
1		Вернуть значение закрытого свойства max_size	∅	

Класс объекта: Sk

Модификатор доступа: public

Метод: get_size

Функционал: Возвращает размер стека

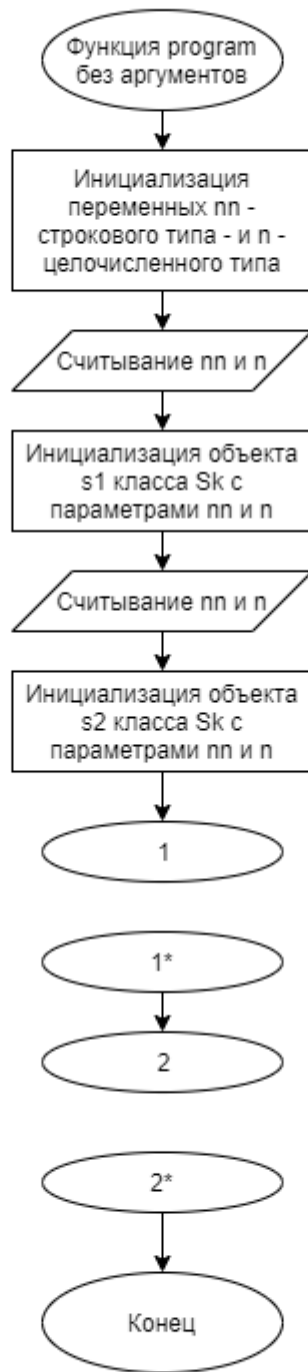
Параметры: отсутствуют

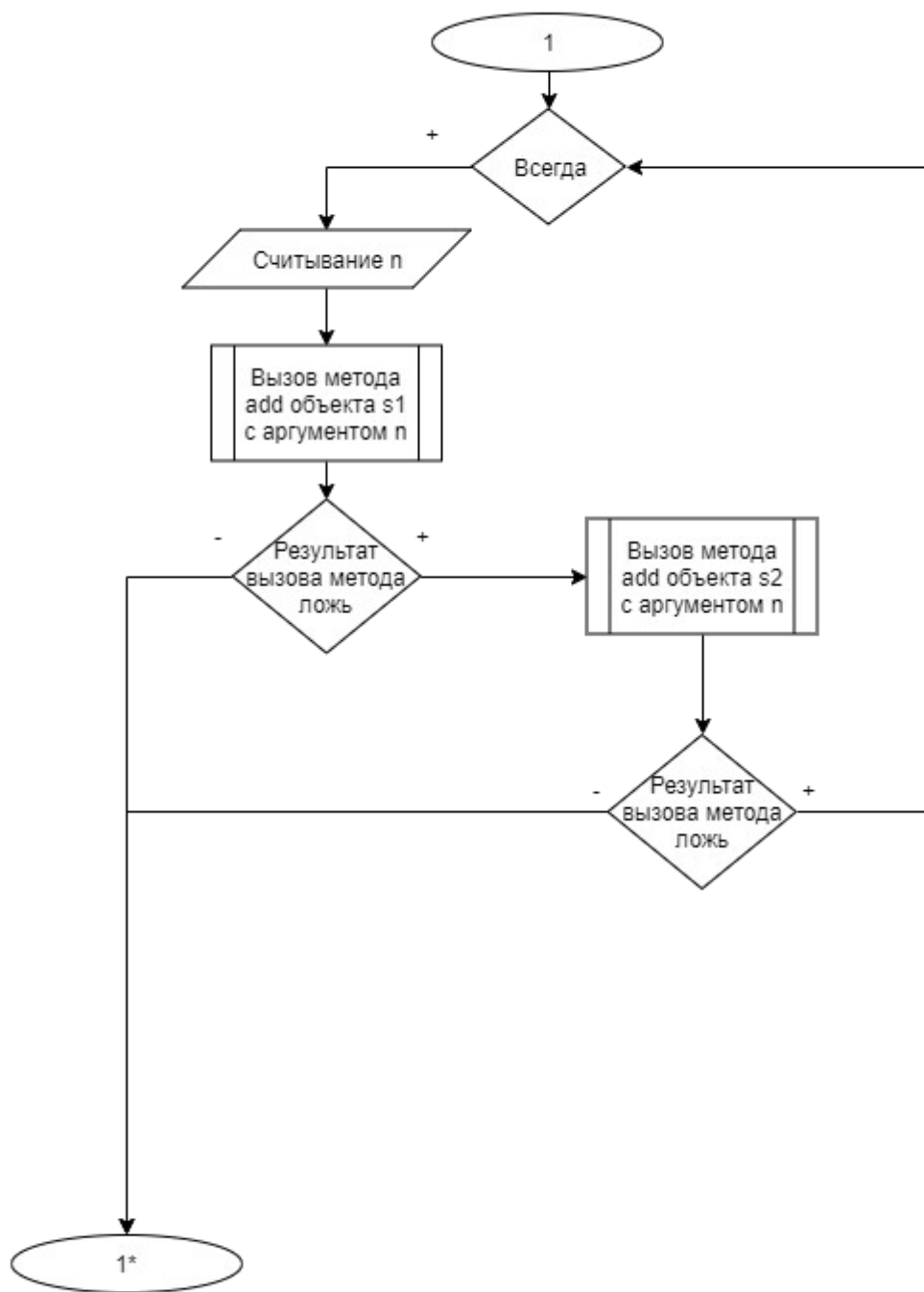
Возвращаемое значение: int - значение текущего размера стека у данного объекта

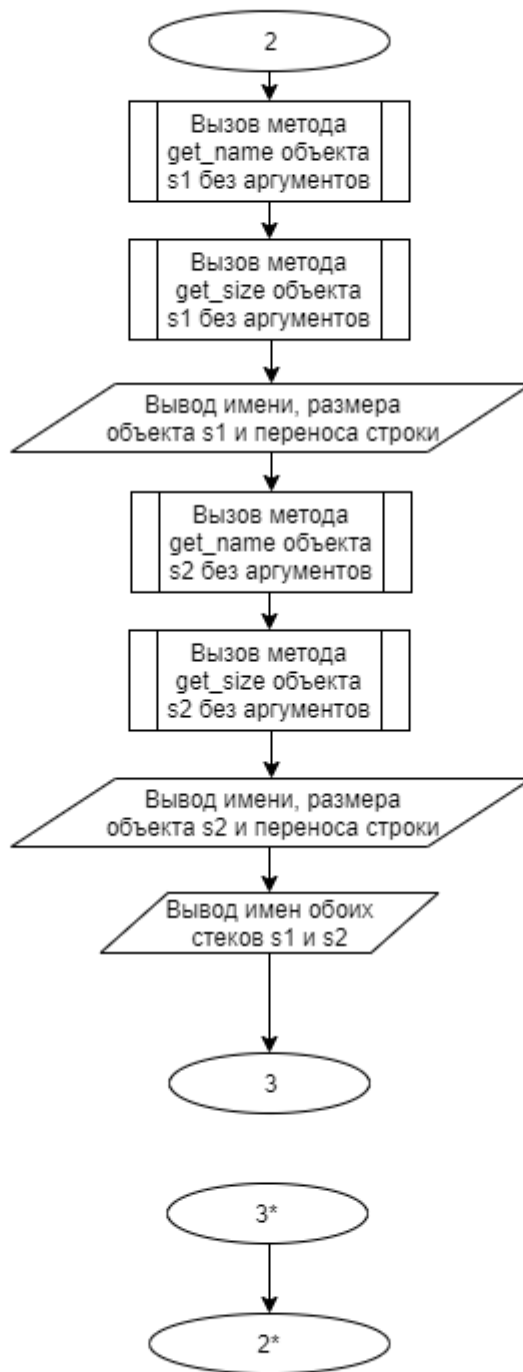
№	Предикат	Действия	№ перехода	Комментарий
1		Вернуть значение закрытого свойства size	∅	

Блок-схема алгоритма





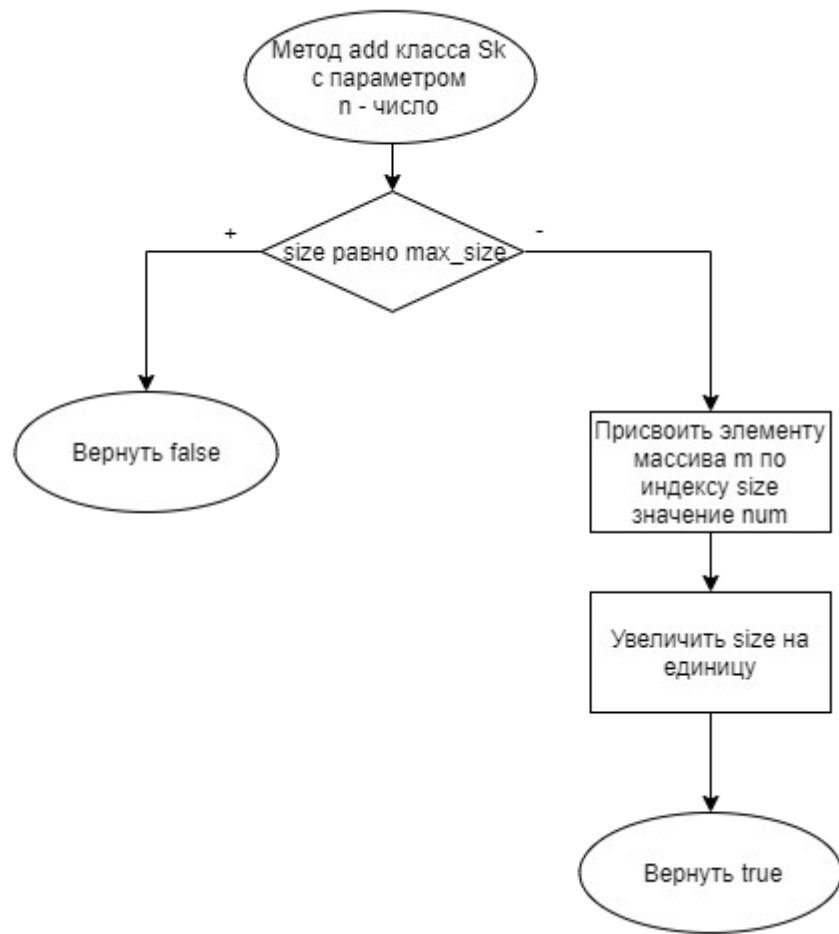


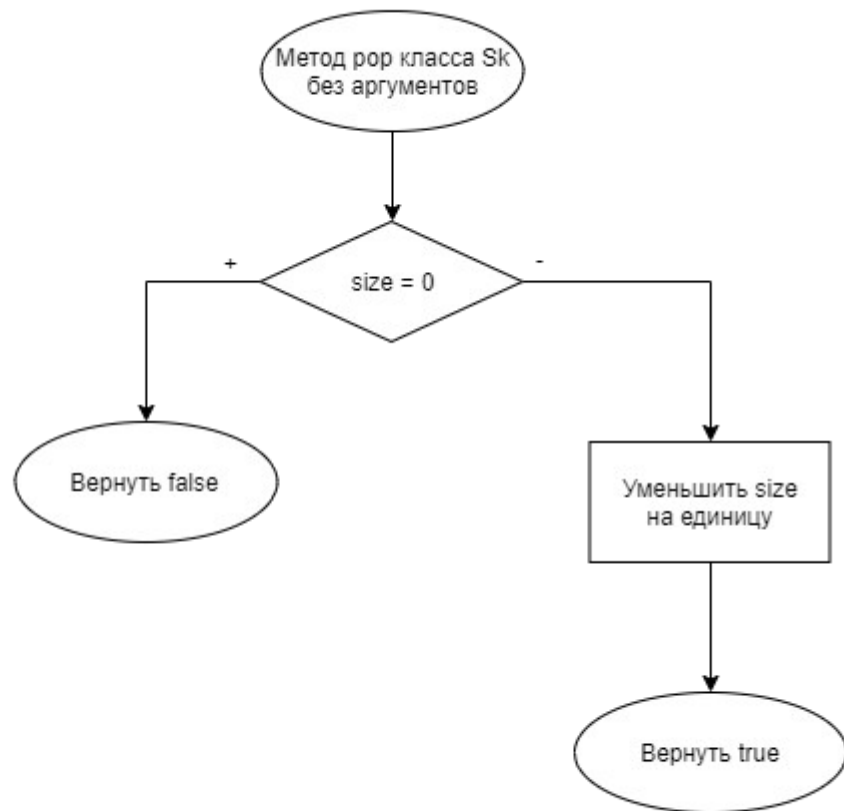












Метод get класса Sk
с параметром index



Вернуть элемент
массива m по индексу
index







Код программы Файл main.cpp

```
#include <iostream>
```

```

#include "program.h"

using namespace std;

int main()
{
    program();
return(0); }

```

Файл program.cpp

```

#include <iostream>
#include "sk.h"

using namespace std;

void program()
{
    string nn;
int n;
    // Создание первого стека
cin >> nn >> n;      Sk s1(nn,
n);
    // Создание второго стека
cin >> nn >> n;      Sk s2(nn,
n);

    // Считывание значений стеков
while (true) {

cin >> n;              if
(!s1.add(n)){
break;
                }
                if (!s2.add(n)){
break;
                }
    }

    // Вывод результата работы программы
cout << s1.get_name() << " " << s1.get_max_size() << "\n";
cout << s2.get_name() << " " << s2.get_max_size() << "\n";      printf("%-
15s%-15s\n", s1.get_name().c_str(), s2.get_name().c_str());
    while (s1.get_size() + s2.get_size()) {
        if (s1.get_size())
            printf("%15d", s1.get(s1.get_size() - 1));
        else
            printf("                ");
        if (s2.get_size())
            printf("%15d", s2.get(s2.get_size() - 1));
    }
}

```

```

        s1.pop();
s2.pop();
        if (s1.get_size() + s2.get_size())
cout << "\n";
    }

return; }

```

Файл program.h

```

#ifndef PROGRAMH
#define PROGRAMH
void program();
#endif

```

Файл sk.cpp

```

#include <string>
#include "sk.h"
#include "program.h"

Sk::Sk(string name, int size){
this->name = name;          this->max_size
= size;          this->m = new int[this-
>max_size];
        this->size = 0;
}
Sk::~Sk(){
        delete this->m;
}
bool Sk::add(int num){
        if (this->size == this->max_size) {
return false;
        }
        this->m[this->size] = num;
        this->size++;
        return true;
} bool Sk::pop(){          if
(this->size > 0) {
this->size--;
return true;
        }
return false;
}

```

```

} int Sk::get(int index) {
return this->m[index];
} string Sk::get_name() {
return this->name;
} int Sk::get_max_size() {
return this->max_size;
} int Sk::get_size() {
return this->size; }

```

Файл sk.h

```

#ifndef MIREAH
#define MIREAH
#include <string>

using namespace std;

void program();

class Sk{ private:
int max_size;
int * m;
string name;
int size; public:
    Sk(string name, int size);
    ~Sk();          bool
add(int num);     bool
pop();           int get(int
index);          string
get_name();      int
get_max_size(); int
get_size();

};

#endif

```

Тестирование

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
A 2 B 3 1 2 3 4 5 6 7 8 9	A 2 B 3 A B 2 2 1 1	A 2 B 3 A B 2 2 1 1
A 3 B 2 1 2 3 4 5 6 7 8 9	A 3 B 2 A B 3 2 2 1 1	A 3 B 2 A B 3 2 2 1 1

A4B2123456789	A4B2AB32211	A4B2AB32211
---------------	-------------	-------------