

Zad 1.

- ❖ (1p). Zdefiniuj klasę abstrakcyjną **Osoba**, w której są następujące składowe:
 - nazwisko, pesel;
 - `Osoba()` – konstruktor tworzący osobę o pustych tekstach (`nazwisko=""`, ...),
 - `Osoba(String nazwisko, String pesel)` – konstruktor tworzący osobę o podanym nazwisku i peselu,
 - Metody: `toString()` – zwraca jako napis nazwisko i pesel.
 - Metody abstrakcyjne:
 - `double Oblicz()` – obliczającą pewną kwotę
- ❖ (2.5p). Zdefiniuj klasę **Lekarz** dziedziczącą z klasy **Osoba**, a w niej:
 - Składowe:
 - `String s` – specjalizacja,
 - `int stawka` – cena wizyty u lekarza,
 - `int t[5]` – tablica, w której zapisana jest liczba pacjentów przyjętych przez lekarza, od poniedz. do piątku,
 - `Lekarz()` – konstruktor tworzący lekarza o specjalizacji „brak”, nazwisku „” i `peselu=""`, `stawka=0`, `t=null`;
 - `Lekarz(String nazwisko, String pesel, String spec, int stawka)` – konstruktor tworzący lekarza o podanym nazwisku, peselu, specjalizacji, stawce i tablicy przyjęć wypełnionej liczbami losowymi [20 .. 30]
 - Metody:
 - `Oblicz()` – metoda zwracająca wartość wyliczoną jako `stawka * liczba pacjentów przyjętych przez lekarza w jednym tygodniu`,
 - `toString()` – metoda zwracająca napis „Lekarz:...(nazwisko i pesel) [specjalizacja=..., stawka=...]”, gdzie w miejsce kropek mają pojawić się właściwe dla danego obiektu wartości,
 - `wyświetlLiczbęPacjentów()` – metoda wyświetlająca liczbę pacjentów przyjętych przez lekarza od poniedziałku do piątku w poszczególne dni tygodnia,
 - `jestInternista` – metoda zwracająca wartość *true*, gdy dany lekarz jest internistą oraz *false* jeśli nie jest
- ❖ (1p). Zdefiniuj klasę **Wizyta**, w której są następujące składowe:
 - `int kodChoroby`, `int cenaWizyty`, atrybuty logiczne określające czy wystawiono zwolnienie i czy wystawiono receptę,
 - `Wizyta()` – konstruktor tworzący wizytę o kodzie =0, cenie=0 i atrybutach logicznych ustawionych na *false*,
 - `Wizyta(int kod, int cena, boolean zwolnienie, boolean recepta)` – konstruktor tworzący wizytę o podanym kodzie, cenie i atrybutach logicznych.
 - Metoda `toString()` – zwraca atrybuty tej klasy jako łańcuch,
- ❖ (2.5p) Zdefiniuj klasę **Pacjent** dziedziczącą z klasy **Osoba**, a w niej:
 - Składowe:
 - `n` - liczba wizyt, które odbył pacjent
 - tablica wizyt
 - `Pacjent()` – konstruktor tworzący pacjenta bez wizyt i nazwisku="" i `peselu=""`, `n=0`, `tablica=null`;
 - `Pacjent(String nazwisko, String pesel, int n)` – konstruktor tworzący pacjenta o podanym nazwisku, peselu i tablicy wizyt o rozmiarze `n`,
 - Metody:
 - `Oblicz()` – metoda zwracająca kwotę, którą zapłacił pacjent za wszystkie swoje wizyty,
 - `toString()` – metoda zwracająca napis „Pacjent:...(nazwisko i pesel:.. [n=...])”, gdzie w miejsce kropek mają pojawić się właściwe dla danego obiektu wartości,
 - `wyświetlWizyty()` – metoda wyświetlająca wszystkie wizyty pacjenta.
- ❖ (3p). Zdefiniuj klasę **Przychodnia**, w której są pola:
 - tablicę 20 obiektów klasy **Osoba**
 - `int n` – zawierającą liczbę faktycznie wpisanych osób do tablicy,
 - metodę `main`, w której wykonaj następujące operacje:
 - wstaw do tablicy kilku lekarzy oraz kilku pacjentów wykorzystując każdy z konstruktorów;
 - wyświetl wszystkie osoby w kolejności występowania w tablicy – osoby mają zostać wyświetlone za pomocą odpowiedniej metody `toString`;
 - wyświetl samych lekarzy oraz liczby pacjentów przyjętych przez nich,
 - wyświetl samych pacjentów oraz ich wizyty;
 - policz ilu jest internistów w przychodni
 - sprawdź, który z lekarzy najwięcej zarobił i wypisz jego specjalizację;