Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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;

− параметризированный 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 наименование закрытого свойства родительского объекта;

В открытом разделе:

− одно свойство целого типа, наименование которого совпадает с наименование открытого свойства родительского объекта;

− параметризированный 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 используются:

1. Объекты стандартного потока ввода/вывода cin/cout
2. Объект класса cl_parent

Таблица ирархии классов

Таблица 1 – Иерархия наследования классов

№	Наименование	Классы наследники	Модификатор доступа	Описание	Номер класса наследника	Комментарий
1	cl_parent			Базовый класс		
		cl_child	public		2	
1	cl_child			Класс производный от cl_parent		

- Класс cl_parent
  - Поля/свойства
    - Целочисленные поля
      - Наименования num_a, num_b
      - Тип данных: целое
      - Модификатор доступа: num_a - private; num_b - public
    - Методы:
      - Конструктор cl_parent
        - Функционал: конструктор с параметрами

        - Метод Pu_change
            - Функционал: присвоение значений полям
        - Метод Print
            - Функционал: вывод значений полей
        - Метод Pr_change
            - Функционал: присвоение закрытому полю значений параметра * 2
- Класс cl_child
    - Поля/свойства
        - Целочисленные поля
            - Наименования num_a, num_b
            - Тип данных: целое
            - Модификатор доступа: num_a - private; num_b - public
        - Методы:
            - Конструктор cl_child
                - Функционал: конструктор с параметрами
            - Метод Pu_change
                - Функционал: присвоение значений полям
            - Метод Print
                - Функционал: вывод значений полей

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_parent

Функционал: Устанавливает значения свойств в закрытом и открытом разделе.

Параметры: int num_a, int num_b.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl_parent

№	Предикат	Действия	№ перехода
1		Вызов метода Pr_change(num_a)	2
2		pu_num = num_b	∅

3.2 Алгоритм метода Pr_change класса cl_parent

Функционал: Устанавливает значения свойств в закрытом разделе.

Параметры: int par.

Возвращаемое значение: Целый тип.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Pr_change класса cl_parent

№	Предикат	Действия	№ перехода
1		pr_num = par * 2	∅

3.3 Алгоритм метода Pu_change класса cl_parent

Функционал: Устанавливает значения свойств в закрытом и открытом разделе.

Параметры: int num_a, int num_b.

Возвращаемое значение: Целый тип.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода Pu_change класса cl_parent

№	Предикат	Действия	№ перехода
1		Pr_change(num_a)	2
2		pu_num = num_b	∅

3.4 Алгоритм метода Print класса cl_parent

Функционал: Вывод значений pr_num и pu_num разделённых 4-мя пробелами.

Параметры: отсутствуют.

Возвращаемое значение: Текст.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Print класса cl_parent

№	Предикат	Действия	№ перехода
1		Вывод значений pr_num и pu_num разделённых 4-мя пробелами	∅

3.5 Алгоритм конструктора класса cl_child

Функционал: Устанавливает значения свойств в закрытом и открытом разделе.

Параметры: int num_a, int num_b.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl_child

№	Предикат	Действия	№ перехода
1		pr_num = num_a	2
2		pu_num = num_b	∅

3.6 Алгоритм метода Pu_change класса cl_child

Функционал: Устанавливает значения свойств в закрытом и открытом разделе.

Параметры: int num_a, int num_b.

Возвращаемое значение: Целый тип.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Pu_change класса cl_child

№	Предикат	Действия	№ перехода
1		pr_num = num_a	2
2		pu_num = num_b	∅

3.7 Алгоритм метода Print класса cl_child

Функционал: Вывод значений pr_num и pu_num разделённых 4-мя пробелами.

Параметры: отсутствуют.

Возвращаемое значение: Текст.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода Print класса cl_child

№	Предикат	Действия	№ перехода
1		Вывод значений pr_num и pu_num разделённых 4-мя пробелами	∅

3.8 Алгоритм функции main

Функционал: Основная функция.

Параметры: отсутствуют.

Возвращаемое значение: Целый тип.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление переменный типа int num_a, num_b	2
2		Ввод переменных num_a, num_b	3
3		Создание указателя на объект obj_a типа cl_parent с параметрами num_a, num_b	4
4		Вызов метода Print() для obj_a типа cl_parent	5
5		Вызов метода Print() для obj_a типа cl_child	6
6	num_a > 0		7
			11
7		Вызов метода Pu_change(num_a+1, num_b+1) для obj_a типа cl_child	8
8		Вызов метода Pu_change(num_a-1, num_b-1) для obj_a типа cl_parent	9
9		Вызов метода Print() для obj_a типа cl_child	10
10		Вызов метода Print() для obj_a типа cl_parent	15
11		Вызов метода Pu_change(num_a+1, num_b+1) для obj_a типа cl_parent	12

№	Предикат	Действия	№ перехода
12		Вызов метода Pu_change(num_a-1, num_b-1) для obj_a типа cl_child	13
13		Вызов метода Print() для obj_a типа cl_parent	14
14		Вызов метода Print() для obj_a типа cl_child	15
15		Отчистка памяти под obj_a	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

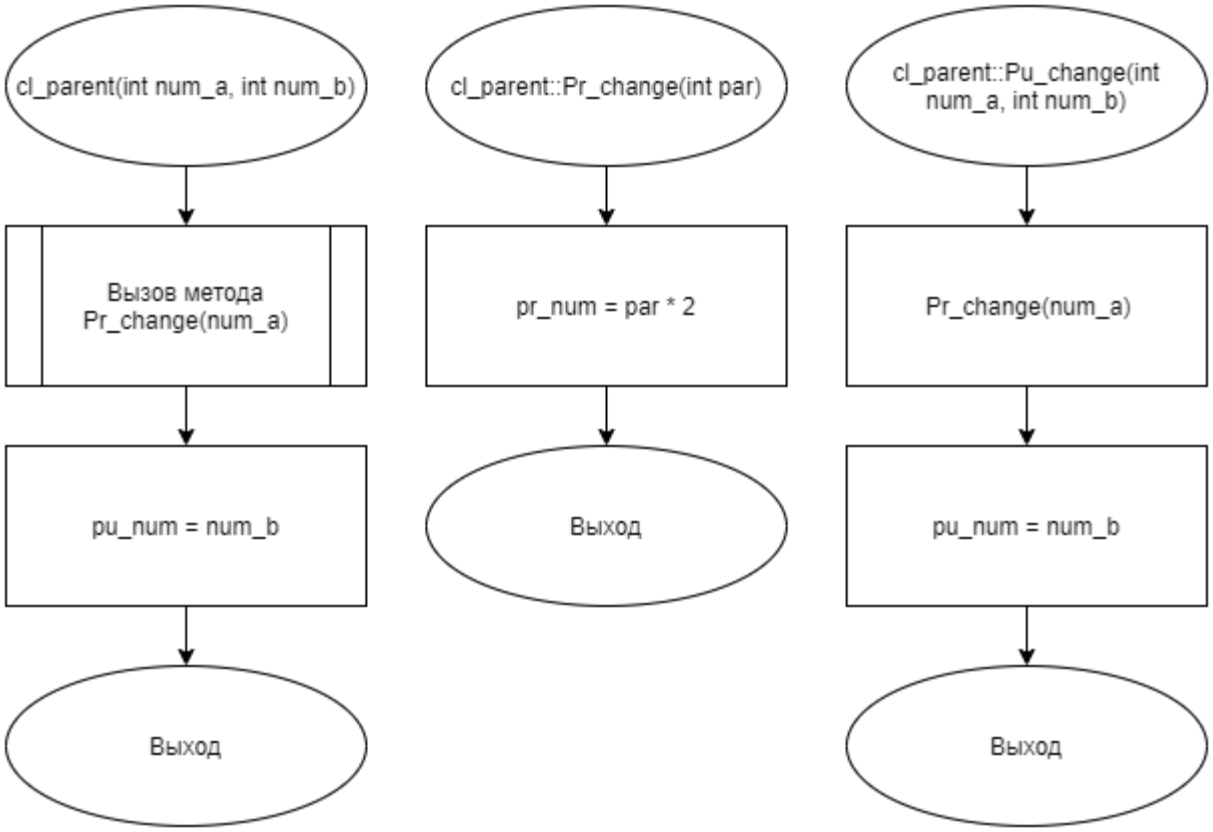

Рисунок 1 – Блок-схема алгоритма

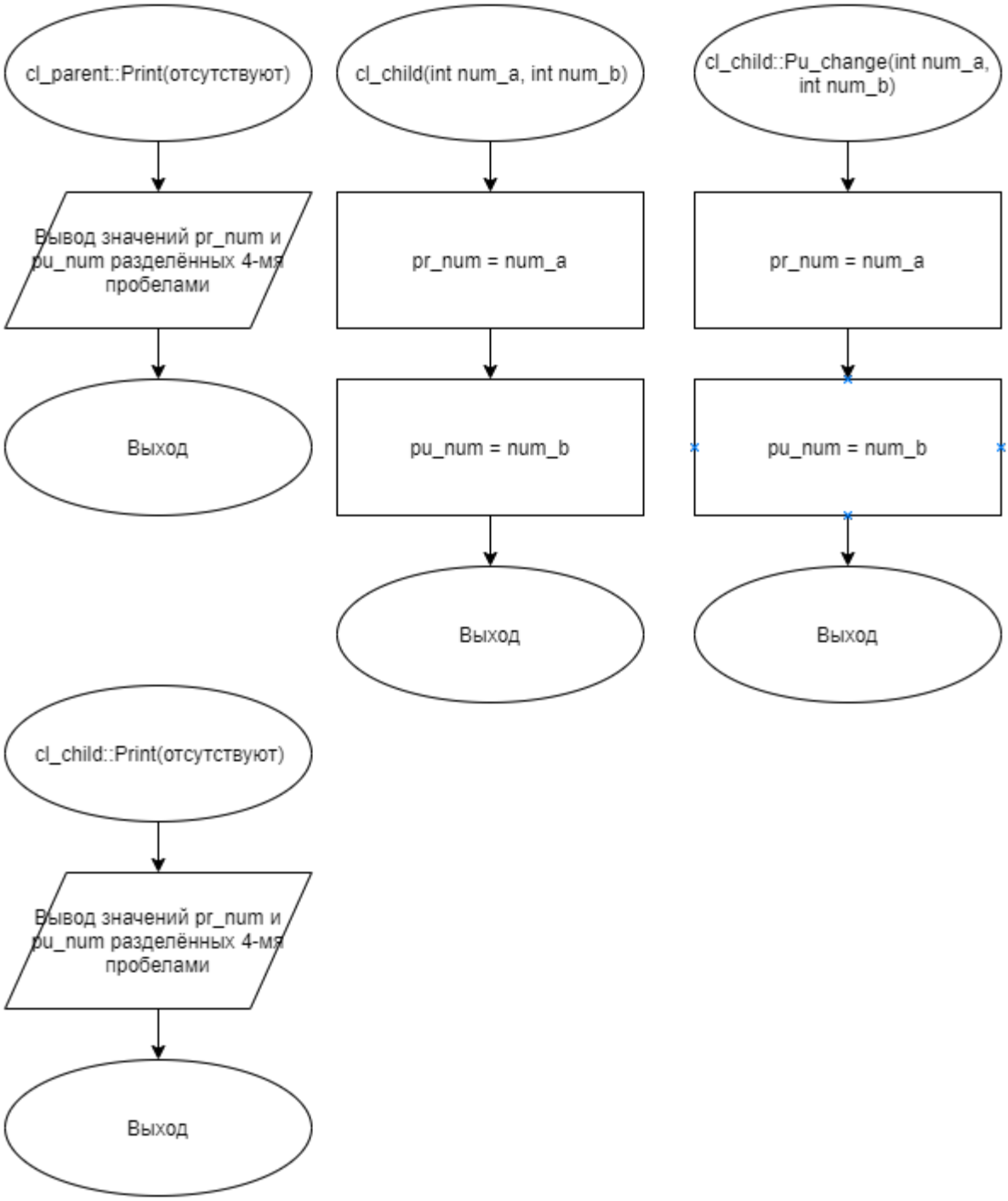

Рисунок 2 – Блок-схема алгоритма

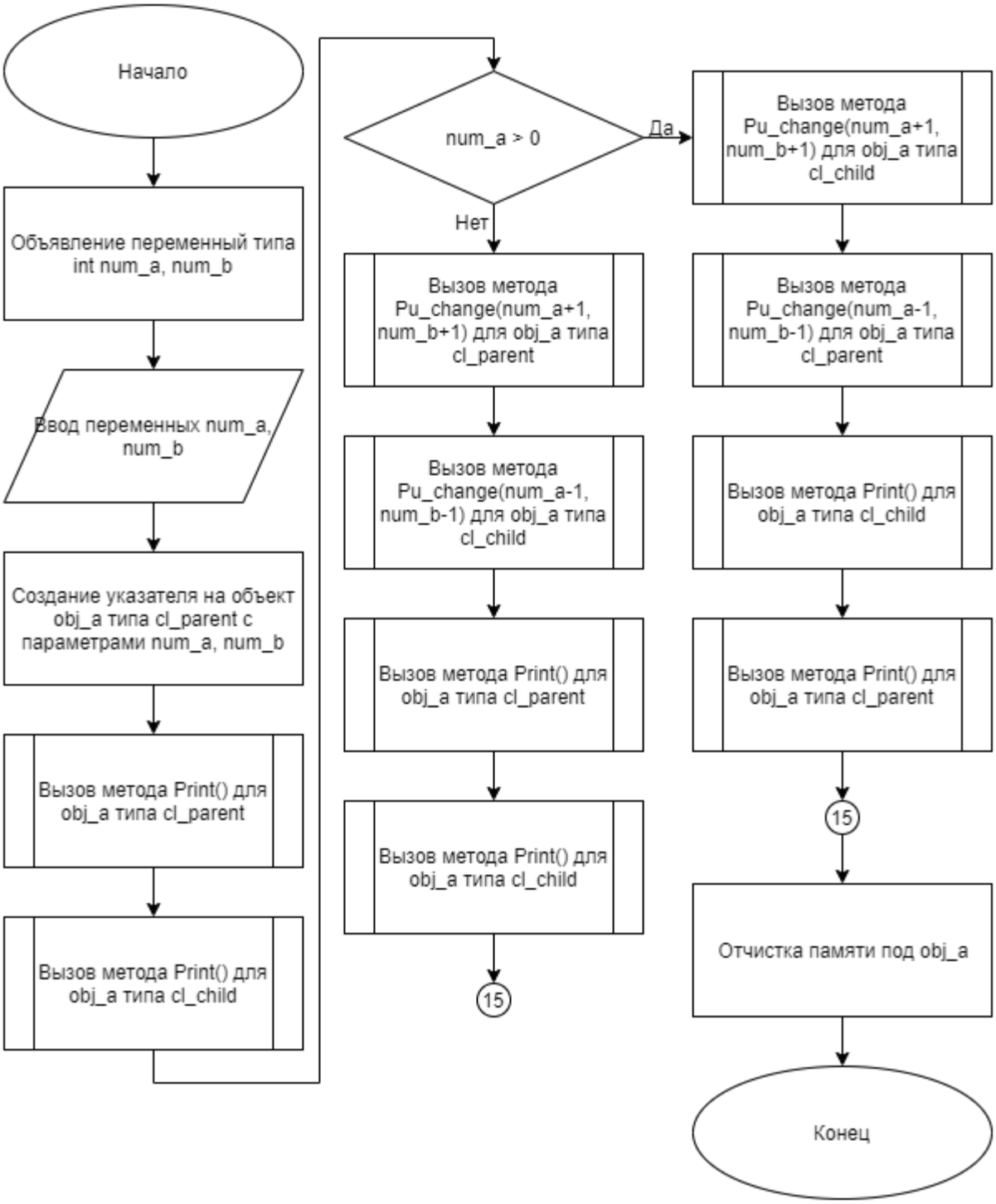

Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl_child.cpp

Листинг 1 – cl_child.cpp

```
#include "cl_child.h"

cl_child::cl_child(int num_a, int num_b):cl_parent(num_a, num_b)
{
	pr_num = num_a;
	pu_num = num_b;
}
void cl_child::Pu_change(int num_a, int num_b)
{
	pr_num = num_a;
	pu_num = num_b;
}
void cl_child::Print()
{
	cout << pr_num << "    " << pu_num;
}
```

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 pr_num;
public:
	int pu_num;
	cl_child(int num_a, int num_b);
	void Pu_change(int num_a, int num_b);
	void Print();
};
#endif
```

5.3 Файл cl_parent.cpp

Листинг 3 – cl_parent.cpp

```
#include "cl_parent.h"

cl_parent::cl_parent(int num_a, int num_b)
{
	Pr_change(num_a);
	pu_num = num_b;
}
void cl_parent::Pr_change(int par)
{
	pr_num = par * 2;
}
void cl_parent::Pu_change(int num_a, int num_b)
{
	Pr_change(num_a);
	pu_num = num_b;
}
void cl_parent::Print()
{
	cout << pr_num << "    " << pu_num;
}
```

5.4 Файл cl_parent.h

Листинг 4 – cl_parent.h

```
#ifndef __CL_PARENT_H__
#define __CL_PARENT_H__
#include <iostream>
using namespace std;
class cl_parent
{
private:
	int pr_num;
	void Pr_change(int par);
public:
	int pu_num;
	cl_parent(int num_a, int num_b);
	void Pu_change(int num_a, int num_b);
	void Print();
};
#endif
```

5.5 Файл main.cpp

Листинг 5 – main.cpp

```
#include "cl_child.h"

int main()
{
	int num_a, num_b;
	cin >> num_a >> num_b;
	cl_parent* obj_a = new cl_child(num_a, num_b);
	((cl_parent*)obj_a)->Print();
	cout << endl;
	((cl_child*)obj_a)->Print();
	if (num_a > 0)
	{
		((cl_child*)obj_a)->Pu_change(num_a+1, num_b+1);
		((cl_parent*)obj_a)->Pu_change(num_a-1, num_b-1);
		cout << endl;
		((cl_child*)obj_a)->Print();
		cout << endl;
		((cl_parent*)obj_a)->Print();
	}
	else
	{
		((cl_parent*)obj_a)->Pu_change(num_a+1, num_b+1);
		((cl_child*)obj_a)->Pu_change(num_a-1, num_b-1);
		cout << endl;
		((cl_parent*)obj_a)->Print();
		cout << endl;
		((cl_child*)obj_a)->Print();
	}
	delete obj_a;
}
```

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
0 0	0 0 0 0 2 1 -1 -1	0 0 0 0 2 1 -1 -1

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.

3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe_posobie_dlya_laboratorny h_rabot_3.pdf (дата обращения 05.05.2021).

4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

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).