1. Utworzyć klasę Obywatel, która zawiera pola prywatne: dzień, miesiąc, rok typu integer oraz płeć typu char.
2. Napisać konstruktor, w którym inicjalizowane są powyższe dane.
3. Utworzyć co najmniej 7 obiektów z przykładowymi wartościami, w taki sposób, żeby 4 z nich miały 4 różne błędy. Prawidłowe dane:

* dzień=<1;31>
* miesiąc=<1;12>
* rok=<1900;2099>
* płeć={‘k’,’K’,’m’,’M’}

1. Zadeklarować wektor klasy Obywatel i wrzucić do niego wszystkie obiekty utworzone w punkcie 3.
2. Napisać metodę klasy Obywatel która sprawdza poprawność daty, w przypadku niespełnienia któregoś warunku wyrzucić wyjątek.
3. Napisać metodę zwracającą 0 lub 1 w zależności od płci podanej przez konstruktor, przy okazji wyrzucić wyjątek jeśli płeć jest nieprawidłowa.
4. Iterując po wektorze wyłapać wyjątki (switchem) w zależności czy wyjątek dotyczył płci czy daty. Wyświetlić odpowiedni komunikat o usunięciu obywatela oraz usunąć obiekt w wektorze.

Podpowiedź1: Przydatna funkcja: wektor.erase(pozycja\_wektora\_do\_usuniecia)

Podpowiedź2: Należy modyfikować iterator po usunięciu obiektu z wektora.

1. Napisać metodę klasy Obywatel zwracający wektor który zawiera pesel danego obywatela. Data urodzenia zapisana jest w następującym porządku: dwie ostatnie cyfry roku, miesiąc i dzień. Dla rozróżnienia stuleci przyjęto metodę kodowania:

* dla lat 1900-1999 miesiąc zapisywany jest w sposób naturalny
* dla lat 2000- 2099 do numeru miesiąca dodawana jest liczba 20

Na każdej pozycji wektora należy umieścić jedną cyfrę tj. dla przykładowego pesela: **49040501580**

* na pozycji 0-> 4
* 1->9
* …
* 6,7,8 pozycja-> cyfry losowe
* 9-> dla kobiety liczba parzysta, dla mężczyzny liczba nieparzysta (skorzystać z metody z pkt 6)
* 10-> suma kontrolna wyliczona w następujący sposób:

Każdą cyfrę numeru mnoży się przez jego wagę i sumuje wyniki mnożenia. Otrzymany wynik dzielimy modulo 10 i odejmujemy od 10 (dla wyniku 10 cyfra kontrolna przyjmuje wartość 0).

Cyfry numeru PESEL X X X X X X X X X X

Wagi 1 3 7 9 1 3 7 9 1 3

Dla naszego przykładowego peselu otrzymujemy:

suma = 1\*4+3\*9+7\*0+9\*4+1\*0+3\*5+7\*0+9\*1+1\*5+3\*8 = 120

120 % 10 = 0

10-0=10

Nasza cyfra kontrolna wynosi 0.

1. Zadeklarować stos który będzie przechowywał wektory peseli zwrócone w punkcie 8.
2. Wrzucić do stosu wektory, ale tylko poprawne (ręcznie wyselekcjonować) w odwrotnej kolejności, tak żeby na dnie stosu był obywatel o najwyższym indeksie wektora obiektów (zadeklarowanym w punkcie numer 4).

Podpowiedź: konieczne zadeklarowanie wektora pomocniczego, który będzie przechowywał tymczasowo wektor peseli.

1. Zadeklarować mapę, gdzie klucz będzie kolejną liczbą naturalną, a druga wartość to typ klasy Obywatel. Wrzucić do mapy prawidłowe obiekty (na takiej samej zasadzie jak w pkt 10).
2. Iterując po mapie (map<int, Obywatel>::iterator it = mapa.begin();) wypisać:

* Klucz mapy
* Datę urodzenia obywatela (druga wartość w mapie) podane w konstruktorze (potrzebna metoda wypisująca te dane-> podobnie jak na przykładzie2 w prezentacji)
* Pesel (zadeklarować metodę która pobiera wartość szczytowego elementu na stosie), ta wartość jest wektorem więc w metodzie należy iterować po każdej pozycji wektora wypisując po kolei cyfry peselu
* Liczbę powtarzających się ‘9’ w peselu (skorzystać z algorytmu *count(początek wektora,koniec wektora, wartość\_szukana)*) za pomocą wyrażenia lambdy (tutaj również należy pobrać wartość szczytowego elementu na sztosie). Po tej czynności usunąć wartość szczytowego elementu.

Przykładowe wyjście po jednej iteracji:

*1. 29-1-2009 09212912136 2*

1. Po zakończeniu pętli iterowania sprawdzić czy stos jest pusty (powinien być) oraz usunąć elementy wektora obiektów i również sprawdzić czy jest pusty.
2. Zadeklarować wektor typu integer oraz wypełnić go dowolnymi wartościami. Zdefiniować obiekt funkcyjny, który posortuje wartości w wektorze w porządku rosnącym (skorzystać z funkcji sort). Wypisać wartości wektora.