КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

\_\_\_Кафедра автоматизованих систем обробки інформації та управління\_\_\_

(повна назва кафедри)

**КУРСОВА РОБОТА**

|  |  |
| --- | --- |
| з | **Об’єктно-орієнтованого програмування** |

(назва дисципліни)

на тему:  *Реєстратура лікарні. Запис до лікарів на прийом*

Студентки 1 курсу ІС-02 групи

Спеціальність 126 Інформаційні системи та технології

Керівник старший викладач кафедри кафедри інформатики та програмної інженерії Халус О.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Дисципліна Структури даних та алгоритми

**Спеціальність 126 «Інформаційні системи та технології»**

Курс 1 Група ІС-02 Семестр 2 .

**ЗАВДАННЯ**

**на курсову роботу студента**

***Педь Олександри Вікторівни***

(прізвище, ім’я, по батькові)

|  |  |
| --- | --- |
| 1. Тема проекту (роботи) | **Реєстратура лікарні. Запис до лікарів на прийом** |

|  |  |
| --- | --- |
| 2. Строк здачі студентом закінченого проекту (роботи) | **18.05.2021** |

3. Вихідні дані до проекту (роботи)

**Варіант №9**

*Реєстратура надає дані стосовно наявності лікарів та розкладу прийому хворих. Хворим можливо записатись на прийом до лікаря, якщо є вільний час у розкладі лікаря. В реєстратурі ведуться картки відвідування хворими лікарні, в які записується час відвідання лікаря, діагноз та лікар, що його поставив*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) *1.Проектування об’єктно-орієнтованої моделі предметної галузі, визначивши необхідні для цього класи та їх структуру. 2.Демонстрування застосування всіх видів відношень при об’єктно-орієнтованому проектуванні.*

*3. Написання програми мовою С++, у якій реалізовано попередньо спроектовану об’єктно-орієнтовану модель*

5. Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Дата видачі завдання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зміст

ВСТУП……………………………………………………………………3

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ……………………………………..4

ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ……………………………5

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ..…………….8

РЕЗУЛЬТАТИ ТЕСТУВАННЯ………….………………………………9

ВИСНОВОК……………………………………………………………...12

ПРОГРАМНИЙ КОД……………………………………………………14

ЛІТЕРАТУРА…………………………………………………………….15

ВСТУП

Метою курсової роботи з дисципліни «Об’єктно - орієнтоване програмування» є:

1. Вивчити типи відношень між класами в ООП.

2. Спроектувати об’єктно - орієнтовану модель предметної галузі згідно з варіантом, визначивши необхідні для цього класи та їх структуру. При об’єктно - орієнтованому проектуванні продемонструвати застосування всіх видів відношень.

3. Написати програми мовами С++ та С#, у яких реалізувати попередньо спроектовану об’єктно - орієнтовану модель.

4. Предметну галузь реалізувати окремим проектом – динамічною бібліотекою (С#) чи окремими класами з відсутнім в них введенням - виведенням (С++).

5. Інтерфейс користувача, наприклад, введення\виведення з консолі, реалізовувати окремим проектом (С#) чи класами з операціями введення\виведення (С++). Код інтерфейсу має бути простим (демонструється використання класів предметної галузі шляхом створення об’єктів та їх застосування, вимагається лише нескладна перевірка коректності вводу, введення з консолі мінімальне).

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1. Програма має надавати можливість користувачу записатись на прийом до лікаря та подивитись свою картку.

Для того, щоб записатись на прийом, користувач повинен ввести свої облікові дані. Далі обрати лікаря до якого хоче записатись та дату (час, день та місяць) прийому.

При введенні своїх даних, користувач повинен мати можливість переглянути історію відвідування лікарні.

1. Необхідно щоб в базі даних програми зберігалися відомості про лікарів (графіки їх роботи), а також інформація про пацієнтів (ПІБ, номер телефону, дата та час відвідування лікаря, діагноз).

Програма має зберігати такі дані лікаря:

* Спеціальність
* Прізвище та ім’я
* Графік роботи
* Часи прийому

Програма має зберігати такі дані пацієнта:

* Прізвище та ім’я
* Номер телефону
* День та час прийому
* Діагноз

Користувач може вносити зміни у графік роботи лікаря, а саме: після запису на прийом, у графіку роботи лікаря цей день та час позначається як ‘зайнятий’.

ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Програма має класи ‘Find’, ‘People’, ‘Staff’, ‘Client’.

**Find - *абстрактний клас з абстрактними методами*:**

coutPerson() - *метод для виведення списку осіб*

**People - *клас, що зберігає в собі дані лікаря:***

string name - *прізвище та ім’я*

int\*\* Timetable - *розклад лікаря*

int day - *кількість робочих днів*

int Weekdays(int\*\* Days, int month) - *зберігає дані про вихідні дні*

int Delete(int\*\* Days, int\* Mi, int k) - *віднімає вихідні дні*

People() - *конструктор за замовчуванням*

People(string name) - *конструктор з параметрами*

string GetName() - *вивід прізвища та імені*

int GetDay() - *вивід кількості робочих днів*

void SetName(string name) - *задання прізвища та імені*

void NewTimetable(int month, int shift) - *заповнює масив робочих днів(1…31) та розклад*

void SetTimetable(int\*\* Days) - *задання розкладу*

int\*\* GetTimetable() - *вивід розкладу*

**Staff - *похідний клас від Find:***

People A - *об’єкт класу People*

People B - *об’єкт класу People*

People C - *об’єкт класу People*

int Ashift - *часи прийому*

int Bshift - *часи прийому*

int Cshift - *часи прийому*

Staff(string name1, string name2, string name3, int Ashift, int Bshift, int Cshift) - *конструктор з параметрами*

Staff(const Staff& other) - *конструктор копіювання*

int FindShift(int New, int month, int Doctor) - *шукає зміну вибраного лікаря*

void FindDay(int Doctor, int data) - вивід розкладу лікаря починаючи з поточної дати

bool CheckDay(int day) - *перевіряє щоб вибраний день був робочим у графіку лікаря*

string FindPerson(int choice) override - *перевизначений, шукає вибраного лікаря*

void coutPerson() override - *перевизначений, вивід лікарів*

void ChangeTimetable(int Doctor, int time, int data) - *змінює розклад*

bool FindDoctor(string name) - *шукає лікаря в базі*

**Client - *похідний клас від Find*:**

string name - *прізвище та ім’я користувача*

int number - *номер користувача*

string\*\* Cards - *база всіх пацієнтів*

Client() - *конструктор за замовчуванням*

Client(string name, int number) ) - *конструктор з параметрами*

void SetClient(string name, int number) - *задання прізвища, імені та номеру телефону*

string GetClientName() - *вивід прізвища імені*

int GetClientNum() - *вивід номеру телефону*

string FindPerson(int choice) override - *перевизначений, шукає спеціальність лікаря щоб задати діагноз*

void NewClient(int New, string doctor, int data, int month, int time) - *додає нового пацієнта в базу даних*

void coutPerson() override - *перевизначений, вивід картки пацієнта*

bool FindClient(string name) - *шукає пацієнтів лікаря*

void Diagnosis(string name) - *ввід діагнозу лікарем*

1. По-перше, я створюю об'єкти класу Staff та задаю потрібні дані (прізвище та ім’я лікаря, час роботи). Також створюю об’єкт класу Client, але поки він не використовується.

Користувач має вибрати:

* Пацієнт
* Лікар

1. Якщо вибирається роль пацієнта, то далі, програма надає можливість користувачеві вибрати дію:

* Вихід
* Зареєструватись на прийом до лiкаря
* Переглянути анкету
  1. Якщо вибрати перший варіант, то програма зупиниться.

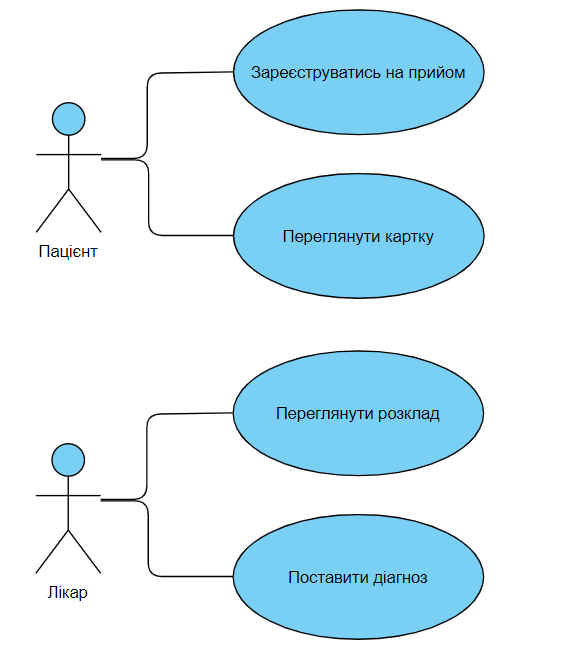
1.2 Обравши ‘Зареєструватись на прийом до лiкаря, користувач повинен ввести свої дані: прізвище, ім’я, номер телефону. Ці дані передаються у об’єкт класу Client за допомогою функції SetClient(name, number). Далі користувач має вибрати до якого спеціаліста він хоче потрапити на прийом. Після цього викликається функція coutPerson() і він бачить список лікарів цієї цієї спеціальності які є в базі, вибирає того, к якому піде. Отримавши номер лікаря у списку, викликається Lore.FindPerson(chDoctor) і знаходить його прізвище та ім’я. Далі користувач має ввести бажану дату прийому. За допомогою функції Lore.FindShift(New1, month, chDoctor) заповнюється розклад (якщо це перший запис на відвідування) та визначається час роботи. Наступним кроком викликається функція Lore.FindDay(chDoctor, data) яка приймає дату введену клієнтом та виводить графік роботи лікаря починаючи з бажаної дати. Користувач має вибрати зручний для нього день та час прийому. Наступним кроком викликається функція ChangeTimetable(chDoctor, time, day) яка помічає вибраний день та час прийому як ‘зайнятий’. Далі всі отримані дані передаються в NewClient(New, Doctor, chSpecialist, day, month, time) і новий пацієнт з даними про приймаючого лікаря, день та час прийому передаються в базу (картку пацієнта). Останнім кроком є заключне виведення всіх даних.

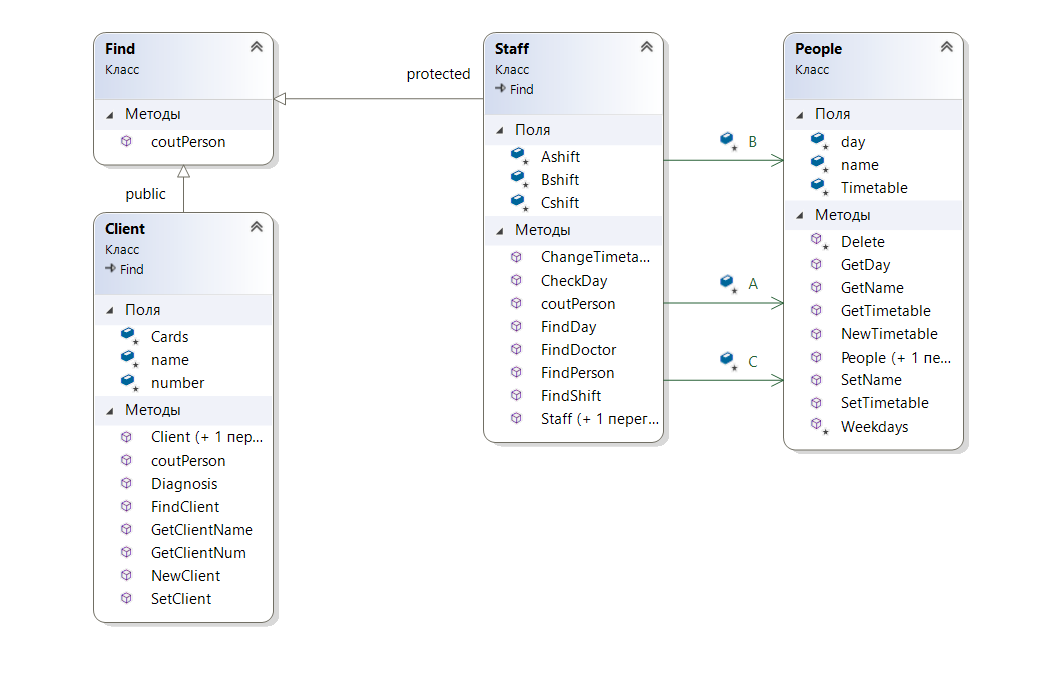
* 1. При виборі останньої дії, визивається функція coutPerson() для об’єкта класу Client. Користувач повинен ввести своє прізвище та ім’я, а далі відбувається пошук в базі пацієнтів. На екрані з’являється вся інформація щодо відвідування лікаря цим пацієнтом.

1. Якщо користувач вибирає роль лікаря, то він повинен ввести свої дані (прізвище та ім’я) та пароль. Далі, він має вибрати дію:

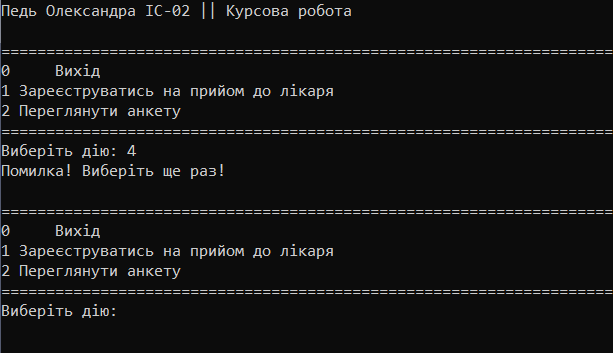
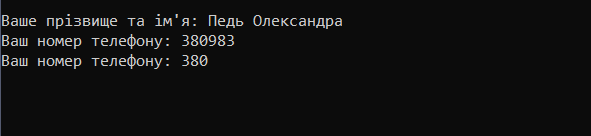
* Вихiд
* Переглянути розклад
* Поставити діагноз
  1. Якщо вибрати перший варіант, то програма зупиниться.
  2. Далі лікар повинен ввести бажаний день. За допомогою функції FindDay(name, data) шукаємо розклад лікаря в базі. Далі викликається функція, що виводить пацієнтів FindClient(name).
  3. Викликається функція Diagnosis(name) і лікар повинен ввести прізвище та ім’я пацієнта. Програма шукає анкету та надає можливість ввести діагноз.

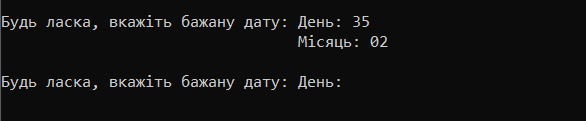
**UseCase діаграма**

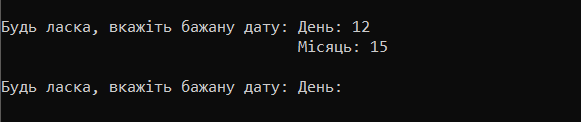


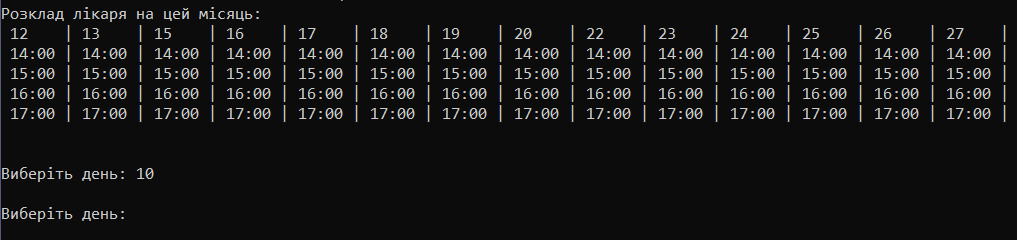
ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

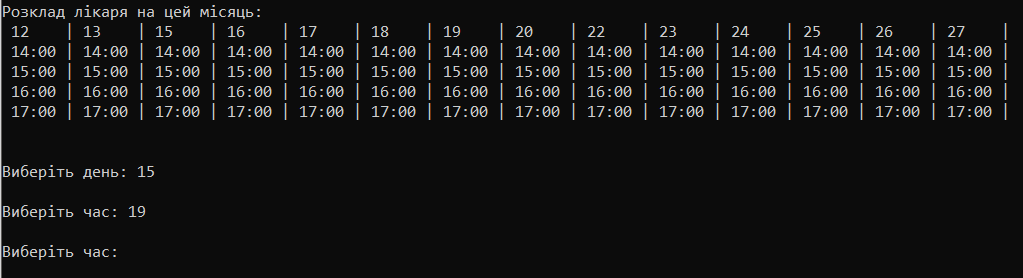
РЕЗУЛЬТАТИ ТЕСТУВАННЯ

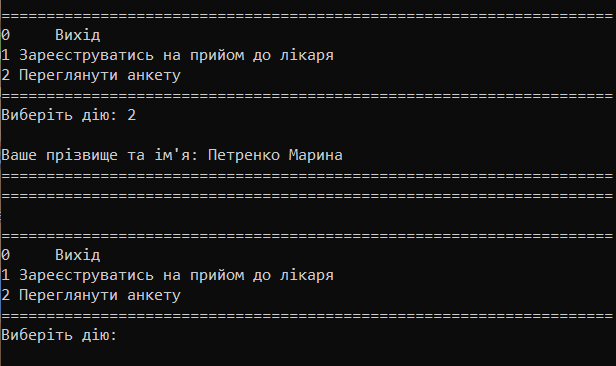
1. При введені неіснуючого варіанту дій програма виводить на екран надпис ‘Зареєструватись на прийом до лiкаря та пропонує вибрати ще раз.
2. Якщо ввести недостатню кількість цифр у номер телефону, то рядок вводу з’явиться ще раз.

1. Якщо при виборі спеціальності лікаря натиснути неіснуючий варіант, то рядок вводу повториться.
2. Якщо ввести неправильний варіант при виборі лікаря, то рядок з’явиться повторно.
3. Якщо бажану дату (день чи місяць) ввести некоректно, то рядок повториться.

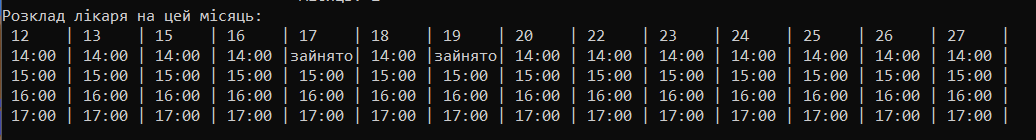


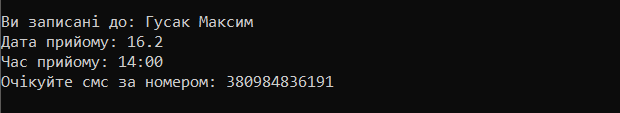
1. Якщо при виборі дня у графіку лікаря, вибрати не коректну дату\час, то поле з вибором з’явиться ще раз.

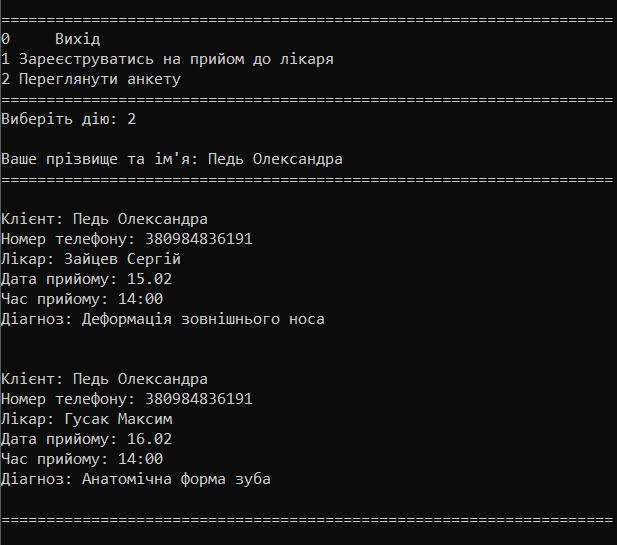


1. Якщо при виборі дії ‘Переглянути анкету’ ввести прізвище та ім2я пацієнта, чиєї картки немає, то меню вибору дії з’явиться знов.

ВИСНОВОК

Згідно з завданням, програма надає дані стосовно розкладу роботи лікаря та прийому хворих.

У хворого є можливість записатись на прийом до будь-якого лікаря у вільний час.

Вказуючи прізвище та ім’я, пацієнт може подивитись свою картку , в яку записується час відвідання лікаря, діагноз та лікар, що його поставив.

ПРОГРАМНИЙ КОД

<https://github.com/Sasha4907/coursework>

ЛІТЕРАТУРА

1. Объектно-ориентированное программирование в С++. Р. Лафоре <https://codernet.ru/books/c_plus/obektno-orientirovannoe_programmirovanie_v_c/>
2. Герберт Шилдт: С++ базовый курс

<https://www.bsuir.by/m/12_100229_1_98220.pdf>

1. Конспект лекцій