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-233

Ситников Алексей Павлович

Проверил:

Белгород 2025 г.

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

**Вариант** ((13 + 5)%10) + 1 = **9.**

**Задание 1:**

Объектная модель: «система кредитования граждан»

Изображение выглядит как текст, диаграмма, План, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Диаграмма классов:

Изображение выглядит как текст, Шрифт, число, документ

Контент, сгенерированный ИИ, может содержать ошибки.



**Задание 2:**

Интерфейс:

#ifndef **UNTITLED9\_INTERFACE\_H**#define **UNTITLED9\_INTERFACE\_H**#include <iostream>  
  
class Client{  
public:  
 virtual void printInformation() = 0;  
  
 virtual bool compareDate(const std::string& date) = 0;  
};  
  
class Investor :public Client{  
private:  
 std::string surname;  
 std::string dateInvest;  
 int contribution;  
 double percent;  
public:  
 Investor(const std::string& name, const std::string& accountNumber, int contribution, double percent);  
 void printInformation() override;  
 bool compareDate(const std::string& date) override;  
  
};  
  
class Creditor :public Client{  
 std::string surname;  
 std::string dateLoan;  
 int loanAmount;  
 double percent;  
 int remainder;  
public:  
 Creditor(std::string surname ,std::string date, int loanAmount, double percent, int remainder);  
 void printInformation() override;  
 bool compareDate(const std::string& date) override;  
};  
  
class Organization :public Client{  
 std::string name;  
 std::string dateCheck;  
 int nomber;  
 int sum;  
public:  
 Organization(std::string name, std::string date, int nomber, int sum);  
 void printInformation() override;  
 bool compareDate(const std::string& date) override;  
};  
#endif //UNTITLED9\_INTERFACE\_H

Реализация:

#include "interface.h"  
  
Investor::Investor(const std::string& surname, const std::string& date, int contribution, double percent)  
 : surname(surname), dateInvest(date), contribution(contribution), percent(percent) {}  
  
void Investor::printInformation(){  
 std::cout **<<** "Фамилия: " **<<** surname **<<** ", дата открытия счёта: " **<<** dateInvest **<<** ", сумма вклада: " **<<** contribution **<<** ", процент: " **<<** percent **<<** std::endl;  
}  
  
bool Investor::compareDate(const std::string& date){  
 for(int i = 0; i < 10; i++){  
 if(date**[**i**]** != this->dateInvest**[**i**]**){  
 return false;  
 }  
 }  
 printInformation();  
 return true;  
}  
  
Creditor::Creditor(std::string surname ,std::string date, int loanAmount, double percent, int remainder)  
 : surname(surname), dateLoan(date), loanAmount(loanAmount), percent(percent), remainder(remainder){}  
  
void Creditor::printInformation(){  
 std::cout **<<** "Фамилия: " **<<** surname **<<** ", дата выдачи кредита: " **<<** dateLoan **<<** ", размер кредита: " **<<** loanAmount **<<** ", процент: " **<<** percent **<<** ", остаток: " **<<** remainder **<<** std::endl;  
}  
  
bool Creditor::compareDate(const std::string& date){  
 for(int i = 0; i < 10; i++){  
 if(date**[**i**]** != this->dateLoan**[**i**]**){  
 return false;  
 }  
 }  
 printInformation();  
 return true;  
}  
  
  
Organization::Organization(std::string name, std::string date, int nomber, int sum)  
 : name(name), dateCheck(date), nomber(nomber), sum(sum) {}  
  
void Organization::printInformation(){  
 std::cout **<<** "Название: " **<<** name **<<** ", дата открытия счёта: " **<<** dateCheck **<<** ", номер счёта: " **<<** nomber **<<** ", сумма на счету: " **<<** sum **<<** std::endl;  
}  
  
bool Organization::compareDate(const std::string& date){  
 for(int i = 0; i < 10; i++){  
 if(date**[**i**]** != this->dateCheck**[**i**]**){  
 return false;  
 }  
 }  
 printInformation();  
 return true;  
}

Модуль main:

#include "interface.h"  
#include <vector>  
#include <windows.h>  
  
int main(){  
 SetConsoleOutputCP(**CP\_UTF8**);  
 std::vector<Client \*> data;  
  
 Investor investor1("Иванов", "22.07.2024", 1000000, 19.8);  
 Investor investor2("Сидоров", "29.07.2024", 350000, 15.3);  
 Investor investor3("Ситников", "25.08.2024", 10000000, 21.4);  
  
 Creditor creditor1("Плякин", "19.06.2024", 1000000, 16.7, 120000);  
 Creditor creditor2("Блинов", "25.08.2024", 500000, 18.3, 12000);  
 Creditor creditor3("Толмачев", "02.03.2024", 4000000, 14.5, 580000);  
  
 Organization org1("AMD", "23.01.2023", 345678923, 12000000);  
 Organization org2("Nvidia", "25.08.2024", 871238455, 30000000);  
 Organization org3("Intel", "13.02.2022", 125674523, 850000000);  
 data.push\_back(&investor1);  
 data.push\_back(&investor2);  
 data.push\_back(&investor3);  
 data.push\_back(&creditor1);  
 data.push\_back(&creditor2);  
 data.push\_back(&creditor3);  
 data.push\_back(&org1);  
 data.push\_back(&org2);  
 data.push\_back(&org3);  
  
 for(Client \* i : data){  
 i->printInformation();  
 }  
 std::cout **<<** std::endl **<<** "Клиенты обратившиеся в 25.08.2024\n";  
 for(Client \*i : data){  
 i->compareDate("25.08.2024");  
 }  
 return 0;  
}

Вывод программы:  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Вывод:** я научился создавать абстрактные классы, использовать виртуальные методы, и наследоваться.