МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа № 5**

по дисциплине: Объектно-ориентированное программирование

тема: «Классы, виды отношений. Наследование»

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверил:

асс. Черников Сергей Викторович

Белгород 2024г.

**Лабораторная работа №5**

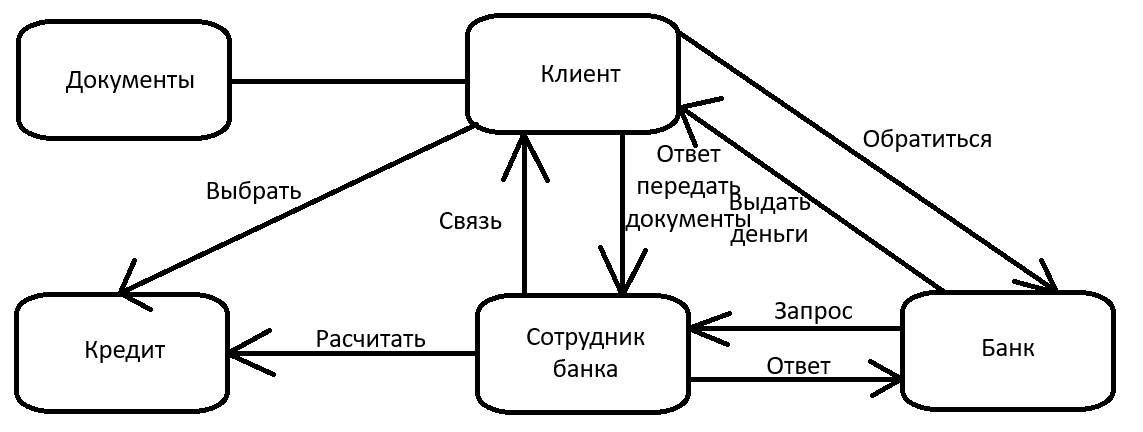
**«Классы, виды отношений. Наследование»**

**Цель работы:** получение теоретических знаний в области разработки классов, получение практических навыков реализаций классов и отношений между ними.

**Вариант 9**

**Задание 1**

Выполнить построение объектной модели заданной предметной области: Система кредитования граждан.



**Задание 2**

Разработать диаграмму классов для описанной объектной модели, и реализовать предложенные классы

1. Создать абстрактный класс Клиент с методами, позволяющими вывести на экран информацию о клиентах банка, а также определить соответствие клиента критерию поиска.

2. Создать производные классы: Вкладчик (фамилия, дата открытия вклада, размер вклада, процент по вкладу), Кредитор (фамилия, дата выдачи кредита, размер кредита, процент по кредиту, остаток долга), Организация (название, дата открытия счёта, номер счёта, сумма на счету) со своими методами вывода информации на экран, и определения соответствия дате (открытия вклада, выдаче кредита, открытия счёта).

3. Создать базу (массив) из n клиентов, вывести полную информацию из базы на экран, а также организовать поиск клиентов, начавших сотрудничать с банком в заданную дату.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Код программы:

#include <iostream>  
#include <string>  
#include <windows.h>  
  
using namespace std;  
  
class Client {  
public:  
 virtual void print\_inf() = 0;  
  
 virtual void search\_client(string) = 0;  
};  
  
class Creditor : public Client {  
 string surname;  
 string date\_give\_credit;  
 double size\_credit;  
 double perecent\_credit;  
 double balance\_credit;  
public:  
 Creditor(string surname, string date\_give\_credit, double size\_credit, double perecent\_credit,  
 double balance\_credit) {  
 this->surname = surname;  
 this->date\_give\_credit = date\_give\_credit;  
 this->size\_credit = size\_credit;  
 this->perecent\_credit = perecent\_credit;  
 this->balance\_credit = balance\_credit;  
 };  
  
 void print\_inf() {  
 cout << "Фамилия " << surname << endl;  
 cout << "Дата взятия кредита " << date\_give\_credit << endl;  
 cout << "Размер кредита " << size\_credit << endl;  
 cout << "Процент по кредиту " << perecent\_credit << endl;  
 cout << "Остаток по кредиту " << balance\_credit << endl;  
 cout << endl << "-------------------------------------------" << endl;  
 }  
  
 void search\_client(string date) {  
 if (date == date\_give\_credit) {  
 cout << "Клиент " << surname << " взял кредит в банке в этот: " << date\_give\_credit << " день" << endl;  
 } else {  
 //cout << "В этот день ничего не произошло";  
 }  
 }  
 //~Creditor();  
};  
  
class Contributor : public Client {  
 string surname;  
 string date\_open\_contribution;  
 double size\_contribution;  
 double perecent\_contribution;  
public:  
 Contributor(string surname, string date\_open\_contribution, double size\_contribution, double perecent\_contribution) {  
 this->surname = surname;  
 this->date\_open\_contribution = date\_open\_contribution;  
 this->size\_contribution = size\_contribution;  
 this->perecent\_contribution = perecent\_contribution;  
 };  
  
 void print\_inf() {  
 cout << "Фамилия " << surname << endl;  
 cout << "Дата открытия вклада " << date\_open\_contribution << endl;  
 cout << "Размер вклада " << size\_contribution << endl;  
 cout << "Процент по вкладу " << perecent\_contribution << endl;  
 cout << endl << "-------------------------------------------" << endl;  
 }  
  
 void search\_client(string date) {  
 if (date == date\_open\_contribution) {  
 cout << "Клиент " << surname << " положил деньги в банке в этот: " << date\_open\_contribution << " день"  
 << endl;  
 } else {  
 // cout << "В этот день ничего не произошло";  
 }  
 }  
 //~Contributor();  
};  
  
class Organization : public Client {  
 string name;  
 string date\_open\_score;  
 int namper\_score;  
 double money\_in\_scor;  
public:  
 Organization(string name, string date\_open\_score, int namper\_score, double money\_in\_scor) {  
 this->name = name;  
 this->date\_open\_score = date\_open\_score;  
 this->namper\_score = namper\_score;  
 this->money\_in\_scor = money\_in\_scor;  
 };  
  
 void print\_inf() {  
 cout << "Название организации " << name << endl;  
 cout << "Дата открытия счета " << date\_open\_score << endl;  
 cout << "Номер счета " << namper\_score << endl;  
 cout << "Денег на счете " << money\_in\_scor << endl;  
 cout << endl << "-------------------------------------------" << endl;  
 }  
  
 void search\_client(string date) {  
 if (date == date\_open\_score) {  
 cout << "Организация " << name << " открыла счет в банке в этот: " << date\_open\_score << " день" << endl;  
 } else {  
 // cout << "В этот день ничего не произошло";  
 }  
 }  
 //~Organization();  
};  
  
int main() {  
 SetConsoleCP(1251);  
 SetConsoleOutputCP(CP\_UTF8);  
 setlocale(LC\_ALL, "Russian");  
  
 const int N = 6;  
 Client \*\*arr = new Client \*[N];  
 string Date;  
 arr[0] = new Organization("Apple", "24.05.2019", 45785468, 11045455544544);  
 arr[1] = new Organization("Samsung", "22.11.2021", 43785788, 412245);  
 arr[2] = new Contributor("LG", "12.11.2022", 10000.5, 7.5);  
 arr[3] = new Contributor("Xiaomi", "18.01.2023", 10000000.5, 6.5);  
 arr[4] = new Creditor("Sven", "12.11.2015", 5110000, 6.5, 56200);  
 arr[5] = new Creditor("Toshiba", "22.10.2018", 511000000, 6.5, 1256200);  
 for (size\_t i = 0; i < N; i++) {  
 arr[i]->print\_inf();  
 }  
 cout << "Введите дату для поиска" << endl;  
 cin >> Date;  
 for (size\_t i = 0; i < N; i++) {  
 arr[i]->search\_client(Date);  
 }  
 for (size\_t i = 0; i < N; i++) {  
 delete arr[i];  
 }  
 delete arr;  
 return 0;  
}

Результат работы программы:

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Вывод:** в ходе лабораторной работы получены теоретические знания в области разработки классов, получены практические навыки реализаций классов и отношений между ними.