/// Andrew Souza

/// Comp 200 - Fall 2023

/// Inheritence Assignment

#include <iostream>

#include <vector>

#include <sstream>

using namespace std;

/// Problem 2

// Implement a base class Appointment and derived classes Doctor,Dentist,Therapy,and HairSalon.

class Appointment {

public:

Appointment() {

this->month = 1;

this->day = 1;

this->year = 2023;

this->description = "none";

this->cost = 0.00;

}

void SetDate(int month, int day, int year) {

this->month = month;

this->day = day;

this->year = year;

}

void SetDescription(string input) {

this->description = input;

}

void SetCost(double cost) {

this->cost = cost;

}

string GetDate() const {

ostringstream out;

out << this->month << "/";

out << this->day << "/";

out << this->year;

return out.str();

}

string GetDescription() const {

return this->description;

}

double GetCost() const {

return this->cost;

}

// Create a virtual function PrintReceipt to print out the cost of the appointment.

virtual void PrintReciept() const {

cout << "Appointment for " << GetDate() << endl;

cout << GetDescription() << endl;

cout << "$" << GetCost() << endl;

}

private:

// An appointment has a date (month,day,and year),a description, and a cost.

int month;

int day;

int year;

string description;

double cost;

};

// Each derived class has a SetDescription which takes in a char parameter

// a, b, and c each assign a hard-coded description, then calls SetCost with a hard-coded value for cost

// For the Doctor class, check if its an appointment for urgent care, checkup, or surgery.

class Doctor : public Appointment {

public:

void SetDescription(char input) {

if (input == 'a') {

Appointment::SetDescription("urgent care");

SetCost(350.00);

} else if (input == 'b') {

Appointment::SetDescription("checkup");

SetCost(60.00);

} else if (input == 'c') {

Appointment::SetDescription("surgery");

SetCost(3000.00);

} else {

cout << "Input a) for urgent care, b) for checkup, or c) for surgery: ";

cin >> input;

SetDescription(input);

}

}

virtual void PrintReciept() const override {

cout << "Doctor's Office Visit on " << GetDate() << endl;

cout << GetDescription() << endl;

cout << "$" << GetCost() << endl;

}

};

// For the Dentist class, check if its an appoint for cleanup, tooth-removal, or braces.

class Dentist : public Appointment {

public:

void SetDescription(char input) {

if (input == 'a') {

Appointment::SetDescription("cleanup");

SetCost(50.00);

} else if (input == 'b') {

Appointment::SetDescription("tooth removal");

SetCost(240.00);

} else if (input == 'c') {

Appointment::SetDescription("braces");

SetCost(960.00);

} else {

cout << "Input a) for cleanup, b) for tooth removal, or c) for braces: ";

cin >> input;

SetDescription(input);

}

}

virtual void PrintReciept() const override {

cout << "Dentist Office Visit on " << GetDate() << endl;

cout << GetDescription() << endl;

cout << "$" << GetCost() << endl;

}

};

// For the Therapy class, check if its for behavioral therapy, Anxiety, or Depression.

class Therapy : public Appointment {

public:

void SetDescription(char input) {

if (input == 'a') {

Appointment::SetDescription("behavioral therapy");

SetCost(60.00);

} else if (input == 'b') {

Appointment::SetDescription("Anxiety");

SetCost(80.00);

} else if (input == 'c') {

Appointment::SetDescription("Depression");

SetCost(100.00);

} else {

cout << "Input a) for behavioral therapy, b) for Anxiety, or c) for Depression: ";

cin >> input;

SetDescription(input);

}

}

virtual void PrintReciept() const override {

cout << "Therapist Visit on " << GetDate() << endl;

cout << GetDescription() << endl;

cout << "$" << GetCost() << endl;

}

};

// For the HairSalon class, check if its for hair dye, hair cut, or a perm.

class HairSalon : public Appointment {

public:

void SetDescription(char input) {

if (input == 'a') {

Appointment::SetDescription("hair dye");

SetCost(50.00);

} else if (input == 'b') {

Appointment::SetDescription("hair cut");

SetCost(30.00);

} else if (input == 'c') {

Appointment::SetDescription("perm");

SetCost(90.00);

} else {

cout << "Input a) for hair dye, b) for hair cut, or c) for perm: ";

cin >> input;

SetDescription(input);

}

}

virtual void PrintReciept() const override {

cout << "Hair Salon Visit on " << GetDate() << endl;

cout << GetDescription() << endl;

cout << "$" << GetCost() << endl;

}

};

int main() {

// Then fill a vector of Appointment\* with a mixture of appointments.

vector<Appointment\*> apptList;

char userOption;

string userDate;

cout << "Input a) for a generic appointment, b) for Doctor's Office, c) for Dentist Office, d) for Therapy, or e) for Hair Salon" << endl;

cout << "Input any other key to exit" << endl;

cin >> userOption;

while (userOption == 'a' || userOption == 'b' ||

userOption == 'c' || userOption == 'd' ||

userOption == 'e') {

char apptOption;

string description;

int month;

int day;

int year;

double cost;

char hold;

cout << "Input date (mm/dd/yyyy):" << endl;

cin >> month >> hold >> day >> hold >> year;

cin.ignore();

if (userOption == 'a') {

Appointment\* newAppt = new Appointment;

cout << "Input description:" << endl;

getline(cin, description);

cout << "Input cost:" << endl;

cin >> cost;

newAppt->SetDescription(description);

newAppt->SetDate(month, day, year);

newAppt->SetCost(cost);

apptList.push\_back(newAppt);

} else if(userOption == 'b') {

Doctor\* newAppt = new Doctor;

cout << "Input a) for urgent care, b) for checkup, or c) for surgery: ";

cin >> apptOption;

newAppt->SetDescription(apptOption);

newAppt->SetDate(month, day, year);

apptList.push\_back(newAppt);

} else if (userOption == 'c') {

Dentist\* newAppt = new Dentist;

cout << "Input a) for cleanup, b) for tooth removal, or c) for braces: ";

cin >> apptOption;

newAppt->SetDescription(apptOption);

newAppt->SetDate(month, day, year);

apptList.push\_back(newAppt);

} else if (userOption == 'd') {

Therapy\* newAppt = new Therapy;

cout << "Input a) for behavioral therapy, b) for Anxiety, or c) for Depression: ";

cin >> apptOption;

newAppt->SetDescription(apptOption);

newAppt->SetDate(month, day, year);

apptList.push\_back(newAppt);

} else if (userOption == 'e') {

HairSalon\* newAppt = new HairSalon;

cout << "Input a) for hair dye, b) for hair cut, or c) for perm: ";

cin >> apptOption;

newAppt->SetDescription(apptOption);

newAppt->SetDate(month, day, year);

apptList.push\_back(newAppt);

}

cout << "Input a) for a generic appointment, b) for Doctor's Office, c) for Dentist Office, d) for Therapy, or e) for Hair Salon" << endl;

cout << "Input any other key to exit" << endl;

cin >> userOption;

}

// Have the user enter a date and print out all appointments that happen on that date.

cout << "Enter a date (mm/dd/yyyy):" << endl;

cin >> userDate;

cout << "Appointments for date " << userDate << endl;

for (unsigned int i = 0; i < apptList.size(); i++) {

if (apptList.at(i)->GetDate() == userDate) {

cout << "--------------" << endl;

apptList.at(i)->PrintReciept();

cout << "--------------" << endl;

cout << endl;

}

}

for (unsigned int i = 0; i < apptList.size(); i++) {

delete apptList.at(i);

}

return 0;

}

