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:

- стандартный объект потока ввода и вывода cin, cout.

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: параметризированный конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса cl_parent

№	Предикат	Действия	№ перехода
1		b =b	2
2		вызов func1(a)	∅

3.2 Алгоритм метода func1 класса cl_parent

Функционал: функция в закрытом доступе.

Параметры: нет.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода func1 класса cl_parent

№	Предикат	Действия	№ перехода
1		a = a * 2	∅

3.3 Алгоритм метода set класса cl_parent

Функционал: переназначение переменных.

Параметры: нет.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода set класса cl_parent

№	Предикат	Действия	№ перехода
1		a = a	2
2		b = b	∅

3.4 Алгоритм метода print класса cl_parent

Функционал: вывод перменных.

Параметры: нет.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода print класса cl_parent

№	Предикат	Действия	№ перехода
1		вывод a, b	∅

3.5 Алгоритм конструктора класса cl_child

Функционал: параметризированный конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса cl_child

№	Предикат	Действия	№ перехода
1		a = a	2
2		b = b	∅

3.6 Алгоритм метода set класса cl_child

Функционал: переназначение переменных.

Параметры: нет.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода set класса cl_child

№	Предикат	Действия	№ перехода
1		a = a	2
2		b = b	∅

3.7 Алгоритм метода print класса cl_child

Функционал: вывод переменныйх.

Параметры: нет.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода print класса cl_child

№	Предикат	Действия	№ перехода
1		вывод a, b	∅

3.8 Алгоритм функции main

Функционал: основная функция.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 8.

Таблица 8 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		ввод значения двух целочисленных переменных	2
2		создать объект производного класса используя целочисленные пеерменные в конструкторе в качестве аргумента в последовательности, как им были присвоены значения	3
3		вывод значений свойств родительного и производного объекта	4
4	исходное значение закрытогго свойства больше нуля	переопределить значения свойств производног объекта, увеличив на единицу введнные исходные значения	5
		переопределить значения свойств производног объекта, увеличив на единицу введнные исходные значения	7
5		переопределить значения свойств производног объекта, уменьшив на единицу введнные исходные значения	6
6		вывод значний свойств производного и родительског=о обекта	∅
7		переопределить значения свойств производног объекта, уменьшив на единицу введнные исходные значения	8

№	Предикат	Действия	№ перехода
8		вывод значений свойств производного и родительского объекта	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

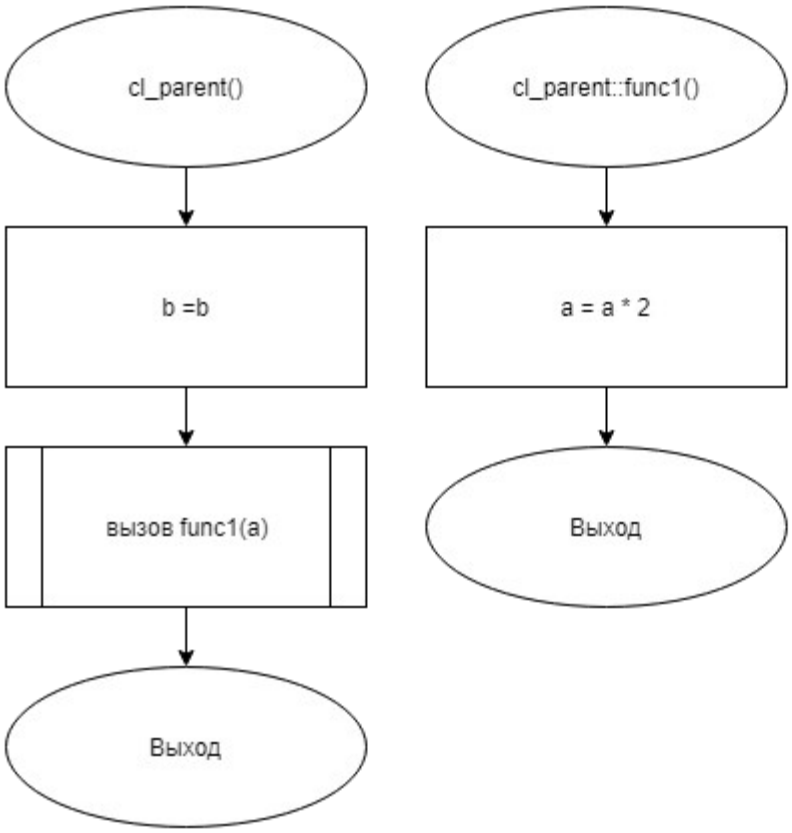

Рисунок 1 – Блок-схема алгоритма

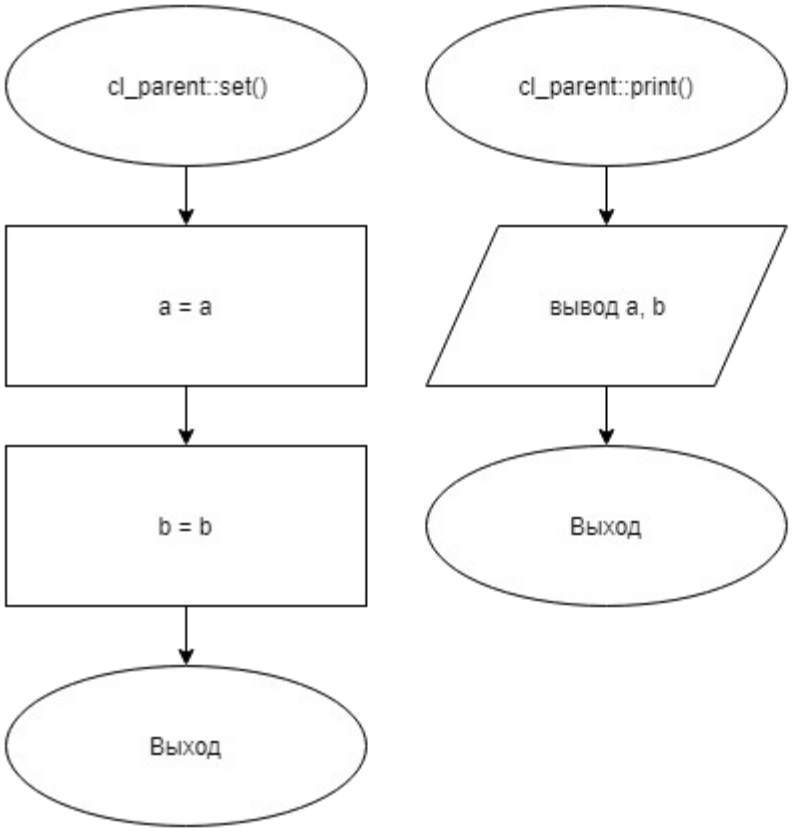

Рисунок 2 – Блок-схема алгоритма

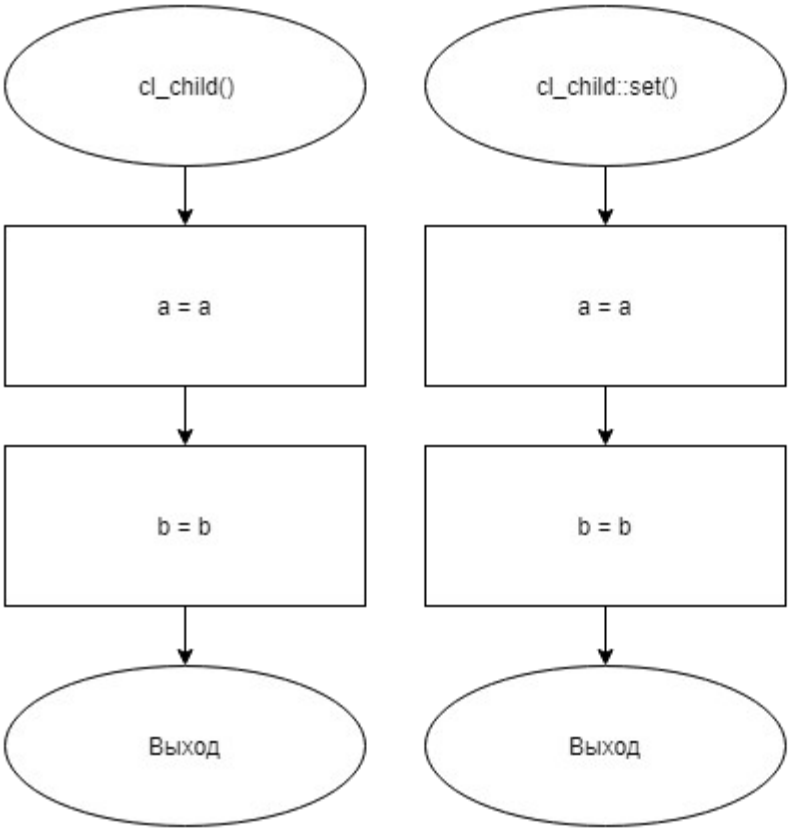

Рисунок 3 – Блок-схема алгоритма

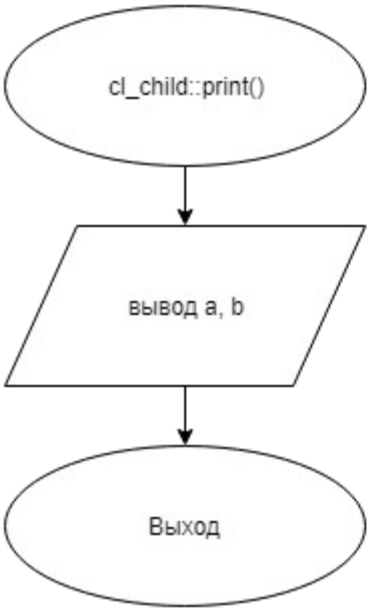

Рисунок 4 – Блок-схема алгоритма

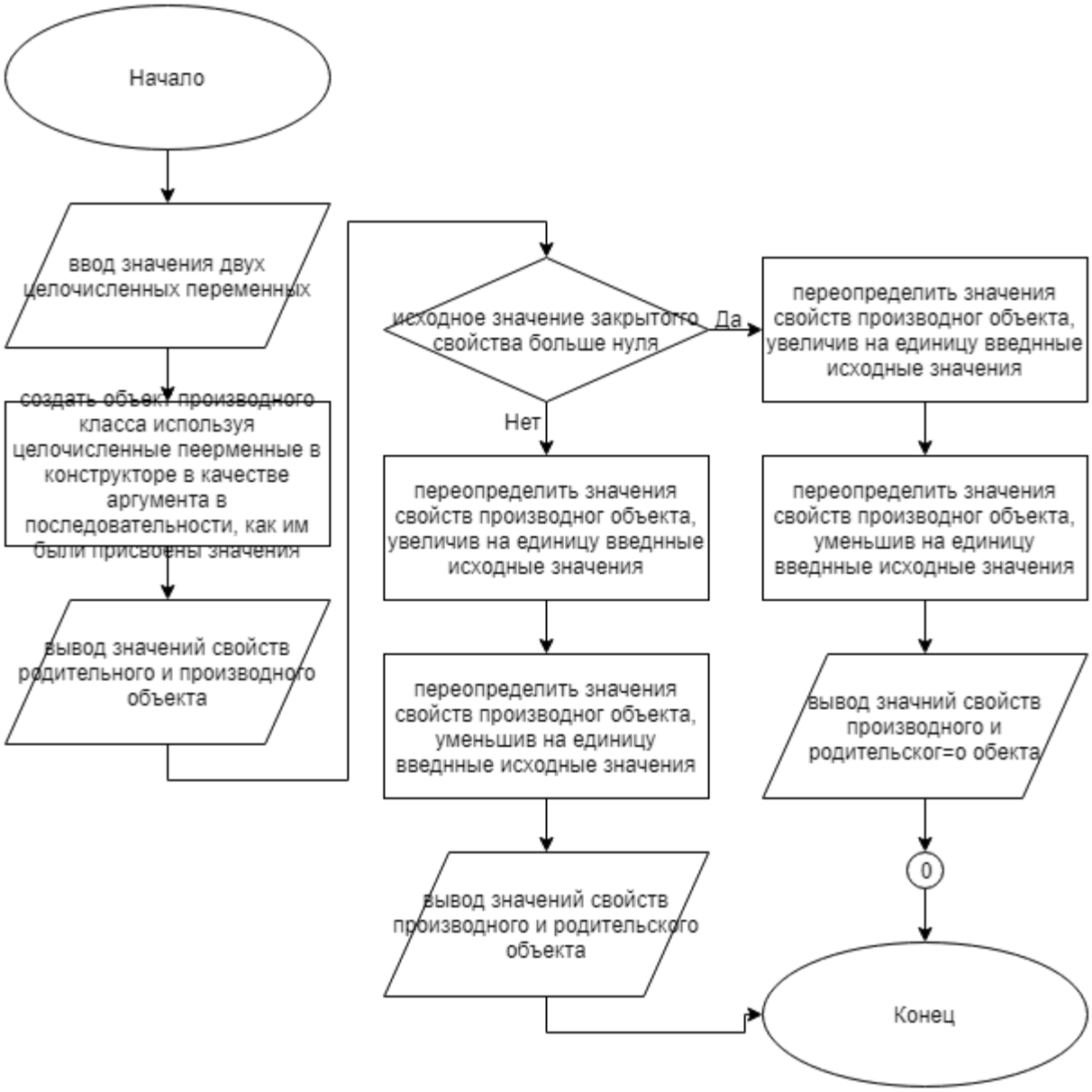

Рисунок 5 – Блок-схема алгоритма

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 a, int b):cl_parent(a, b), a(a), b(b){}

void cl_child::set(int a, int b)
{
   this->a = a;
   this->b = b;
}

void cl_child::print()
{
   cout << a << "    " << b;
}
```

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 a;
public:
   int b;
   cl_child(int a, int b);
   void set(int a, int b);
   void print();
};
#endif
```

5.3 Файл cl_parent.cpp

Листинг 3 – cl_parent.cpp

```
#include "cl_parent.h"

cl_parent::cl_parent(int a, int b):b(b)
{
   func1(a);
}

void cl_parent::func1(int a)
{
   this->a = a * 2;
}

void cl_parent::set(int a, int b)
{
   func1(a);
   this->b = b;
}

void cl_parent::print()
{
   cout << a << "    " << b;
}
```

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 a;
   void func1(int a);
public:
   int b;
   cl_parent(int a, int b);
   void set(int a, int b);
```

```
   void print();
};

#endif
```

5.5 Файл main.cpp

Листинг 5 – main.cpp

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

int main()
{
   int a, b;
   cin >> a >> b;
   cl_child obj1(a, b);
   obj1.cl_parent::print();
   cout << endl;
   obj1.print();
   cout << endl;
   if (a > 0)
   {
      obj1.set(a+1, b + 1);
      obj1.cl_parent::set(a-1, b-1);
      obj1.print();
      cout << endl;
      obj1.cl_parent::print();
      cout << endl;
   }
   else
   {
      obj1.set(a-1, b - 1);
      obj1.cl_parent::set(a+1, b+1);
      obj1.cl_parent::print();
      cout << endl;
      obj1.print();
   }
   return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 9.

Таблица 9 – 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

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).