

#include <bits/stdc++.h>

#define MOBILE 1

#define LANDLINE 2

#define NEW\_CONNECTION 1

#define BILL\_PAYMENT 2

#define BILL\_DETAILS 3

using namespace std;

class NegativeBill : public exception

{

    public:

    const char\* what() const throw(){

        return "Negative Fee Exception.Fee already paid!.Thank You\n";

    }

};

class NumberException : public exception

{

    public:

    const char\* what() const throw(){

        return "Connection Number Exception.This Number Do not exist.\nIf you think this is a mistake contact our customer care-'zohozohozoho'.Thank You\n";

    }

};

class ConnectionExistsException: public exception

{

    public:

    const char\* what() const throw(){

        return "Connection Exists Please Use existing connection or provice another connection number.Thank You\n";

    }

};

class User{

    public:

        string name,address,number;

        float bill=0;

        virtual void setBill()=0;

        virtual void payBill()=0;

        bool validateNumber(){

            if(number.size()!=10 && number.size()!=11){

                cout << "Enter valid Connection Number!\n";

                return false;

            }

            return true;

        }

        User(){

            cin.ignore ();

            cout << "Name:";getline(cin,name);

            cout << "Present Address: ";getline(cin,address);

            new\_number:

            cout << "Prefered Connection Number: ";cin >> number;

            if(!validateNumber())goto new\_number;

        }

};

class Mobile : public User

{

    float fee = 500.0;

    public:

    void setBill(){

        bill += fee;

    }

    void payBill(){

        if(bill-fee<0)throw NegativeBill();

        bill -= fee;

    }

};

class Landline : public User

{

    float fee = 750.0;

    public:

    void setBill(){

        bill += fee;

    }

    void payBill(){

        if(bill-fee<0)throw NegativeBill();

        bill -= fee;

    }

};

template <typename T>

class ZohoISP{

    public:

        unordered\_map<string, T\*> connections;

        bool isValid(string number){

            return connections.find(number)!=connections.end();

        }

        void addConnection(T\* connection)

        {

            if(isValid(connection->number)) throw ConnectionExistsException();

            connections[connection->number] = connection;

            cout << "\n\n   ------------New Connection Created!------------\n" << endl;

            updateBill(connection->number);

        }

        void payBill(string number)

        {

            if(!isValid(number))throw NumberException();

            connections[number]->PayBill();

            cout << "Payment Screen:::::::::::\n\n";

            cout << "Bill Paid For Number\n" << number << endl;

        }

        void updateBill(string number)

        {

            if(!isValid(number))throw NumberException();

            connections[number]->setBill();

            cout << "   Bill Updated for Number : " << number << " ,Bill :" << connections[number]->bill << endl;

        }

        void BillDetails(string number){

            if(!isValid(number))throw NumberException();

            cout << "Bill Details::::::::::\n\n";

            cout << "   Your Remaining Payment Fee :" <<connections[number]->bill << endl;

        }

};

int main(){

    cout << "<<<<<<<<<<<<===========Welcome To ZOHO Telecommunication Service=========>>>>>>>>>>\n";

    ZohoISP<Mobile> MobileManager;

    ZohoISP<Landline> LandLineManager;

    int choice = 0;

    while(true){

        try{

            int connection;

            connectionSelection:

            cout << "\n\n\n";

            cout << "Select you connection::::\n";

            cout << "1.) Mobile Connection\n" << "2.) Landline Connection\n";

            cout << "Choice:";cin >> connection;

            if(connection!=MOBILE && connection!=LANDLINE){

                cout << "Incorrect Response. Try Again!\n";

                goto connectionSelection;

            }

            connectionOperation:

            cout<<"\n\n";

            cout << "1.) New Connection\n" << "2.) Bill Payment\n" << "3.) Bill Details\n";

            cout << "Choice:";cin >> choice;

            string number;

            switch(choice){

                case NEW\_CONNECTION:

                    cout << "\n\n";

                    cout << "You Are Just One Step Closer To Enjoying Light Speed Internet Service\n";

                    new\_user:

                    User \*u;

                    if(connection==MOBILE){

                        Mobile \*user = new Mobile();

                        u = user;

                        try{

                            MobileManager.addConnection(user);

                        }

                        catch(ConnectionExistsException e){

                            cout << e.what();

                            delete user;

                            goto new\_user;

                        }

                    }

                    else{

                        Landline \*user = new Landline();

                        u = user;

                        try{

                            LandLineManager.addConnection(user);

                        }

                        catch(ConnectionExistsException e){

                            cout << e.what();

                            delete user;

                            goto new\_user;

                        }

                    }

                    cout << "New Connection have been successfully creater for user :" << u->name << " with " << ((connection==MOBILE)?"Mobile":"Landline" )<< " Connection :" << u->number << endl;

                    cout << "Have Nice Day\n";

                    break;

                case BILL\_DETAILS:

                    cout << "Connection Number :";

                    cin >> number;

                    if(connection==MOBILE){

                        try{

                            MobileManager.BillDetails(number);

                        }

                        catch(NumberException e){

                            cout << e.what() << endl;

                        }

                    }

                    else{

                        try{

                            LandLineManager.BillDetails(number);

                        }

                        catch(NumberException e){

                            cout << e.what() << endl;

                        }

                    }

                    break;

                case BILL\_PAYMENT:

                    cout << "Connection Number :";

                    cin >> number;

                    if(connection==MOBILE){

                        try{

                            MobileManager.BillDetails(number);

                        }

                        catch(NumberException e){

                            cout << e.what() << endl;

                        }

                    }

                    else{

                        try{

                            LandLineManager.BillDetails(number);

                        }

                        catch(NumberException e){

                            cout << e.what() << endl;

                        }

                    }

                    break;

                default:

                    cout << "Incorrect Response. Try Again!\n";

                    goto connectionOperation;

            }

        }

        catch(exception e){

            cout << e.what() << endl;

            cout << "Sorry for the incovnience caused. ZOHO ISP manager restarting......\n\n\n";

        }

        cout << "1. New Query\n2. Exit\n";

        cout << "Choice :";

        cin >> choice;

        if(choice!=1)break;

    }

    return 0;

}

