#pragma once

#include "stdafx.h"

#include <iostream>

#include <string>

#include <array>

#include <vector>

#include <conio.h>

#include <ctime>

using namespace std;

#ifndef Accounts

// Define Accounts

// ID class

class ID

{

public:

void OutputID(); // Outputs object ID

void GenerateID(\_\_int16 iPreviousIDPass); // Generates a new ID based on the previous objects ID incremented by 1

\_\_int16 ReturnID(); // Returns object ID for future objects to generate a new one off the previous

protected:

\_\_int16 iID; // ID variable

};

// ObjectNumber class

class ObjectNumber

{

public:

\_\_int16 GetNumberOfObjects(const string &sObjectTypePass); // Gets number of objects such as doctors, surgeries and receptionists for object creation

};

// Person class

class Person

{

public:

void GetDetails(\_\_int16 &iNumberPass); // Gets account username and password details

void DisplayDetails(\_\_int16 &iNumberPass); // Displays login details for test purposes

bool Login(const string &sNamePass, const string &sPasswordPass); // Checks username and password details to see if they are correct or incorrect

void LoginConfirmation(bool &bCheckPass); // Displays login confirmation

void IncorrectDetails(); // Displays incorrect details message

string ReturnName(); // Return name back to main for booking

protected:

string sName; // Username login details

string sPassword; // Password login details

};

// Receptionist class

class Receptionist : public Person, public ObjectNumber

{

public:

protected:

};

// Doctor class

class Doctor : public Person, public ID, public ObjectNumber

{

public:

Doctor(); // Doctor constructor

~Doctor(); // Doctor destructor

void GetDoctorDetails(\_\_int16 &iNumberPass); // Gets specialist area information

void ShowDoctorDetails(\_\_int16 &iNumberPass); // Show doctor details for booking options

void MinusAppointmentSlot(); // Removes appointment slot for doctor when a booking is created with the doctor

void AddAppointmentSlot(); // Adds back appointment slot for doctor when booking is completed or removed

bool& NoAvailableSlots(bool &bCheckPass); // Prevents bookings from being made if no available slots available

protected:

string sSpecialistArea; // Doctor's specialist area

const \_\_int16 iMaxAppointmentSlots = 10; // Assigns maximum number of appointment slots a doctor can have

\_\_int16 iAppointmentSlots; // Doctor's appointment slots

};

// Surgery class

class Surgery : public ObjectNumber

{

public:

Surgery(); // Surgery constructor

~Surgery(); // Surgery destructor

void GetSurgeryDetails(\_\_int16 &iNumberPass); // Gets name, location, doctor and available appointment slot information

void ShowSurgeryDetails(\_\_int16 &iNumberPass); // Show surgery name, location, doctor and available slot information for booking

void ShowSurgeryBookedDetails(); // Show surgery details for specific booked customer appointment

void MinusAvailableSlot(); // Removes available slot for surgery when a booking is created

void AddAvailableSlot(); // Adds available slot back to the surgery option when a booking has been deleted that was tied to it

bool& NoAvailableSlots(bool &bCheckPass); // Adds available slot for surgery when a booking is deleted

protected:

string sSurgeryName; // Surgery name

string sLocation; // Surgery location

string sDoctor; // Doctor who does the surgery

\_\_int16 iAvailableSlots; // Surgery available slots

};

// Customer class

class Customer : public Person, public ID

{

public:

Customer(); // Customer constructor

~Customer(); // Customer destructor

void GetCustomerDetails(\_\_int16 &iPreviousIDPass); // Get customer phone number, address and ailment details

void ShowAilment(\_\_int16 &iPreviousIDPass); // Shows ailment of the customer in booking section

string ReturnAilment(); // Returns customer ailment back to main for booking details

void ShowCustomerDetails(\_\_int16 &iNumberPass); // Shows all customer details

protected:

\_\_int32 iPhoneNumber; // Customer's phone number

string sAddress; // Customer's address

string sAilment; // Customers ailment

};

#endif