

Laboratorium 2

Zagadnienia:

- Krotki, rekordy, unie , opcje
- Dopasowanie wzorca

Zadanie 1.

[1p]

Zdefiniuj typ **RecDate** bazujący na rekordach reprezentujący datę. Napisz funkcje konwertujące **RecDate** na krotkę oraz krotkę na **RecDate**.

Zadanie 2.

[1p]

Napisz funkcję porównującą dwie daty reprezentowane przez **krotki** zwracającą 0 gdy daty są równe, 1 gdy druga jest późniejsza, -1 gdy pierwsza data jest późniejsza.

Zadanie 3.

[2p]

Napisz funkcję sprawdzającą poprawność daty reprezentowanej **w postaci rekordu (RecDate)**. Wykorzystaj mechanizm dopasowania wzorca. Data jest poprawna , jeśli miesiąc jest liczbą od 1 do 12, a dzień liczbą od 1 do **d**, gdzie **d=30** dla **m=4, 6, 9, 11**, dla **m=2** **d=28** lub **d=29** (dla lat przestępnych), dla pozostałych miesięcy 31.

Zadanie 4.

[2p]

Napisz funkcję, która zwróci datę wczytaną z klawiatury, lub wartość pustą (None), jeśli wprowadzona data będzie niepoprawna. Wykorzystaj **typ opcji**. Zaimplementuj prostą obsługę wyjątków (np. gdy zamiast liczby wprowadzony będzie tekst).

```
//prosta obsługa wyjątków
try
    // kod potencjalnie wyrzucający wyjątki
    // let rok = System.Console.ReadLine()
    //     ...jak data jest poprawna to zwróć Some(r,m,d)
with
| exn -> // obsługa wyjątku, np. zwrócenie None
```

Zadanie 5.

[1p]

Zdefiniuj typ **MyEvent**, reprezentujący wydarzenia. Atrybuty: nazwa zdarzenia, data i godzina, czas trwania (w minutach), miejsce, osoba. Typ powinien bazować na rekordach, ale ma wykorzystywać też krotki.

Zadanie 6.

[3p]

Napisz funkcję o sygnaturze *MyEvent -> MyEvent -> MyEvent option*, która pracuje na typie **MyEvent** z poprzedniego zadania. Funkcja ta łączy dwa zdarzenia w jedno i zwraca je (Some(event)), jeśli zdarzenia następują tuż po sobie i dotyczą tego samego miejsca i osoby. W przeciwnym wypadku zwracana jest wartość pusta (None).