

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм конструктора класса cl_child.....	10
3.2 Алгоритм метода setup класса cl_child.....	10
3.3 Алгоритм метода print класса cl_child.....	11
3.4 Алгоритм метода change класса cl_parent.....	11
3.5 Алгоритм конструктора класса cl_parent.....	11
3.6 Алгоритм метода setup класса cl_parent.....	12
3.7 Алгоритм метода print класса cl_parent.....	12
3.8 Алгоритм функции main.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	16
5.1 Файл cl_child.cpp.....	16
5.2 Файл cl_child.h.....	16
5.3 Файл cl_parent.cpp.....	17
5.4 Файл cl_parent.h.....	17
5.5 Файл main.cpp.....	18
6 ТЕСТИРОВАНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	20

# 1 ПОСТАНОВКА ЗАДАЧИ

Описать класс `cl_parent` объекта, в котором следующий состав элементов:

В закрытом разделе:

- одно свойство целого типа;
- метод, с одним целочисленным параметром, который меняет значение свойства в закрытом разделе на удвоенное значение параметра.

В открытом разделе:

- одно свойство целого типа;
- параметризованный конструктор, с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе. Значение закрытого свойства меняется посредством вызова метода из закрытого раздела;
- метод с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе. Значение закрытого свойства меняется посредством вызова метода из закрытого раздела;
- метод, который выводит на экран значение обоих свойств. Сперва значение закрытого свойства, потом значение открытого свойства.

Назовем объект данного класса родительским. Соответственно его класс родительским классом.

На базе родительского объекта сконструируем производный объект. Производный объект должен сохранить открытый доступ к открытым элементам родительского класса. Он должен иметь следующие собственные элементы:

В закрытом разделе:

- одно свойство целого типа, наименование которого совпадает с наименованием закрытого свойства родительского объекта;

В открытом разделе:

- одно свойство целого типа, наименование которого совпадает с наименованием открытого свойства родительского объекта;
- параметризованный конструктор, с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе;
- метод с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе. Наименование метода совпадает с наименованием аналогичного метода родительского объекта;
- метод, который выводит на экран значение обоих свойств. Сперва значение закрытого свойства, потом значение открытого свойства. Наименование метода совпадает с наименованием аналогичного метода родительского объекта.

Разработать производный класс используя класс `cl_parent` в качестве родительского.

В основной функции реализовать алгоритм:

1. Ввод значения двух целочисленных переменных.
2. Создать объект производного класса используя целочисленных переменных в конструкторе в качестве аргументов в последовательности, как им были присвоены значения. Первый аргумент содержит значение для свойства закрытого раздела, второй для свойства открытого раздела.
3. Вывод значений свойств родительского объекта.
4. Вывод значений свойств производного объекта.
5. Если исходное значение закрытого свойства больше нуля, то:
  - 5.1. Переопределить значения свойств производного объекта, увеличив на единицу введенные исходные значения.
  - 5.2. Переопределить значения свойств родительского объекта, уменьшив на единицу введенные исходные значения.
  - 5.3. Вывод значений свойств производного объекта.

5.4. Вывод значений свойств родительского объекта.

6. Иначе:

6.1. Переопределить значения свойств родительского объекта, увеличив на единицу введенные исходные значения.

6.2. Переопределить значения свойств производного объекта, уменьшив на единицу введенные исходные значения.

6.3. Вывод значений свойств родительского объекта.

6.4. Вывод значений свойств производного объекта.

## 1.1 Описание входных данных

В первой строке:

«Целое число» «Целое число»

**Пример ввода:**

8 5

## 1.2 Описание выходных данных

Начиная с первой строки:

«Целое число»	«Целое число»
«Целое число»	«Целое число»
«Целое число»	«Целое число»
«Целое число»	«Целое число»

**Пример вывода:**

16	5
8	5
9	6
14	4

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `child` класса `cl_child` предназначен для дочерний объект;
- функция `main` для основная функция программы;
- `cin` - объект стандартного потока ввода с клавиатуры;
- `cout` - объект стандартного потока вывода на экран;
- `if .. else` - условный оператор.

Класс `cl_child`:

- свойства/поля:
  - поле закрытое свойство класса:
    - наименование — `closed`;
    - тип — `int`;
    - модификатор доступа — `private`;
  - поле открытое свойство класса:
    - наименование — `opened`;
    - тип — `int`;
    - модификатор доступа — `public`;
- функционал:
  - метод `cl_child` — параметризованный конструктор;
  - метод `setup` — устанавливает значения свойств в закрытом и открытом разделах;
  - метод `print` — выводит на экран значения полей.

Класс `cl_parent`:

- свойства/поля:
  - поле закрытое поле класса:
    - наименование — `closed`;

- тип — int;
- модификатор доступа — private;
- поле открытое поле класса:
  - наименование — opened;
  - тип — int;
  - модификатор доступа — public;
- функционал:
  - метод change — меняет значение свойства в закрытом разделе;
  - метод cl\_parent — параметризованный конструктор;
  - метод setup — устанавливает значения свойств в закрытом и открытом разделах;
  - метод print — выводит на экран значения полей.

*Таблица 1 – Иерархия наследования классов*

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl_child			дочерний класс	
		cl_parent	virtual public		2
2	cl_parent			родительский класс	

## 3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

### 3.1 Алгоритм конструктора класса `cl_child`

Функционал: параметризованный конструктор.

Параметры: `int value_1`, `int value_2`.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса `cl_child`

№	Предикат	Действия	№ перехода
1		вызов метода <code>change()</code> с параметром <code>value_1</code>	2
2		присваивание полю <code>opened</code> значения <code>value_2</code>	Ø

### 3.2 Алгоритм метода `setup` класса `cl_child`

Функционал: устанавливает значения свойств в закрытом и открытом разделах.

Параметры: `int value_1`, `int value_2`.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода `setup` класса `cl_child`

№	Предикат	Действия	№ перехода
1		присваивание полю <code>closed</code> значения аргумента <code>value_1</code>	2
2		присваивание полю <code>opened</code> значения аргумента <code>value_2</code>	Ø



### 3.3 Алгоритм метода print класса cl\_child

Функционал: выводит на экран значения полей.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода print класса cl\_child

№	Предикат	Действия	№ перехода
1		вывод на экран " " (4 пробела) и переход на новую строку	Ø

### 3.4 Алгоритм метода change класса cl\_parent

Функционал: меняет значение свойства в закрытом разделе.

Параметры: int value.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода change класса cl\_parent

№	Предикат	Действия	№ перехода
1		присваивание закрытому полю closed значение удвоенного аргумента val	Ø

### 3.5 Алгоритм конструктора класса cl\_parent

Функционал: параметризованный конструктор.

Параметры: int value\_1, int value\_2.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса *cl\_parent*

№	Предикат	Действия	№ перехода
1		вызов метода <code>change()</code> с параметром <code>value_1</code>	2
2		присваивание полю <code>opened</code> значение <code>value_2</code>	Ø

### 3.6 Алгоритм метода `setup` класса *cl\_parent*

Функционал: устанавливает значения свойств в закрытом и открытом разделах.

Параметры: `int value_1`, `int value_2`.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода `setup` класса *cl\_parent*

№	Предикат	Действия	№ перехода
1		вызов метода <code>change()</code> с параметром <code>value_1</code>	2
2		присваивание полю <code>opened</code> значение <code>value_2</code>	Ø

### 3.7 Алгоритм метода `print` класса *cl\_parent*

Функционал: выводит на экран значения полей.

Параметры: нет.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода `print` класса *cl\_parent*

№	Предикат	Действия	№ перехода
1		вывод на экран " " (4 пробела) и переход на новую строку	Ø

### 3.8 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		создание переменных value_closed и value_opened типа int	2
2		ввод значений value_closed и value_opened с клавиатуры	3
3		создание объекта child класса cl_child параметрами value_closed и value_opened	4
4		вызов метода print() класса cl_parent объекта child	5
5		вызов метода print() объекта child	6
6	value_closed > 0	вызов метода setup() объекта child с параметрами value_closed+1 и value_opened+1	7
		вызов метода setup() класса cl_parent объекта child с параметрами value_closed+1 и value_opened+1	10
7		вызов метода setup() класса cl_parent объекта child с параметрами value_closed-1 и value_opened-1	8
8		вызов метода print() объекта child	9
9		вызов метода print() класса cl_parent объекта child	∅
10		вызов метода setup() объекта child с параметрами value_closed-1 и value_opened-1	11
11		вызов метода print() класса cl_parent объекта child	12
12		вызов метода print() объекта child	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

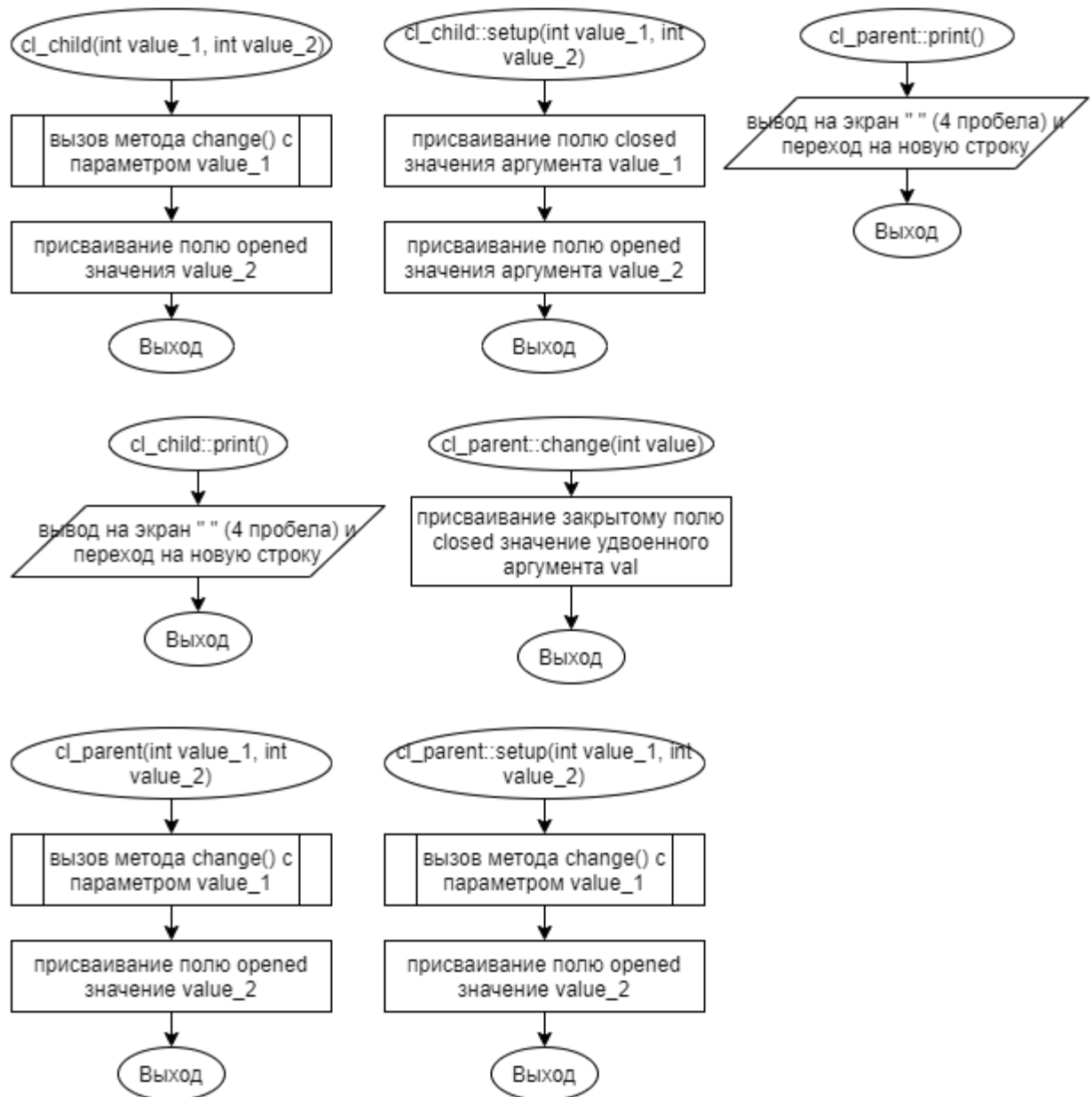


Рисунок 1 – Блок-схема алгоритма

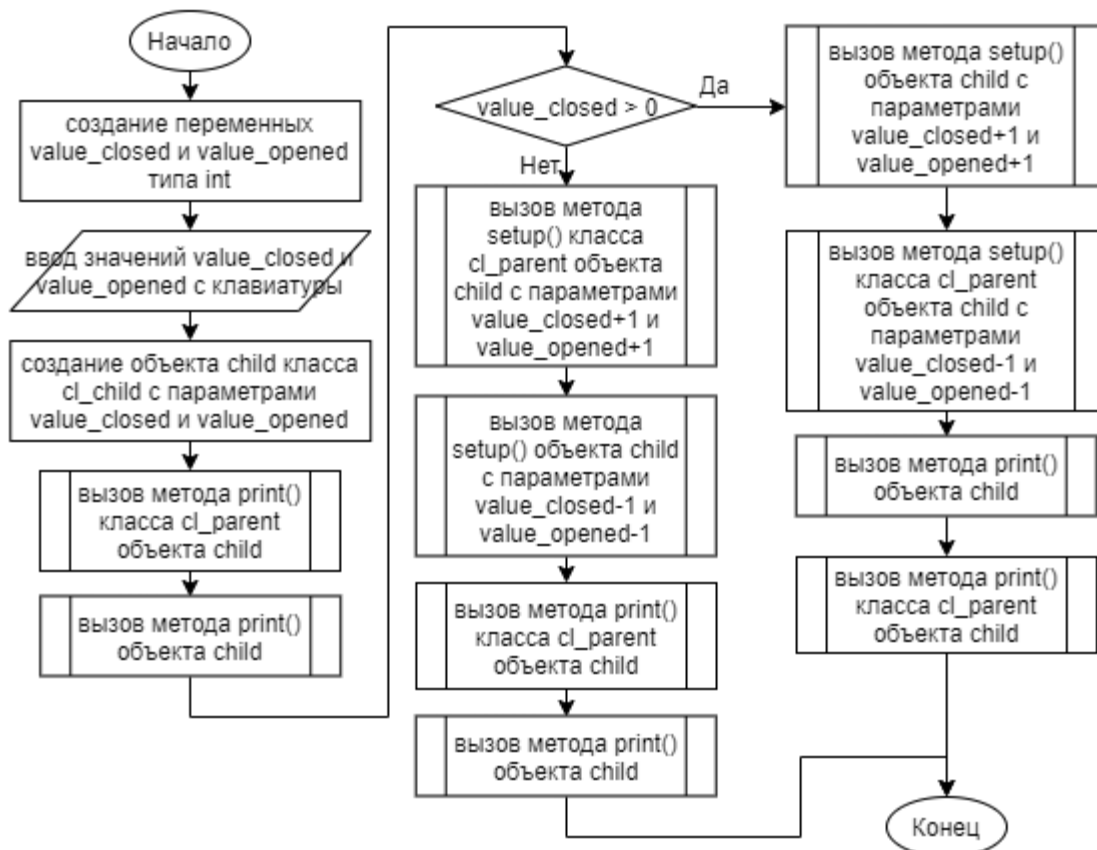


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл `cl_child.cpp`

*Листинг 1 – `cl_child.cpp`*

```
#include <iostream>
#include "cl_child.h"

cl_child::cl_child(int value_1, int value_2) : cl_parent(value_1, value_2)
{
    closed = value_1;
    opened = value_2;
}
void cl_child::setup(int value_1, int value_2)
{
    closed = value_1;
    opened = value_2;
}
void cl_child::print()
{ std::cout << closed << "    " << opened << std::endl; }
```

### 5.2 Файл `cl_child.h`

*Листинг 2 – `cl_child.h`*

```
#ifndef __CL_CHILD__H
#define __CL_CHILD__H

#include "cl_parent.h"

class cl_child: public cl_parent
{
private:
    int closed;
public:
    int opened;
    cl_child(int value_1, int value_2);
    void setup(int value_1, int value_2);
    void print();
}
```

```
};  
  
#endif
```

## 5.3 Файл cl\_parent.cpp

*Листинг 3 – cl\_parent.cpp*

```
#include <iostream>  
#include "cl_parent.h"  
  
void cl_parent::change(int value)  
{ closed = value * 2; }  
cl_parent::cl_parent(int value_1, int value_2)  
{  
    change(value_1);  
    opened = value_2;  
}  
void cl_parent::setup(int value_1, int value_2)  
{  
    change(value_1);  
    opened = value_2;  
}  
void cl_parent::print()  
{ std::cout << closed << "    " << opened << std::endl; }
```

## 5.4 Файл cl\_parent.h

*Листинг 4 – cl\_parent.h*

```
#ifndef __CL_PARENT__H  
#define __CL_PARENT__H  
  
class cl_parent  
{  
private:  
    int closed;  
    void change(int value);  
public:  
    int opened;  
    cl_parent(int value_1, int value_2);  
    void setup(int value_1, int value_2);  
    void print();  
};
```

```
#endif
```

## 5.5 Файл main.cpp

*Листинг 5 – main.cpp*

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "cl_parent.h"
#include "cl_child.h"

int main()
{
    int value_closed, value_opened;
    std::cin >> value_closed >> value_opened;
    cl_child child(value_closed, value_opened);
    child.cl_parent::print();
    child.print();
    if (value_closed > 0)
    {
        child.setup(value_closed + 1, value_opened + 1);
        child.cl_parent::setup(value_closed - 1, value_opened - 1);
        child.print();
        child.cl_parent::print();
    }
    else
    {
        child.cl_parent::setup(value_closed + 1, value_opened + 1);
        child.setup(value_closed - 1, value_opened - 1);
        child.cl_parent::print();
        child.print();
    }
    return(0);
}
```



## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 10.

Таблица 10 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
8 5	16 5 8 5 9 6 14 4	16 5 8 5 9 6 14 4
3 2	6 2 3 2 4 3 4 1	6 2 3 2 4 3 4 1
10 5	20 5 10 5 11 6 18 4	20 5 10 5 11 6 18 4

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.
2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: [https://mirea.aco-avvora.ru/student/files/methodichescoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avvora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf) (дата обращения 05.05.2021).
3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: [https://mirea.aco-avvora.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avvora.ru/student/files/Prilozheniye_k_methodichke.pdf) (дата обращения 05.05.2021).
4. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).