**ЛАБОРАТОРНАЯ РАБОТА №3**

**«ИССЛЕДОВАНИЕ МЕХАНИЗМА МНОЖЕСТВЕННОГО**

**НАСЛЕДОВАНИЯ»**

**Цель работы**: Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизма множественного наследования.

**Вариант задания**

Описать интерфейс ввода-вывода. Описать иерархию классов, заданную по варианту. Для каждого класса описать конструкторы и деструктор (понеобходимости), функции ввода и вывода значений полей. В каждом классе должны присутствовать минимум одно уникальное поле и один уникальный метод. Проиллюстрировать корректную работу механизма множественного наследования — для этого создать объекты базовых классов и заполнить их поля данными, вывести на печать. Создать объект класса-наследника, его поля заполнить значениями соответствующих полей базовых классов. Вывести на печать данные полученного объекта.

Вариант 11

Базовые классы: Бумага (цвет, фактура, производитель), Дата (день, месяц, год).Класс-наследник: Календарь.

**2. Код программы на языке C++**

#include <iostream>

#include <string>

using namespace std;

class paper

{

string color, factur, manufacture;

public:

paper(string col, string fact, string manufact) : color(col), factur(fact), manufacture(manufact) {}

string toString() {

return "Информация о бумаге : Производство " + manufacture + " Фактура : " + factur + "Цвет :" + color+".";

}

};

class date {

int d, m, y;

public:

date(int d,int m,int y) : d(d),m(m),y(y){}

string toString() {

return to\_string(d) + "." + to\_string(m) + "." + to\_string(y);

}

};

class calendar : public paper, public date {

public:

calendar(string col, string fact, string manufact, int d, int m, int y) : paper(col, fact, manufact), date(d, m, y) {}

calendar(paper obj1, date obj2) :paper(obj1), date(obj2) {}

string toString() {

return paper::toString() +"\t"+ date::toString();

}

};

int main()

{

system("chcp 1251");

paper лист("Красный", "Глянец", "Крым бумага производство");

date дата(3, 9, 2001);

calendar календарь(лист, дата);

cout << лист.toString() << endl;

cout << дата.toString() << endl;

cout << календарь.toString() << endl;

}

**3. Тестирование и отладка**

Для тестирования данной программы, сами объекты класса инициализируются в основном коде программы, с такими данными. Лист : ("Красный", "Глянец", "Крым бумага производство") , Дата : дата(3, 9, 2001). Далее создается третий объект, который наследует полностью первый и второй , включая данные из них.

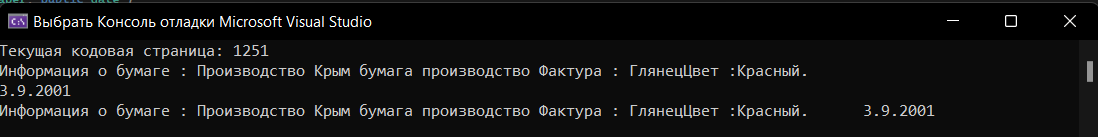


Рисунок 1 – Результат выполнения программы.

В результате тестирования, видно, что происходит множественное наследование, тоесть полиморфизм.

**Вывод**

При выполнении данной лабораторной работы были приобретены практических навыков при написании объектно-ориентированных программ с использованием механизма множественного наследования.