**Лабораторна робота N7.**

**Тема:** Взаємодія між класи.

**Методичні вказівки**

**Завдання 1**

**Задача 1.4.** Розробити систему «Платежі». Клієнт має Рахунок в банку і Банківську карту (КК). Клієнт може оплатити Замовлення, зробити платіж на інший Рахунок, заблокувати КК і анулювати Рахунок. Адміністратор може заблокувати КК за перевищення платежу.

**Код:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class Account;**

**class BankCard;**

**class BankCard {**

**double limit;**

**bool blocked;**

**public:**

**BankCard(double limit) : limit(limit), blocked(false) {}**

**bool isBlocked() const { return blocked; }**

**void block() {**

**blocked = true;**

**cout << "Картка заблокована!\n";**

**}**

**bool pay(double amount) {**

**if (blocked) {**

**cout << "Картка заблокована. Оплата неможлива.\n";**

**return false;**

**}**

**if (amount > limit) {**

**cout << "Сума перевищує ліміт картки!\n";**

**return false;**

**}**

**cout << "Оплата успішна на суму: " << amount << " грн.\n";**

**return true;**

**}**

**friend class Administrator;**

**};**

**class Account {**

**double balance;**

**bool closed;**

**public:**

**Account(double initial) : balance(initial), closed(false) {}**

**void transfer(Account &to, double amount) {**

**if (closed) {**

**cout << "Рахунок анульовано. Переказ неможливий.\n";**

**return;**

**}**

**if (balance >= amount) {**

**balance -= amount;**

**to.balance += amount;**

**cout << "Переказ на " << amount << " грн успішний.\n";**

**} else {**

**cout << "Недостатньо коштів.\n";**

**}**

**}**

**void cancel() {**

**closed = true;**

**cout << "Рахунок анульовано.\n";**

**}**

**void showBalance() const {**

**cout << "Баланс: " << balance << " грн.\n";**

**}**

**};**

**class Client {**

**string name;**

**Account account;**

**BankCard card;**

**public:**

**Client(string name, double initial, double limit)**

**: name(name), account(initial), card(limit) {}**

**void payOrder(double amount) {**

**if (card.pay(amount)) {**

**account.transfer(account, amount); // move within own account**

**}**

**}**

**void sendPayment(Client &other, double amount) {**

**account.transfer(other.account, amount);**

**}**

**void blockCard() {**

**card.block();**

**}**

**void cancelAccount() {**

**account.cancel();**

**}**

**void showStatus() {**

**cout << "Клієнт: " << name << endl;**

**account.showBalance();**

**cout << "Картка " << (card.isBlocked() ? "заблокована" : "активна") << endl;**

**}**

**BankCard& getCard() {**

**return card;**

**}**

**};**

**class Administrator {**

**public:**

**void checkLimitAndBlock(BankCard &card, double amount) {**

**if (amount > card.limit) {**

**cout << "Адміністратор: перевищено ліміт! Блокуємо картку.\n";**

**card.block();**

**}**

**}**

**};**

**int main() {**

**Client alice("Аліса", 1000, 500);**

**Client bob("Боб", 500, 300);**

**Administrator admin;**

**alice.showStatus();**

**alice.payOrder(200);**

**alice.sendPayment(bob, 100);**

**admin.checkLimitAndBlock(alice.getCard(), 600); // перевищення ліміту**

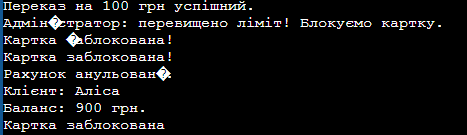
**alice.blockCard();**

**alice.cancelAccount();**

**alice.showStatus();**

**return 0;**

**}**

**Скрін:  
**