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 используется:

- объект parent класса Parent предназначен для Родительский класс;
- объект children класса Children предназначен для Производный класс;
- Объект стандортного потока ввода с клавиатуры cin;
- Объект стандартного потока вывода на экран cout;
- Условный оператор if..else.

Класс Parent:

- свойства/поля:
    - поле Закрытое свойство:
        - наименование — close;
        - тип — int;
        - модификатор доступа — private;
    - поле Открытое свойство:
        - наименование — open;
        - тип — int;
        - модификатор доступа — public;
- функционал:
    - метод Parent — Конструктор параметризованный;
    - метод change — Изменение значения закрытого свойства;
    - метод set — Установка значения свойств;
    - метод output — Вывод значения свойств на экран.

Класс Children:

- свойства/поля:
    - поле Закрытое свойство:
        - наименование — close;

        - тип — int;
        - модификатор доступа — private;
    - поле Открытое свойство:
        - наименование — open;
        - тип — int;
        - модификатор доступа — public;
- функционал:
    - метод Children — Конструктор параметризованный;
    - метод set — Изменение значения свойств;
    - метод output — Вывод значения свойств на экран.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	Parent			Родительский класс	
		Children	public		2
2	Children			Производный 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 Parent

Функционал: Конструктор параметризованный.

Параметры: Целочисленные переменые x и y для инициализации закрытого и открытого свойства.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Parent

№	Предикат	Действия	№ перехода
1		Вызов метода change с параметром x	2
2		Открытое свойство инициализируется зачением y	∅

3.2 Алгоритм метода change класса Parent

Функционал: Изменение значения закрытого свойства.

Параметры: Целочисленная переменная x для инициализации закрытого свойства.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода change класса Parent

№	Предикат	Действия	№ перехода
1		Закрытому свойству присваивается зачение x	∅

3.3 Алгоритм метода set класса Parent

Функционал: Установка значения свойств.

Параметры: Целочисленные переменые x и y для изменения значения закрытого и открытого свойства.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода set класса Parent

№	Предикат	Действия	№ перехода
1		Вызов метода change с параметром x	2
2		Открытому свойству присваивается зачение y	∅

3.4 Алгоритм метода output класса Parent

Функционал: Вывод значения свойств на экран.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода output класса Parent

№	Предикат	Действия	№ перехода
1		Вывод на экран "(close) 4 пробела (open)"	∅

3.5 Алгоритм конструктора класса Children

Функционал: Конструктор параметризованный.

Параметры: Целочисленные переменые x и y для инициализации закрытого и открытого свойства.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса Children

№	Предикат	Действия	№ перехода
1		Закрытое свойство инициализируется зачением x	2
2		Открытое свойство инициализируется зачением y	∅

3.6 Алгоритм метода set класса Children

Функционал: Изменение значения свойств.

Параметры: Целочисленные переменые x и y для изменения значения закрытого и открытого свойства.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода set класса Children

№	Предикат	Действия	№ перехода
1		Закрытому свойству присваивается зачение x	2
2		Открытому свойству присваивается зачение y	∅

3.7 Алгоритм метода output класса Children

Функционал: Вывод значения свойств на экран.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода output класса Children

№	Предикат	Действия	№ перехода
1		Вывод на экран "(close) 4 пробела (open)"	∅

3.8 Алгоритм функции main

Функционал: Работа с родительским и производным классами.

Параметры: нет.

Возвращаемое значение: Целочисленное значение.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных x и y	2
2		Ввод с клавиатуры значения переменных x и y	3
3		Создание объекта children класса Children с параметрами x и y	4
4		Вызов метода output базового класса для children	5
5		Вызов метода output объекта children	6
6	x > 0	Вызов метода set объекта children с параметрами x+1 и y+1	7
		Вызов метода set базового класса для children с параметрами x+1 и y+1	10
7		Вызов метода set базового класса для childrenc параметрами x-1 и y-1	8
8		Вызов метода output объекта children	9
9		Вызов метода output базового класса для children	∅
10		Вызов метода set объекта children с параметрами x-1 и y-1	11
11		Вызов метода output базового класса для children	12
12		Вызов метода output объекта children	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

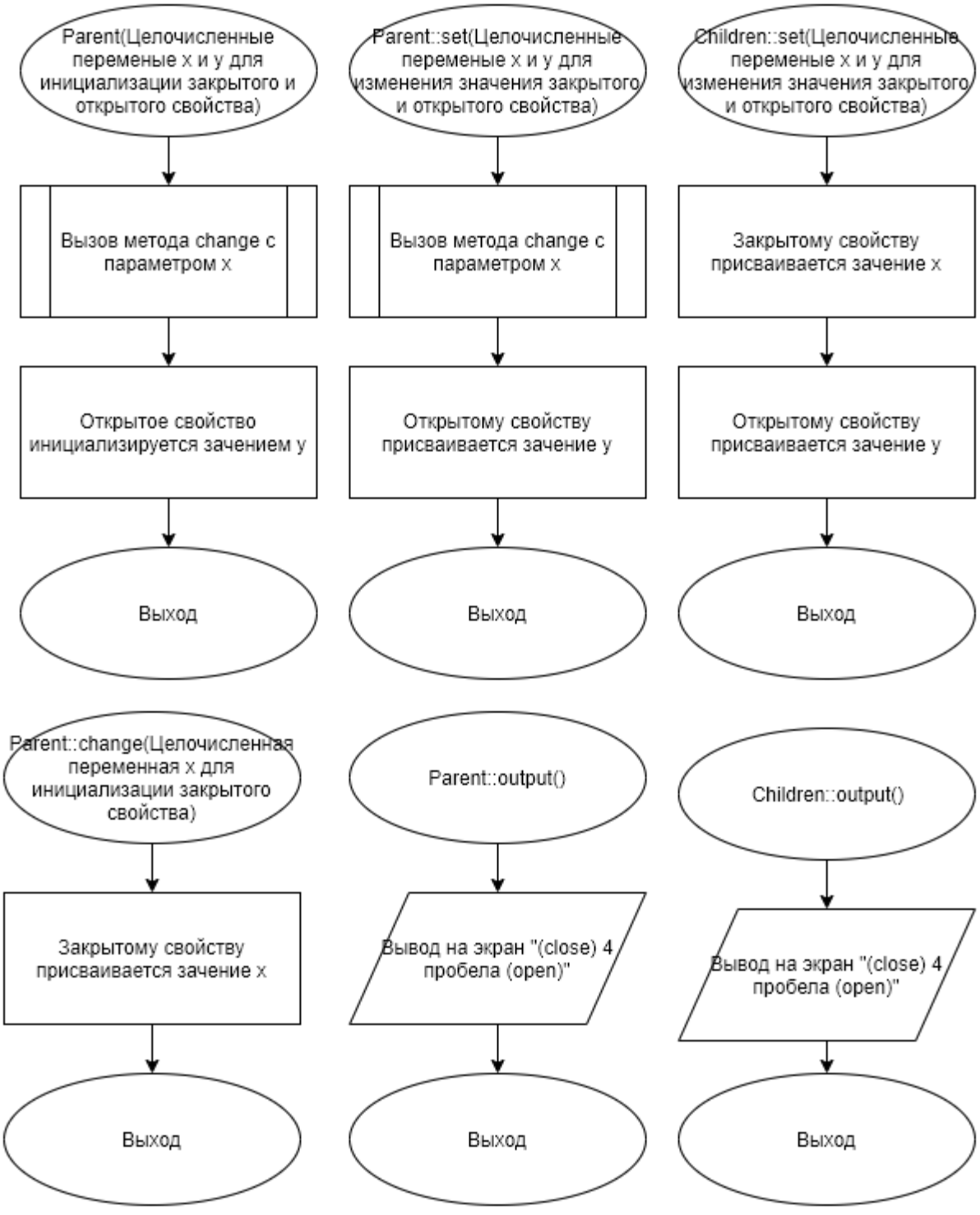

Рисунок 1 – Блок-схема алгоритма

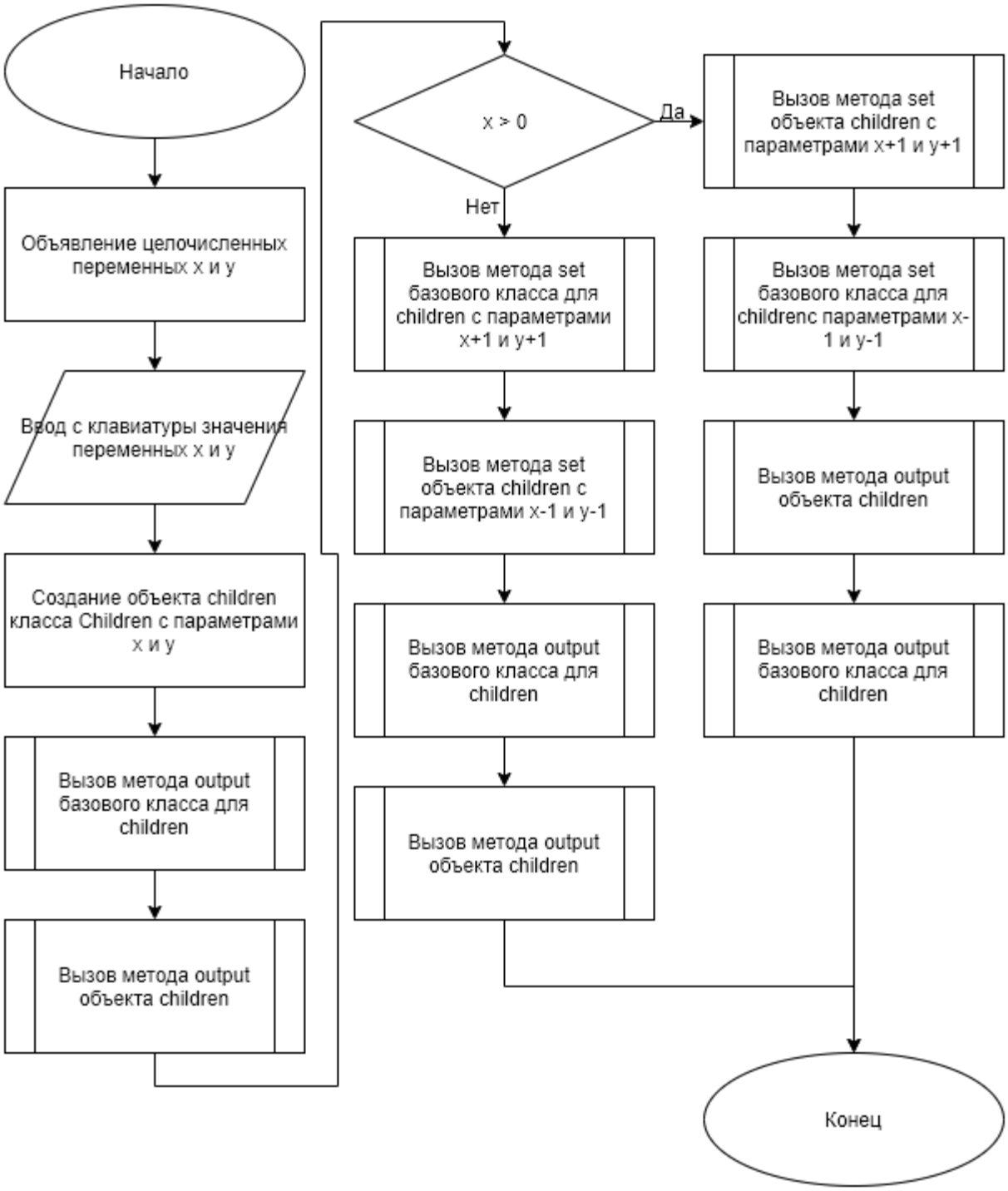

Рисунок 2 – Блок-схема алгоритма

Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Children.cpp

Листинг 1 – Children.cpp

```
#include "Children.h"

Children::Children(int x, int y):Parent(x, y)
{
   close = x;
   open = y;
}

void Children::set(int x, int y)
{
   close = x;
   open = y;
}

void Children::output()
{
   cout << close << "    " << open << endl;
}
```

5.2 Файл Children.h

Листинг 2 – Children.h

```
#ifndef __CHILDREN__H
#define __CHILDREN__H
#include <iostream>
#include "Parent.h"

using namespace std;

class Children: public Parent
{
private:
   int close;
```

```
public:
   int open;
   Children(int x, int y);
   void set(int x, int y);
   void output();
};

#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "Parent.h"
#include "Children.h"

using namespace std;

int main()
{
   int x, y;
   cin >> x >> y;
   Children children(x, y);
   children.Parent::output();
   children.output();
   if (x > 0) {
      children.set(x + 1, y + 1);
      children.Parent::set(x - 1, y - 1);
      children.output();
      children.Parent::output();
   }
   else {
      children.Parent::set(x + 1, y + 1);
      children.set(x - 1, y - 1);
      children.Parent::output();
      children.output();
   }
}
```

5.4 Файл Parent.cpp

Листинг 4 – Parent.cpp

```
#include "Parent.h"

Parent::Parent(int x, int y)
{
   change(x);
   open = y;
}

void Parent::change(int x)
{
   close = 2 * x;
}

void Parent::set(int x, int y)
{
   change(x);
   open = y;
}

void Parent::output()
{
   cout << close << "    " << open << endl;
}
```

5.5 Файл Parent.h

Листинг 5 – Parent.h

```
#ifndef __PARENT__H
#define __PARENT__H
#include <iostream>

using namespace std;

class Parent
{
private:
   int close;
   void change(int x);

public:
   int open;
   Parent(int x, int y);
   void set(int x, int y);
```

```
        void output();
};

#endif
```

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 4	6 4 3 4 4 5 4 3	6 4 3 4 4 5 4 3
0 1	0 1 0 1 2 2 -1 0	0 1 0 1 2 2 -1 0

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).