

## Programowanie zaawansowane, lab. 6

1. Zaprojektuj przechowywaną w pliku tekstowym w zaproponowanym przez Ciebie formacie bazę danych zawierającą informacje o ludziach. Każda **Osoba** powinna mieć określone:

- unikalny identyfikator
- imię
- nazwisko
- data urodzenia (dzień, miesiąc, rok)
- płeć
- wykształcenie
- stan cywilny (w rozumieniu czy żonaty/zamężna)
- ilość dzieci
- czy karany
- miejsce pracy
- pensja (jeśli nie jest bezrobotny)
- data zatrudnienia (jeśli nie jest bezrobotny)

Zwróć uwagę, że w ramach oszczędności miejsca część z powyższych informacji możesz zapisać w formie skróconej, np. płeć, wykształcenie i stan cywilny w postaci umownych wartości liczbowych (0,1,2,3,4...).

2. Zaimplementuj klasę reprezentującą **Osobę** z Twojej bazy danych wraz ze wszystkimi dotyczącymi jej informacjami.

Do płci, wykształcenia i stanu cywilnego zdefiniuj odpowiednie typy enumeracyjne (ponieważ mamy tam jedną z kilku z góry określonych wartości). Napisz metody, które będą konwertowały wartości liczbowe na enumeracyjne oraz w drugą stronę. Pomyśl nad obsługą dat – warto skorzystać tutaj z klasy **Calendar** i metod:

- getInstance()
- set(int field, int value)
- set(int year, int month, int date)
- get(int field)

Zastanów się również, w jaki sposób możesz zagwarantować, że nowo tworzone rekordy będą posiadały unikalne identyfikatory.

3. Napisz program, który po uruchomieniu będzie wczytywał Twoją bazę danych z pliku, wyświetlał w czytelnej formie wszystkie znajdujące się w niej **Osoby**, a następnie komunikat:

**Wybierz operację:**

1. **Wyświetl wszystkie osoby**
2. **Wyświetl osoby o podanym imieniu**
3. **Wyświetl osoby starsze, niż podana ilość lat**
4. **Wyświetl osobę o podanym ID**
5. **Edytuj osobę o podanym ID**
6. **Usuń osobę o podanym ID**
7. **Dodaj nową osobę**

**8. Czyść listę**

**9. Zapisz listę do pliku**

**10. Wczytaj listę z pliku (resetuj)**

**11. Wyjdź z programu**

Zaimplementuj wszystkie podane wyżej funkcjonalności. Po wpisaniu przez użytkownika numeru danej operacji ma się ona wykonać, a następnie ponownie wyświetlić powyższy komunikat (program ma działać w pętli, dopóki użytkownik nie wybierze operacji 11).

Wskazówki:

- Do formatowanego czytania z pliku możesz użyć klasy **Scanner**, natomiast do zapisywania **FileWriter**.
- Jeśli kolejne dane są oddzielone jakimś umownym znakiem, np. „;”, warto skorzystać podczas czytania pliku z metody **useDelimiter()** z klasy **Scanner**.
- Do operacji 3 warto wykorzystać metody z klasy **Calendar** – **before()** lub **after()**.