

Dzień dobry. Nazywam się Marek Dalewski i chciałbym w ciągu najbliższych 10 minut przedstawić Państwu zagadnienie przetwarzania kolekcji w paradygmacie programowania funkcyjnego.

Zacznijmy jednak od ujednolicenia znaczenia użytego w tytule słowa „kolekcje”. Dla części z Państwa może być to niezrozumiałe – jakie „kolekcje”, co to w ogóle są „kolekcje”. Mamy pewne potoczne rozumienie tego słowa i w trakcie niniejszej prezentacji właśnie na takim – potocznym – znaczeniu będę się opierał. Tak więc mówiąc „kolekcja” mam na myśli pewien zbiór. Zbiór czegoś... Sama struktura danych użyta do implementacji nie ma dla nas w tym przypadku znaczenia – będziemy operować na wyższym poziomie abstrakcji. No właśnie zbiór czegoś... ale czego? W przykładach użyję dość prostego zbioru – liste zakupów spożywczych.

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

Lista zawiera 4 produkty (chleb, oliwa, sok pomarańczowy, czekolada). Dla każdego produktu znajdziemy informacje o jego >>>

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

nazwie, >>>

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

cenie jednostkowej >>>

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

oraz ilości sztuk do kupienia.

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

Tak więc mamy zbiór, mamy kolekcję. Ale w tytule była mowa o przetwarzaniu. W jaki sposób mogli byśmy chcieć przetworzyć ową listę? Jednym z przykładów mogło by być zmienienie tej listy w inną, gdzie mielibyśmy tylko nazwy produktów. W jaki sposób można to osiągnąć? Np. biorąc po kolei produkty z listy i wyrzucając nie potrzebne dane (w tym wypadku cenę jednostkową i ilość sztuk). Tak otrzymane wyniki wstawimy do drugiej listy.

Operację taką nazywamy mapowaniem.

cena jednostkowa * liczba sztuk

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

Mapowanie może być również nieco bardziej wyrafinowane. Tu dla każdego elementu nazwę pozostawiamy bez zmian, a cenę jednostkową i liczbę sztuk zastępujemy ich iloczynem.

Chleb	4,00 PLN
Oliwa	30,00 PLN
Sok pomarańczowy	9,00 PLN
Czekolada	3,50 PLN

W wyniku otrzymujemy listę produktów wraz złączną ceną, którą zapłacimy za każdy z nich.

liczba sztuk > 1

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

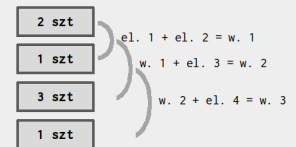
Inną operacją jest filtracja. Jak zapewne się Państwo domyślają polega na utworzeniu nowej listy, ale tylko przy użyciu eluentów, które spełniają pewne kryteria. Filtrację realizujemy poprzez wzięcie każdego elementu z lisy oryginalnej i umieszczenie go na liście wynikowej wtedy, gdy spełnia pewien warunek. Biorąc nasz przykład, możemy chcieć skonstruować listę, zawierającą produkty, których kupujemy więcej niż jeden (liczba sztuk jest większa od jeden).

Chleb	2,00 PLN	2 szt
Sok pomarańczowy	3,00 PLN	3 szt

W wyniku takiej filtracji otrzymamy następującą listę.

2 szt
1 szt
3 szt
1 szt

Czasem jednak zachodzi potrzeba wyliczenia czegoś na kolekcji i uzyskania wyniku. Np. mając listę sztuk każdego z produktów mogli byśmy chcieć otrzymać łączną ilość sztuk wszystkich produktów na liście. Taką operację nazywamy redukcją.



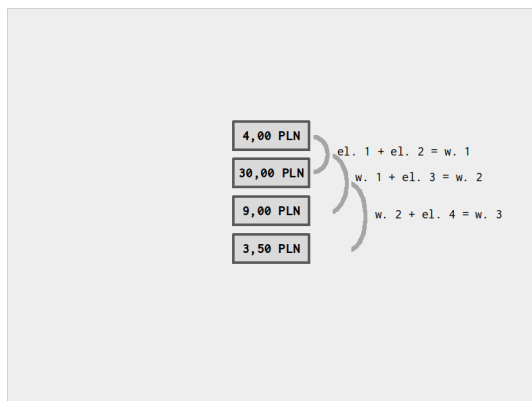
Realizujemy ją poprzez wzięcie pierwszych dwóch elementów listy i wykonanie na nich zadanej operacji (w naszym przypadku dodania). Wynik łączymy z wartością kolejnego elementu, i tak aż do końca listy.

Chleb	2,00 PLN	2 szt
Oliwa	30,00 PLN	1 szt
Sok pomarańczowy	3,00 PLN	3 szt
Czekolada	3,50 PLN	1 szt

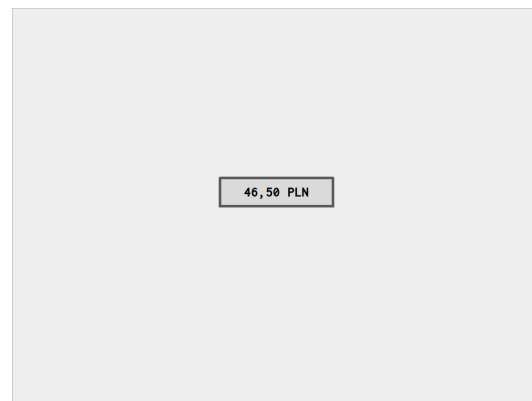
Przedstawione operacje są bardzo proste. Możemy się więc zastanawiać, jak wykonać coś bardziej złożonego. Proszę jednak zwrócić uwagę na możliwość składania wymienionych operacji. Np. by wyliczyć łączny koszt zakupów możemy najpierw dokonać mapowania >>>

cena jednostkowa • liczba sztuk			
Chleb	2,00 PLN	2 szt	4,00 PLN
Oliwa	30,00 PLN	1 szt	30,00 PLN
Sok pomarańczowy	3,00 PLN	3 szt	9,00 PLN
Czekolada	3,50 PLN	1 szt	3,50 PLN

usuwając nazwę produktu, a cenę jednostkową i liczbę sztuk zstąpić ich iloczynem.

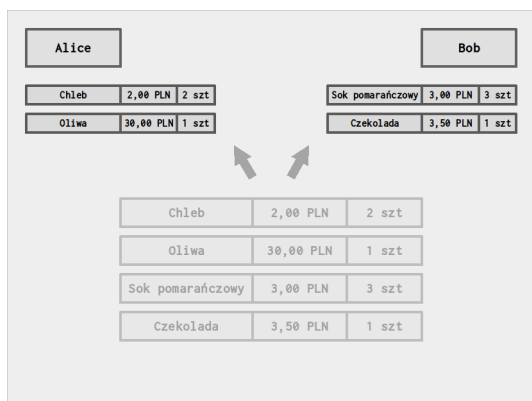


Następnie dokonać redukcji na kolekcji wynikowej, dodając wszystkie elementy.

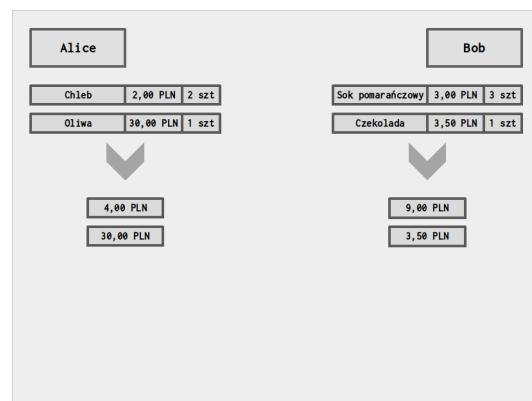


W wyniku otrzymamy jedną wartość będącą łącznym kosztem całych zakupów.

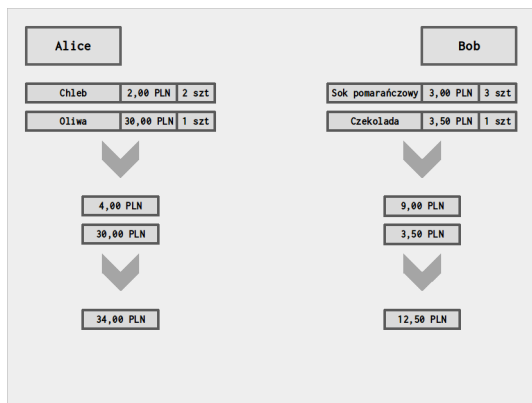
No dobrze, przetwarzanie przy użyciu wymienionych operacji jest możliwe, ale dlaczego akurat w ten sposób? Co to ma wspólnego z przetwarzaniem naprawdę dużych zbiorów danych, tj. naprawdę dużych kolekcji? Otóż okazuje się, że zastosowanie odpowiedniej abstrakcji przetwarzania (jaką dają między innymi operacje mapowania, filtracji i redukcji) umożliwia zrównoleglenie przetwarzania.



Rozważmy dokładnie ten sam problem – obliczmy łączny koszt zakupów. Jednak tym razem zamiast wykonywać wszystkie operacje samemu, na jednej liście, podzielmy ją między 2 osoby – Alce i Bob. Niech dokonają oni dokładnie tych samych operacji co my.

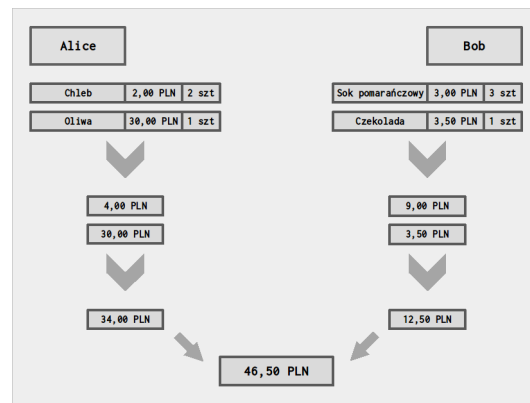


Zaczynialiśmy od mapowania usuwając nazwę produktu oraz zastępując cenę jednostkową i ilość ich iloczynem.



Następnie dokonywaliśmy redukcji, sumując wartości na liście.

W ten sposób Alice i Bob otrzymują wynik dla ich list. Nas jednak istnieje łączny wynik. Pozostaje nam więc odebranie wyników cząstkowych od Alice i Boba i ich agregacja.



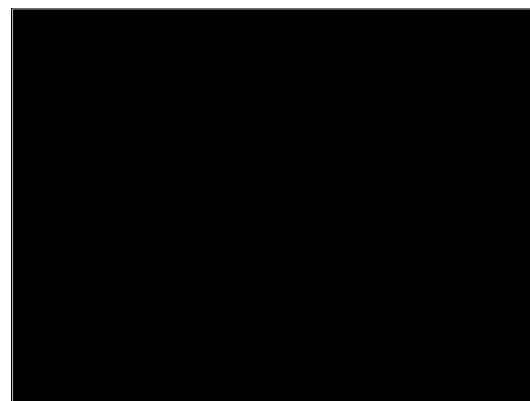
W ten sposób otrzymujemy łączny wynik. I otrzymujemy go znacznie szybciej, ponieważ możliwe było zrównoleglenie operacji.

Oczywiście, w praktyce dla tak małego zbioru ciężko zobaczyć wzrost wydajności, ale mogą państwo sobie wyobrazić listę zakupów zawierającą nie cztery a np. miliard elementów oraz nie dwóch ale tysiąc uczestników dokonujących przetwarzania.

Marek Dalewski
Funkcyjne przetwarzanie kolekcji

<https://github.com/daishe>

<https://github.com/daishe/funkcyjne-przetwarzanie-kolekcji>



To wszystko, co miałem przygotowane dziś dla Państwa. Bardzo dziękuję państwu za uwagę. Wszystkie moje materiały związane z tematem, jak i również niniejsza prezentacja znajdują się na moim GitHubie. Jeszcze raz dziękuję wszystkim za uwagę.