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:

- объекты стандартных потоков ввода и вывода cin и cout (используются для ввода с клавиатуры и вывода на экран соответственно);
- условный оператор (оператор ветвления) if...else;
- объект obj класса cl_child.

Класс cl_parent:

- Свойства (поля):
  - Закрытое поле:
    - Наименование - closed;
    - Тип - целое число (int);
    - Модификатор доступа - закрытый;
  - Открытое поле:
    - Наименование - open;
    - Тип - целое число (int);
    - Модификатор доступа - открытый;
- Методы:
  - Параметризованный конструктор:
    - Функционал - создание объекта класса cl_parent с закрытым полем равным удвоенному значению первого параметра и открытым полем равным значению второго параметра;
    - Возвращаемое значение - void (отсутствует);
    - Модификатор доступа - открытый;
    - Параметры - два целочисленных параметра: for_closed и for_open;
  - Метод change_closed:

        - Функционал - присваивание закрытому полю удвоенного значения параметра;
        - Возвращаемое значение - void (отсутствует);
        - Модификатор доступа - закрытый;
        - Параметры - один целочисленный параметр param;
    - Метод set_fields:
        - Функционал - установка для закрытого поля значения равного удвоенному значению первого параметра и для открытого поля значения, равного второму параметру;
        - Возвращаемое значение - void (отсутствует);
        - Модификатор доступа - открытый;
        - Параметры - два целочисленных параметра: for_closed и for_open;
    - Метод print:
        - Функционал - вывод значений закрытого и открытого полей;
        - Возвращаемое значение - void (отсутствует);
        - Модификатор доступа - открытый;
        - Параметры - отсутствуют.

Класс cl_child:

- Свойства (поля):
    - Закрытое поле:
        - Наименование - closed;
        - Тип - целое число (int);
        - Модификатор доступа - закрытый;
    - Открытое поле:
        - Наименование - open;
        - Тип - целое число (int);

        - Модификатор доступа - открытый;
- Методы:
    - Параметризованный конструктор:
        - Функционал - создание объекта класса cl_child с закрытым полем равным значению первого параметра и открытым полем равным значению второго параметра;
        - Возвращаемое значение - void (отсутствует);
        - Модификатор доступа - открытый;
        - Параметры - два целочисленных параметра: for_closed и for_open;
    - Метод set_fields:
        - Функционал - установка для закрытого поля значения равного значению первого параметра и для открытого поля значения, равного второму параметру;
        - Возвращаемое значение - void (отсутствует);
        - Модификатор доступа - открытый;
        - Параметры - два целочисленных параметра: for_closed и for_open;
    - Метод print:
        - Функционал - вывод значений закрытого и открытого полей;
        - Возвращаемое значение - void (отсутствует);
        - Модификатор доступа - открытый;
        - Параметры - отсутствуют.

Таблица иерархии классов:

- Таблица 1 – Иерархия наследования классов

Номер	Имя класса	Классы-наследники	Модификаторы доступа	Описание	Номер

			при наследовании		
1	cl_parent			Базовый класс в иерархии классов	
		cl_child	public		2
2	cl_child			Наследник класса cl_parent	

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 функции main

Функционал: основной алгоритм программы.

Параметры: отсутствуют.

Возвращаемое значение: целочисленный код завершения работы программы.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной value1 значением 0	2
2		Инициализация целочисленной переменной value2 значением 0	3
3		Ввод value1	4
4		Ввод value2	5
5		Создание объекта obj класса cl_child с использованием параметризованного конструктора и значениями value1 и value 2 в качестве параметров	6
6		Вызов метода print родительского объекта объекта obj	7
7		Вывод "\n" (переход на новую строку)	8
8		Вызов метода print объекта obj	9

№	Предикат	Действия	№ перехода
9		Вывод "\n" (переход на новую строку)	10
10	Значение переменной value1 больше 0		11
			16
11		Вызов метода set_fields объекта obj с значениями суммы значения value1 и 1 и суммы значения value2 и 1 в качестве параметров	12
12		Вызов метода set_fields родительского объекта объекта obj с значениями разности значения value1 и 1 и разности значения value2 и 1 в качестве параметров	13
13		Вызов метода print объекта obj	14
14		Вывод "\n" (переход на новую строку)	15
15		Вызов метода print родительского объекта объекта obj	∅
16		Вызов метода set_fields родительского объекта объекта obj с значениями суммы значения value1 и 1 и суммы значения value2 и 1 в качестве параметров	17
17		Вызов метода set_fields объекта obj с значениями разности значения value1 и 1 и разности значения value2 и 1 в качестве параметров	18
18		Вызов метода print родительского объекта объекта obj	19
19		Вывод "\n" (переход на новую строку)	20

№	Предикат	Действия	№ перехода
20		Вызов метода print объекта obj	∅

3.2 Алгоритм конструктора класса cl_parent

Функционал: создание объекта класса cl_parent с закрытым полем равным удвоенному значению первого параметра и открытым полем равным значению второго параметра.

Параметры: два целочисленных параметра: for_closed и for_open.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса cl_parent

№	Предикат	Действия	№ перехода
1		Вызов метода change_closed с параметром for_closed	2
2		Присваивание полю open параметра for_open	∅

3.3 Алгоритм метода change_closed класса cl_parent

Функционал: присваивание закрытому полю удвоенного значения параметра.

Параметры: один целочисленный параметр param.

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода change_closed класса cl_parent

№	Предикат	Действия	№ перехода
1		Присваивание полю closed результата умножения параметра param на 2	∅

3.4 Алгоритм метода set_fields класса cl_parent

Функционал: установка для закрытого поля значения равного удвоенному значению первого параметра и для открытого поля значения, равного второму параметру.

Параметры: два целочисленных параметра: for_closed и for_open.

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода set_fields класса cl_parent

№	Предикат	Действия	№ перехода
1		Вызов метода change_closed с параметром for_closed	2
2		Присваивание полю open параметра for_open	∅

3.5 Алгоритм метода print класса cl_parent

Функционал: вывод значений закрытого и открытого полей.

Параметры: отсутствуют.

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода print класса cl_parent

№	Предикат	Действия	№ перехода
1		Вывод значения поля closed, " " (4 пробела) и поля open	∅

3.6 Алгоритм конструктора класса cl_child

Функционал: создание объекта класса cl_child с закрытым полем равным значению первого параметра и открытым полем равным значению второго параметра.

Параметры: два целочисленных параметра: for_closed и for_open.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса cl_child

№	Предикат	Действия	№ перехода
1		Присваивание полю closed параметра for_closed	2
2		Присваивание полю open параметра for_open	∅

3.7 Алгоритм метода set_fields класса cl_child

Функционал: установка для закрытого поля значения равного значению первого параметра и для открытого поля значения, равного второму параметру.

Параметры: два целочисленных параметра: for_closed и for_open.

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода set_fields класса cl_child

№	Предикат	Действия	№ перехода
1		Присваивание полю closed параметра for_closed	2
2		Присваивание полю open параметра for_open	∅

3.8 Алгоритм метода print класса cl_child

Функционал: вывод значений закрытого и открытого полей.

Параметры: отсутствуют.

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода print класса cl_child

№	Предикат	Действия	№ перехода
1		Вывод значения поля closed, " " (4 пробела) и поля open	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-11.

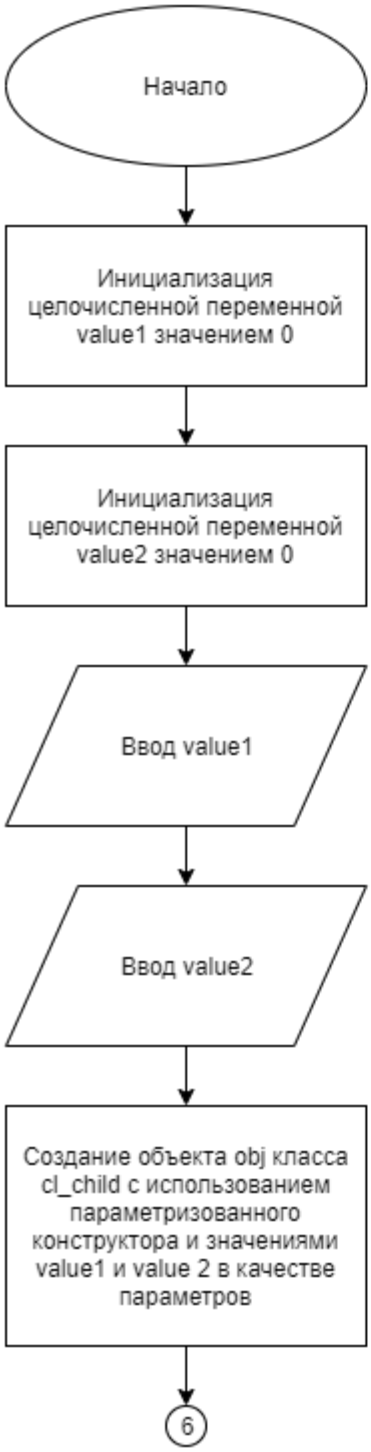

Рисунок 1 – Блок-схема алгоритма

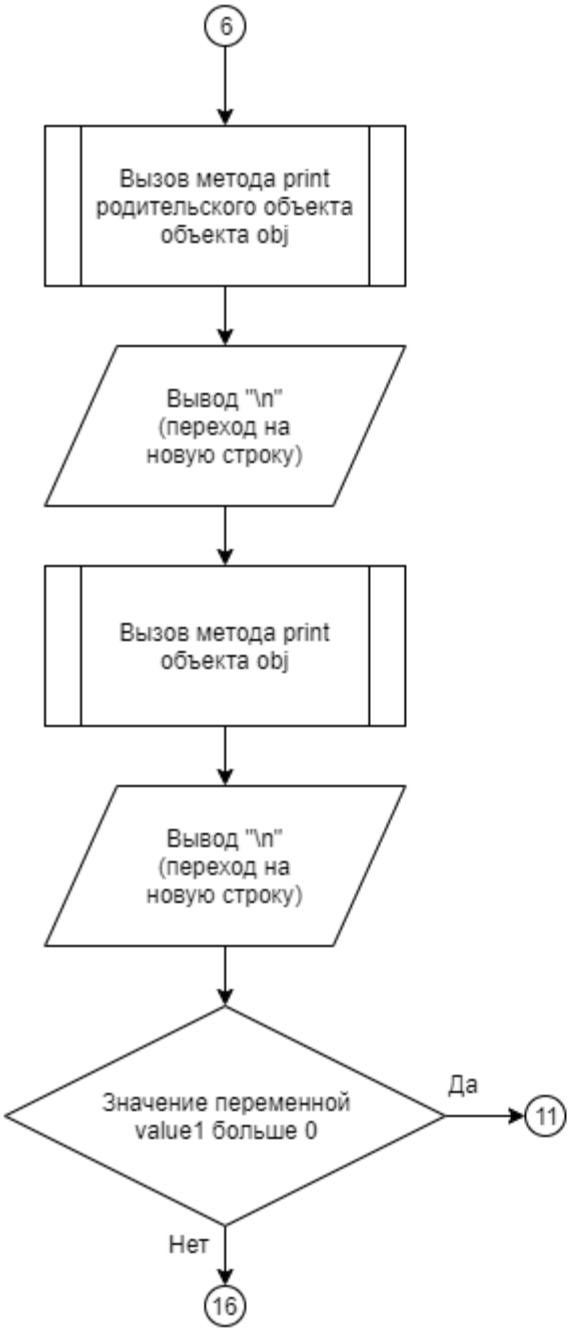

Рисунок 2 – Блок-схема алгоритма

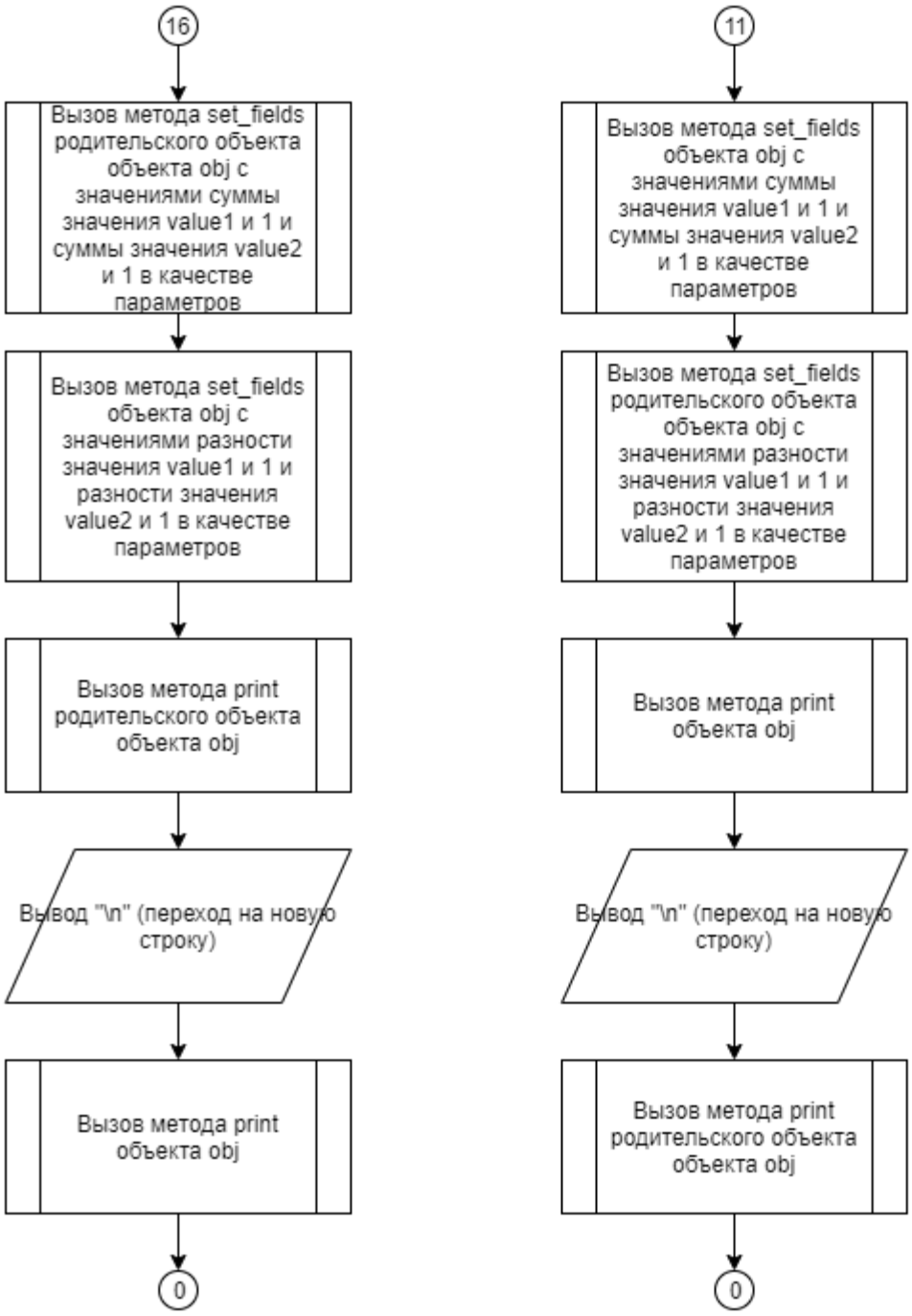

Рисунок 3 – Блок-схема алгоритма

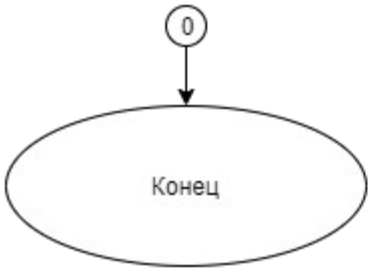

Рисунок 4 – Блок-схема алгоритма

Рисунок 5 – Блок-схема алгоритма

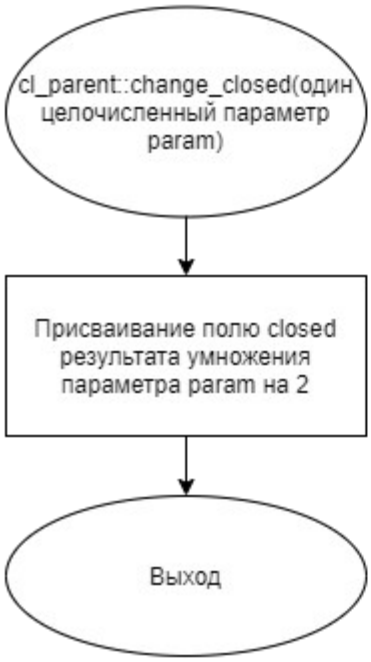

Рисунок 6 – Блок-схема алгоритма

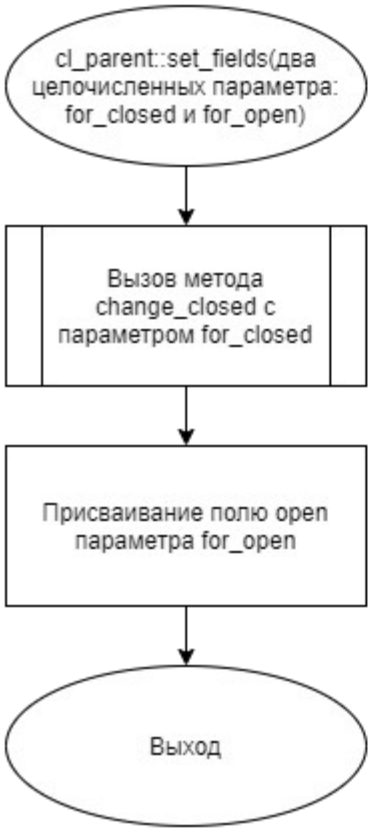

Рисунок 7 – Блок-схема алгоритма

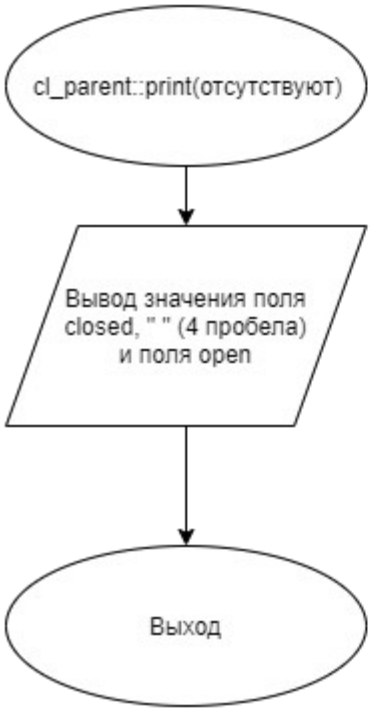

Рисунок 8 – Блок-схема алгоритма

Рисунок 9 – Блок-схема алгоритма

Рисунок 10 – Блок-схема алгоритма

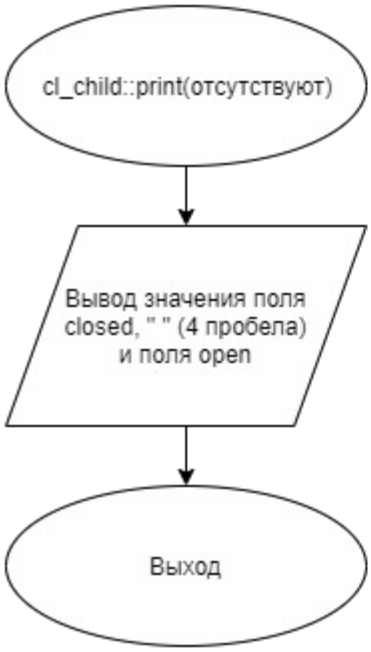

Рисунок 11 – Блок-схема алгоритма

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"

// Параметризованный конструктор
cl_child::cl_child(int for_closed, int for_open) : cl_parent(for_closed, for_open)
{
	// Присваивание закрытому полю параметра for_closed
	closed = for_closed;
	// Присваивание открытому полю параметра for_open
	open = for_open;
}

// Метод установки значений открытого и закрытого полей
void cl_child::set_fields(int for_closed, int for_open)
{
	// Присваивание закрытому полю параметра for_closed
	closed = for_closed;
	// Присваивание открытому полю параметра for_open
	open = for_open;
}

// Метод вывода значений открытого и закрытого полей
void cl_child::print()
{
	// Вывод значений открытого и закрытого полей через 4 пробела
	std::cout << closed << "    " << open;
}
```

5.2 Файл cl_child.h

Листинг 2 – cl_child.h

```
#ifndef __CL_CHILD__H
#define __CL_CHILD__H

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

```
class cl_child : public cl_parent
{
private:
	// Закрытое поле
	int closed;
public:
	// Открытое поле
	int open;
	// Параметризованный конструктор
	cl_child(int for_closed, int for_open);
	// Метод установки значений открытого и закрытого полей
	void set_fields(int for_closed, int for_open);
	// Метод вывода значений открытого и закрытого полей
	void print();
};

#endif
```

5.3 Файл cl_parent.cpp

Листинг 3 – cl_parent.cpp

```
#include "cl_parent.h"

// Метод установки значения закрытого поля
void cl_parent::change_closed(int param)
{
	// Присваивание закрытому полю удвоенного значения параметра
	closed = param * 2;
}

// Параметризованный конструктор
cl_parent::cl_parent(int for_closed, int for_open)
{
	// Вызов метода установки значения закрытого поля
	change_closed(for_closed);
	// Присваивание открытому полю значения параметра for_open
	open = for_open;
}

// Метод установки значений закрытого и открытого полей
void cl_parent::set_fields(int for_closed, int for_open)
{
	// Вызов метода установки значения закрытого поля
	change_closed(for_closed);
	// Присваивание открытому полю значения параметра for_open
	open = for_open;
}

// Метод вывода значений открытого и закрытого полей
void cl_parent::print()
{
	// Вывод значений открытого и закрытого полей через 4 пробела
```

```
	std::cout << closed << "    " << open;
}
```

5.4 Файл cl_parent.h

Листинг 4 – cl_parent.h

```
#ifndef __CL_PARENT__H
#define __CL_PARENT__H

#include <iostream>

class cl_parent
{
private:
	// Закрытое поле
	int closed;
	// Метод установки значения закрытого поля
	void change_closed(int param);
public:
	// Открытое поле
	int open;
	// Параметризованный конструктор
	cl_parent(int for_closed, int for_open);
	// Метод установки значений закрытого и открытого полей
	void set_fields(int for_closed, int for_open);
	// Метод вывода значений открытого и закрытого полей
	void print();
};

#endif
```

5.5 Файл main.cpp

Листинг 5 – main.cpp

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

int main()
{
	// Инициализация целочисленной переменной value1
	int value1 = 0;
	// Инициализация целочисленной переменной value2
	int value2 = 0;
	// Ввод value1
	std::cin >> value1;
	// Ввод value2
```

```
	std::cin >> value2;
	// Создание объекта obj класса cl_child
	// с использованием параметризованного
	// конструктора
	cl_child obj(value1, value2);

	// Вызов метода print родительского
	// объекта объекта obj
	obj.cl_parent::print();
	// Переход на новую строку
	std::cout << '\n';
	// Вызов метода print объекта obj
	obj.print();
	// Переход на новую строку
	std::cout << '\n';

	// Если значение value1 больше 0
	if (value1 > 0)
	{
		// Вызов метода set_fields объекта obj
		obj.set_fields(value1 + 1, value2 + 1);
		// Вызов метода set_fields родительского
		// объекта объекта obj
		obj.cl_parent::set_fields(value1 - 1, value2 - 1);

		// Вызов метода print объекта obj
		obj.print();
		// Переход на новую строку
		std::cout << '\n';
		// Вызов метода print родительского
		// объекта объекта obj
		obj.cl_parent::print();
	}
	else
	{
		// Вызов метода set_fields родительского
		// объекта объекта obj
		obj.cl_parent::set_fields(value1 + 1, value2 + 1);
		// Вызов метода set_fields объекта obj
		obj.set_fields(value1 - 1, value2 - 1);

		// Вызов метода print родительского
		// объекта объекта obj
		obj.cl_parent::print();
		// Переход на новую строку
		std::cout << '\n';
		// Вызов метода print объекта obj
		obj.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
-8 5	-16 5 -8 5 -14 6 -9 4	-16 5 -8 5 -14 6 -9 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-avrora.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-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

4. Шилдт Г. C++: 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).