

## Постановка задачи

Создать базовый класс, содержащий метод наименованием `in_out` без параметров. Метод организует ввод исходных данных: размерности целочисленного массива и значений элементов целочисленного массива. На базе базового класса создать производный класс 1 содержащий метод целого типа наименованием `func`. Метод от значения первого элемента целочисленного массива последовательно вычитает значения остальных элементов. Полученный результат возвращает.

На базе класса 1 создать производный класс 2 содержащий метод целого типа наименованием `func`. Метод суммирует элементы целочисленного массива. Полученный результат возвращает.

На базе класса 2 создать производный класс 3 содержащий метод наименованием `in_out` без параметров для вывода исходных данных и вычисленных значений.

В основной программе использовать один указатель на объект класса и один объект.

- В основной программе реализовать алгоритм:
1. Объявить указатель на объект класса.
  2. Создать объект оператором `new`.
  3. Ввод исходных данных: размерности целочисленного массива и значений элементов данного массива.
  4. Вычисление значение вычитания от первого элемента массива всех остальных.
  5. Вычисление суммы элементов массива.
  6. Вывод полученных результатов.

## Описание входных данных

**Первая строка:**  
«размерность целочисленного массива»  
Размерность массива натуральное число больше или равно 1 и меньше или равно 100.

**Вторая строка:**  
«последовательность целых чисел»  
Количество целых чисел равно размерности массива и разделены пробелами. Значение каждого числа принадлежит интервалу  $[-100, 100]$ .

## Описание выходных данных

**Первая строка:**  
Array dimension: «размерность целочисленного массива»

**Вторая строка:**  
The original array: «последовательность целых чисел»  
Для вывода каждого числа выделяется поле из 5 позиций.

**Третья строка:**  
Min: «значение вычитания из первого элемента массива всех остальных»

**Четвертая строка:**  
Sum: «значение суммы элементов массив»

## Метод решения

Оператор ввода-вывода

Оператор цикла

Условный оператор

Иначе

Заголовок метода

Заголовок класса

Присвоение

Класс Base

Свойства класса Base:

public:

- `int *arr = 0` - динамический целочисленный массив
- `unsigned int size = 0` - размер массива

Методы класса Base:

public:

- `virtual void in_out()` - Ввод исходных данных: размерности массива и значений его элементов

Класс Class1 : public Base

Методы класса Class1:

public:

- `virtual int func()` - Вычисление вычитания от первого элемента массива всех остальных

Класс Class2 : public Class1

Методы класса Class2:

public:

- `int func()` - Вычисление суммы элементов массива

Класс Class3 : public Class2

Методы класса Class3:

public:

- `void in_out()` - Вывод полученных результатов

## **Описание алгоритма**

Функция: main

Функционал: Основная функция

Параметры: нет

Возвращаемое значение: int, код ошибки

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление указателя obj на объект класса Class3	2	
2		Создание объекта оператором new	3	
3		Ввод исходных данных: размерности и значений элементов массива	4	
4		Вызов метода in_out объекта obj(метод Class3)	∅	Вычисление вычитания от первого элемента массива всех остальных Вычисление суммы элементов массива Вывод результата

Класс объекта: Base

Модификатор доступа: public

Метод: in\_out

Функционал: Ввод исходных данных: размерности и значений элементов массива

Параметры: нет

Возвращаемое значение: void

№	Предикат	Действия	№ перехода	Комментарий
1		Ввод исходных данных: размерности и значений элементов массива циклом за size тактов	∅	

Класс объекта: Class1

Модификатор доступа: public

Метод: func

Функционал: Вычисление вычитания от первого элемента массива всех остальных

Параметры: нет

Возвращаемое значение: int, разность между 1ым элемнтом и всеми остальными элементами массива

№	Предикат	Действия	№ перехода	Комментарий
1		Вычисление и возвращение вычитания от первого элемента массива всех остальных циклом за size-1 тактов	∅	

Класс объекта: Class2

Модификатор доступа: public

Метод: func

Функционал: Вычисление суммы элементов массива

Параметры: нет

Возвращаемое значение: int, сумма всех элементов массива

№	Предикат	Действия	№ перехода	Комментарий
1		Получение и возвращение суммы элементов массива за size тактов	∅	

Класс объекта: Class3

Модификатор доступа: public

Метод: in\_out

Функционал: Вывод полученных результатов

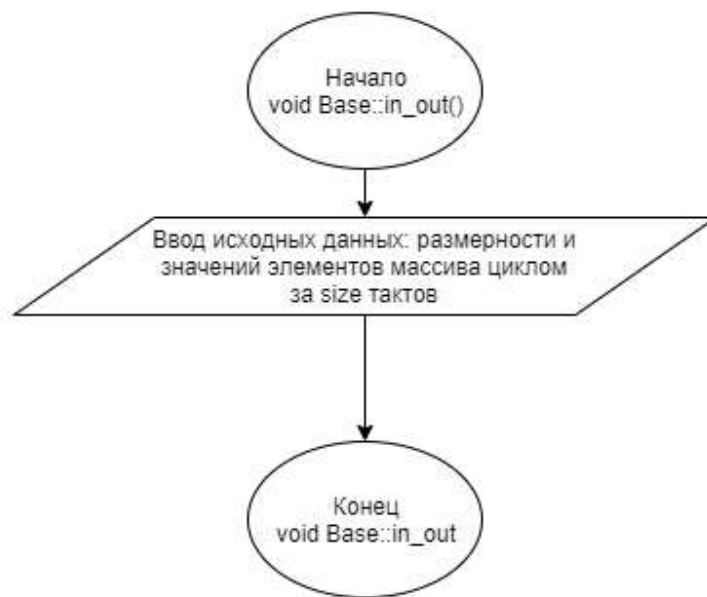
Параметры: нет

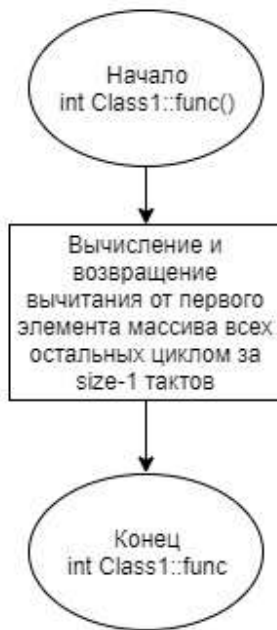
Возвращаемое значение: void

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод размерности массива size согласно форматированию	2	
2		Вывод исходного массива arr согласно форматированию за size тактов согласно форматированию	3	
3		Вызов метода func класса Class1 этого объекта и вывод результата как разницы от первого элемента массива и всех остальных элементов согласно форматированию	4	
4		Вызов метода func класса Class2 и вывод результата как сумма всех элементов массива согласно форматированию	∅	

## Блок-схема алгоритма













**Код программы**

## Файл Base.cpp

```
#include "Base.h"

void Base::in_out() {
    cin >> size;
    if (size <= 0) return;
    delete arr;
    arr = new int[size];
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }
}
```

## Файл Base.h

```
#ifndef BASE_H
#define BASE_H

#include <iostream>
#include <string>

using namespace std;

class Base {
public:
    int *arr = 0;
    unsigned int size = 0;
public:
    virtual void in_out();
};

#endif
```

## Файл Class1.cpp

```
#include "Class1.h"

int Class1::func() {
    if (size < 1) throw string("Array is empty!");
    int a = arr[0];
    for (int i = 1; i < size; i++) {
        a -= arr[i];
    }
    return a;
}
```

## Файл Class1.h

```
#ifndef CLASS1_H
#define CLASS1_H

#include "Base.h"

class Class1 : public Base {
public:
    virtual int func();
};

#endif
```

## Файл Class2.cpp

```
#include "Class2.h"

int Class2::func() {
    if (size < 1) throw string("Array is empty!");
    int a = 0;
    for (int i = 0; i < size; i++) {
        a += arr[i];
    }
    return a;
}
```

## Файл Class2.h

```
#ifndef CLASS2_H
#define CLASS2_H

#include "Class1.h"

class Class2 : public Class1 {
public:
    int func();
};
```

```
#endif
```

## Файл Class3.cpp

```
#include "Class3.h"

void Class3::in_out() {
    cout << "Array dimension: " << size;
    cout << "\nThe original array:";
    for (int i = 0; i < size; i++) {
        printf("%5d", arr[i]);
    }
    cout << "\nMin: " << Class1::func();
    cout << "\nSum: " << func();
}
```

## Файл Class3.h

```
#ifndef CLASS3_H
#define CLASS3_H

#include "Class2.h"

class Class3 : public Class2 {
public:
    void in_out();
};

#endif
```

## Файл main.cpp

```
#include "main.h"
#include "Class3.h"
#include "Base.h"

int main()
{
    Class3 *obj;
    obj = new Class3();
    obj->Base::in_out();
    obj->in_out();
    return(0);
}
```

```
}
```

## Файл main.h

```
#ifndef MAIN_H
#define MAIN_H

#include <stdlib.h>
#include <stdio.h>
#include <iostream>

using namespace std;

#endif
```

## Тестирование

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
3 10 51 13	Array dimension: 3 The original array: 10 51 13 Min: -54 Sum: 74	Array dimension: 3 The original array: 10 51 13 Min: -54 Sum: 74
1 5	Array dimension: 1 The original array: 5 Min: 5 Sum: 5	Array dimension: 1 The original array: 5 Min: 5 Sum: 5

