

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: «ООП»

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Буханов Дмитрий Геннадьевич

Харитонов Сергей Дмитриевич

Белгород 2023 г.

Тема: Классы, виды отношений. Наследование.

Вариант по списку: 23

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

Задание к работе: В соответствии с вариантом $((\text{номер по списку} + 5) \% 10) + 1$ выполнить построение объектной модели (использовать не менее 5 объектов) заданной предметной области (задание 1), разработать диаграмму классов для описанной объектной модели (не менее 7 классов), и реализовать предложенные классы (задание 2).

Вариант: 9 $((23 + 5) \% 10) + 1$

Задание 1: Система кредитования граждан.

Диаграмма объектов:

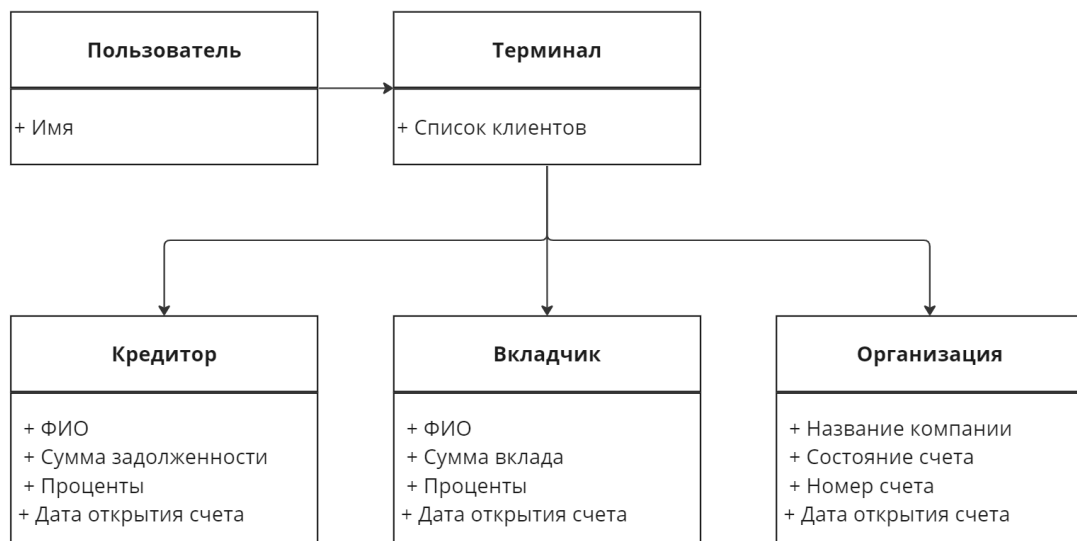
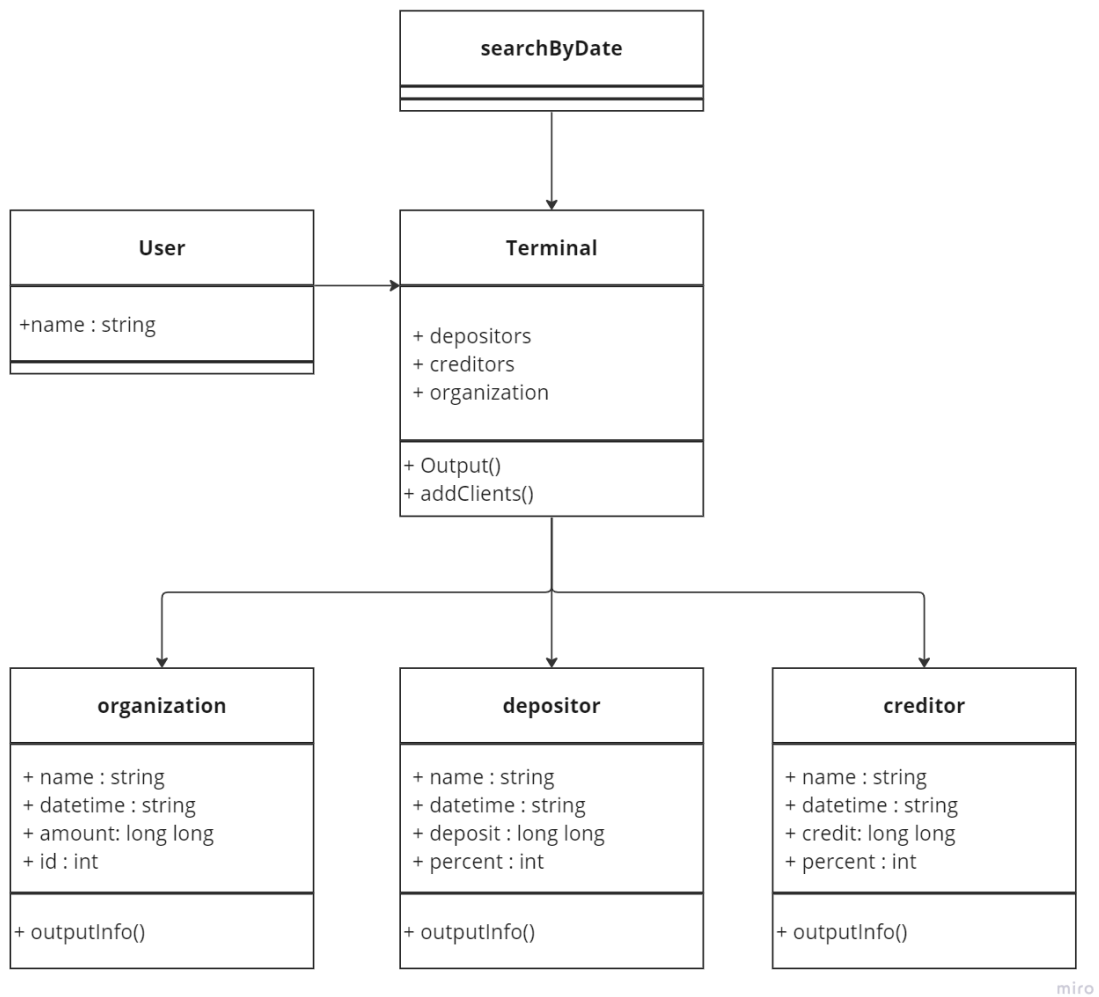


Диаграмма классов:



Задание 2

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

```
class Users {
    vector<depositor> depositors;
    vector<creditor> creditors;
    vector<organization> organizations;

    Users(
        vector<depositor> &depositors,
        vector<creditor> &creditors,
        vector<organization> &organizations
    ) {
        this->depositors = depositors;
        this->creditors = creditors;
        this->organizations = organizations;
    }
};
```

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)

```
class depositor {
public:
    string name;
    string datetime;
    long long deposit;
    int percent;

    depositor(string name, string datetime, long long deposit, int percent) {
        this->name = name;
        this->datetime = datetime;
        this->deposit = deposit;
        this->percent = percent;
    }

    void Output() const {
        cout << "name: " << name << "   datetime: " << datetime << "   deposit: "
              << deposit << "   percent: " << percent << endl;
    }
};

class creditor {
public:
    string name;
    string datetime;
    long long credit;
    int percent;

    creditor(string name, string datetime, long long credit, int percent) {
        this->name = name;
        this->datetime = datetime;
        this->credit = credit;
        this->percent = percent;
    }

    void Output() const {
        cout << "name: " << name << "   datetime: " << datetime << "   credit: "
              << credit << "   percent: " << percent << endl;
    }
};

class organization {
public:
    string name;
    string datetime;
    long long amount;
    int id;

    organization(string name, string datetime, long long amount, int id) {
```

```

this->name = name;
this->datetime = datetime;
this->amount = amount;
this->id = id;
}

void Output() const {
    cout << "name: " << name << " datetime: " << datetime << " amount: "
         << amount << " id: " << id << endl;
}
};

```

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.

```

class Users {
    vector<depositor> depositors;
    vector<creditor> creditors;
    vector<organization> organizations;

public:
    void Output() {
        cout << "depositors: \n";
        for (auto &i: depositors)
            i.Output();

        cout << "creditors: \n";
        for (auto &i: creditors)
            i.Output();

        cout << "organizations: \n";
        for (auto &i: organizations)
            i.Output();
    }

    void searchByDate(string datetime) {
        for (auto &i: depositors)
            if (i.datetime == datetime)
                i.Output();
        for (auto &i: creditors)
            if (i.datetime == datetime)
                i.Output();
        for (auto &i: organizations)
            if (i.datetime == datetime)
                i.Output();
    }

    Users(
        vector<depositor> &depositors,
        vector<creditor> &creditors,
        vector<organization> &organizations
    ) {
        this->depositors = depositors;
        this->creditors = creditors;
        this->organizations = organizations;
    }
};

```

```

int main() {
    SetConsoleOutputCP(CP_UTF8);
    vector< depositor > dep = {{"Владимир", "20.10.1999", 10000, 5},
                              {"Даниил", "01.01.2007", 312341, 7}};
    vector< creditor > cred = {{"Людмила", "08.07.2022", 10000, 5},
                              {"Даниил", "01.03.2023", 312341, 7}};
    vector< organization > org = {{"Бистро", "15.05.2020", 1231231, 1},
                                  {"000 \Как у бабушки\\"", "01.03.2023", 58378645734, 2},,};
    Users users(dep, cred, org);

    users.Output();
    cout << "\n\n Поиск по дате: 01.03.2023\n";
    users.searchByDate("01.03.2023");
    return 0;
}

```

Вывод: в ходе лабораторной работы мы получили теоретические знания в области разработки классов, получили практические навыков реализации классов и отношений между ними