

Przychodnia

0. Stwórz projekt – aplikacja konsolowa lub WPF (przemyśl wybór, bo zmiana może być czasochłonna).

1. Stwórz abstrakcyjną klasę *Osoba*. W tej klasie wykonaj następujące czynności:

a) dodaj pole `imieNazwisko` typu `string` z mod. `protected`

b) dodaj konstruktor z parametrem typu `string`, pobrany parametr należy ustawić jako pole `imieNazwisko`.

2. Stwórz klasę *Pacjent* dziedziczącą z klasy *Osoba*. W klasie potomnej wykonaj następujące czynności:

a) dodaj prywatne pola `wiek` (typu `int`) i `choroba` (typu `string`)

b) dodaj konstruktor z trzema parametrami kolejno typów (`string`, `int`, `string`), pobrane parametry należy ustawić jako odpowiednia pola: `imieNazwisko`, `wiek`, `choroba`;

c) przesłoń metodę `ToString()` tak, aby zwracała informacje o Pacjencie np.

Pacjent, imię i nazwisko: Jan Kowalski, wiek 50, choroba: grypa.

3. Stwórz klasę *Lekarz* dziedziczącą z klasy *Osoba*. W klasie potomnej wykonaj następujące czynności:

a) dodaj prywatne pole `specjalnosc` typu `string`,

b) dodaj konstruktor z dwoma parametrami typu `string`, pobrane parametry należy ustawić kolejno jako pola `imieNazwisko`, `specjalnosc`

c) przesłoń metodę `ToString()` tak, aby zwracała informacje o Lekarzu np.:

Lekarz, imię i nazwisko: Anna Nowak, specjalność: internista.

4. Stwórz interfejs *IPrzychodnia* i dodaj w nim deklarację następujących metod:

`UstawLekarza(string imieN, string specjalnosc)` – typu `void`

`ZapiszDoLekarza(string imieN, int wiek, string choroba)` – typu `void`

`WykonajPorade()` – typ zwracany `string`

`WykonajBadanie()` – typ zwracany `string`

`CzasOczekiwania()` – typ zwracany `int`

`GenerujRaport()` – typu `void`

5. Stwórz klasę Przychodnia. Wykonaj w niej następujące czynności:

a) dodaj prywatne pole lekarz typu Lekarz,

b) dodaj prywatne pole pacjenci typu Stack<Pacjent> (generyczny stos odpowiadający kolejce pacjentów czekających na lekarza, celowo jest to stos a nie kolejka, proszę tego nie zmieniać)

c) do klasy podepnij interfejs IPrzychodnia

d) dodaj następujące metody

- UstawLekarza(string, string) – ustawia parametry dla pola lekarz,

- ZapiszDoLekarza(string, int, string) – dodaje do stosu pacjenci obiekt typu Pacjent, parametry należy ustawić jako pola imieNazwisko, wiek, choroba,

- WykonajPoradę() – metoda usuwa i zwraca ostatni element ze stosu pacjenci, we zwracanym stringu należy zrobić dopisek o wykonanej poradzie, przykładowy zwracany string:

Wykonano poradę! Pacjent, imię i nazwisko: Jan Kowalski, wiek 50, choroba: grypa.

- WykonajBadanie() – metoda zwraca (bez usuwania) ostatni element ze stosu pacjenci, we zwracanym string należy zrobić dopisek o wykonanym badaniu, przykładowo string:

Wykonano badanie! Pacjent, imię i nazwisko: Jan Nowaczyk, wiek 44, choroba: grypa.

- CzasOczekiwania() – zwraca liczbę typu int, która jest częścią całkowitą liczby elementów w stosie pacjenci podzielonej przez 5 np. jeśli w kolejce jest 10 osób, to $10/5=2$ i tyle jest zwracane; jeśli zaś jest w kolejce 16 osób, to $16/5 = 3,2$ to zwracane jest 3,

- przesłoń metodę ToString(), tak aby zwracała w stringu w kolejnych wierszach informację o lekarzu, danych pacjentów ze stosu pacjenci oraz czas oczekiwania np.

Lekarz: Jan Nowak, specjalność : okulista

Pacjenci oczekujący:

- Pacjent: Antoni Kowalski, wiek 33, choroba: zaćma,
- Pacjent: Zofia Nowak, wiek 44, choroba: jaskra.

Czas oczekiwania: 0.

- GenerujRaport() – informacje pobrane poprzez metodę ToString() w tej klasie mają być zapisane w pliku tekstowym, nazwa pliku obowiązkowo ma mieć format: RaportddmmyyyyGGMM.txt, gdzie GGMM to godzina zapisana w formacie 24-godzinnym, ddmmYYYY – data, wartości mają być pobrane aktualne z systemu, w nazwie pliku nie ma być spacji czy dodatkowych znaków,
- dodaj metodę CzyLekarz() bez parametru, typ zwracany bool, zwraca ona true jeśli pole lekarz w klasie nie jest puste (różne od null); zwraca false w przeciwnym wypadku.

6. Zrób aplikację WPF lub konsolową do przetestowania metod i klas z punktów 1-5.

Logika aplikacji:

- przychodnia ma zawsze jednego lekarza, ustawienie danych lekarza powinno być wywoływane jako pierwsze, ma być sprawdzone czy pobrane stringi nie są puste, dopiero wtedy ma być opcja na dodawanie pacjentów do stosu pacjenci
- zapisywanie pacjentów ma sprawdzać czy wprowadzone stringi nie są puste i pole wiek ma być liczbą dodatnią mniejszą niż 109 (jeśli jest błąd ma być komunikat dla użytkownika)
- wykonanie porady lub badania powinno wyświetlić komunikat ma MessageBoxie lub konsoli używając odpowiedniej metody
- generowanie raportu powinno poza wywołaniem metody wyświetlić komunikat o pomyślnym zakończeniu tego procesu

Punktacja:

Polecenie 1-4 – po 1 pkt

Polecenie 5 – 2pkt

Polecenie 6:

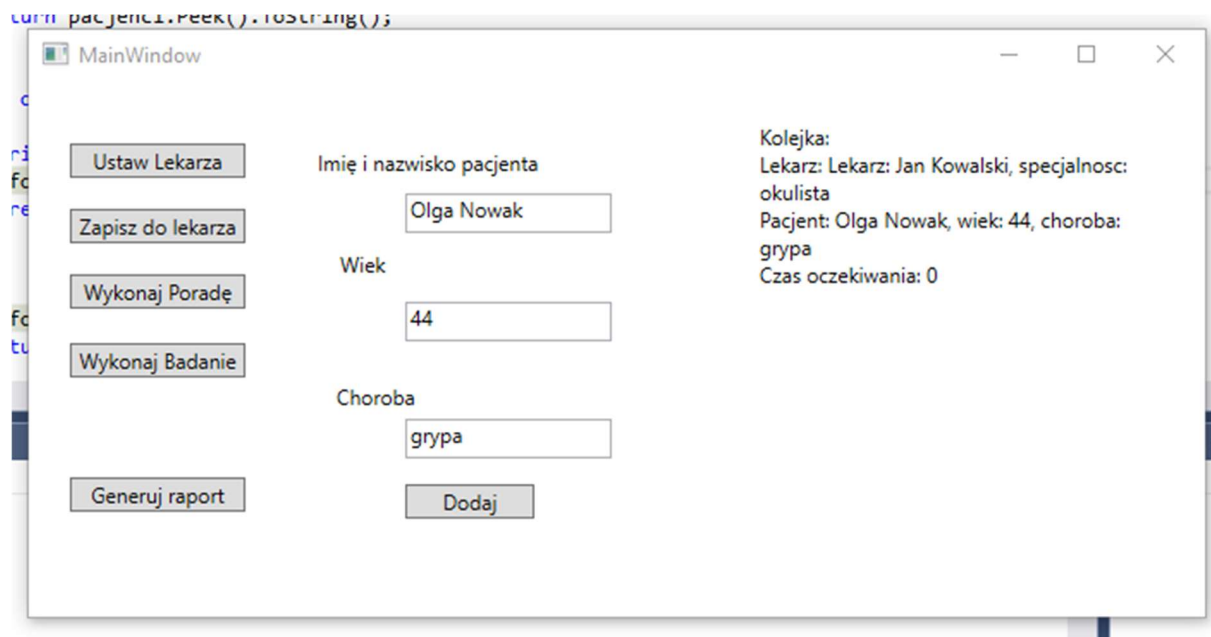
+1 pkt za poprawne działanie kontrolek (tylko dla WPF)

+1 pkt za poprawny algorytm aplikacji przechodzenia do kolejnych etapów (tylko konsola)

+ 1 punkt za obsługę możliwych wyjątków przy zmianie typów i usuwaniu elementów z pustego stosu

+ 2 za logikę aplikacji

Przykładowy screen WPF:



Przykładowy screen (konsola):

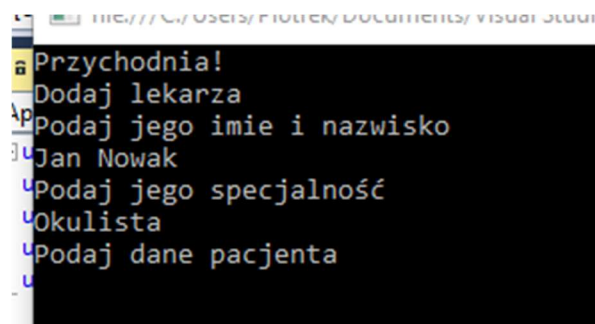


Diagram UML:

