

# Zaawansowane kolekcje - zadania

## Spis treści

Zadanie 1 .....	1
Zadanie 2 .....	1
Zadanie 3 .....	1

## Zadanie 1

Wyobraź sobie, że idziesz do muzeum motoryzacji i wypisujesz sobie samochody, które chcesz tam zobaczyć. Wypisujesz informacje takie jak: marka, model oraz rocznik. Idziesz tam z inną osobą i ona też wypisuje swoje. Przedstaw to co napisaliście w postaci 2 list zawierających obiekty klasy `Car`. Klasa `Car` ma posiadać pola `brand`, `model` oraz `year`. Następnie sklej ze sobą te 2 listy, tak aby wyeliminować powtarzające się samochody, które dodaliście i przedstaw zawartość na ekranie posortowaną najpierw po roku rosnąco, potem po marce rosnąco oraz po modelu rosnąco. Załóżmy, że taka byłaby kolejność oglądania eksponatów w muzeum.

## Zadanie 2

Napisz implementację książki telefonicznej, która umożliwi wprowadzenie przez użytkownika osoby wraz z numerem telefonu, w sposób pokazany poniżej:

- Jan Kowalski +48000192871
- Stefan Nawrocki +48000125847
- Anna Tomaszewska +48000421458

Dodaj te osoby do Mapy w taki sposób, żeby kluczem było nazwisko, a wartością zbiór osób występujących pod takim nazwiskiem, posortowanych w kolejności nazwisko, a potem imię. Po każdej dodanej osobie, wyświetl zawartość mapy na ekranie. Spraw, żeby osoby były posortowane w książce po nazwisku. Dla uproszczenia załóż, że osoby mogą mieć tylko jedno imię i tylko jedno nazwisko.

## Zadanie 3

Napisz program, w którym stworzysz klasę `Invoice`, zawierającą pola `id` oraz `amount`. Wykorzystaj klasę `Scanner` do odczytu danych od użytkownika (wiem, że jeszcze jej nie omawialiśmy, w internecie znajdziesz dużo przykładów). Pozwól użytkownikom dodawać faktury, które będą odkładane na stosie do przetworzenia. W swoim programie zdefiniuj 2 osoby, jedną która będzie faktury dodawała (Prezes - `CEO`) oraz drugą która będzie te faktury przetwarzała (Księgowy - `ACCOUNTANT`). `CEO` korzystając z tego programu może dodać nową fakturę, a księgowy fakturę już istniejącą przetworzyć i zdjąć ze stosu. Księgowy może zdjąć i przetworzyć tylko pierwszą fakturę z góry stosu. Z racji, że program będzie przyjmował od użytkownika komendy tekstowe, musimy ustalić na jakie zachowania ma reagować. Załóżmy zatem, że jeżeli wpiszemy:

- "CEO: ADD INVOICE(190.20)" - oznacza to, że **CEO** dodaje fakturę na kwotę **190.20** i trafia ona na stos do rozliczenia. Identyfikator faktury to może być np. **1/2020**, gdzie **1** jest counterem dodawanym automatycznie przez program, natomiast rok jest dodawany automatycznie na podstawie obecnego roku.
- "ACCOUNTANT: PROCESS INVOICE" - oznacza to, że księgowy rozlicza fakturę z identyfikatorem **1/2020** i zostaje ona zdjęta ze stosu do rozliczania.

Jeżeli program nie rozpoznaje komendy, ma wypisać na ekranie: "O co Ci chodzi?". Zapewnij, aby przy każdym kroku na ekranie drukowała się zawartość stosu wraz z sumą kwot faktur jeszcze nierozliczonych. Nie narzucam obsługi przypadków skrajnych, czyli np, jak ma się zachować księgowy jak wpisze mu komendę "rozlicz fakturę", a nie ma żadnych faktur. Pozwalam Ci wymyślić co zrobić w takich sytuacjach ☺. Przykład danych wejściowych:

```
CEO: ADD INVOICE(19.20)
CEO: ADD INVOICE(31.40)
ACCOUNTANT: PROCESS INVOICE
CEO: ADD INVOICE(21.20)
CEO: ADD INVOICE(120.99)
ACCOUNTANT: PROCESS INVOICE
ACCOUNTANT: PROCESS INVOICE
ACCOUNTANT: PROCESS INVOICE
```

Przykład co może zostać wydrukowane na ekranie:

```
CEO: ADD INVOICE(123.40)
CEO added invoice: ID: 1/2021, amount: 123.40
Current invoices amount: 123.40. Invoices: [{ID: 1/2021, AMOUNT: 123.40}]

CEO: ADD INVOICE(12.41)
CEO added invoice: {ID: 2/2021, AMOUNT: 12.41}
Current invoices amount: 135.81. Invoices: [{ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]

CEO: ADD INVOICE(492.10)
CEO added invoice: {ID: 3/2021, AMOUNT: 492.10}
Current invoices amount: 627.91. Invoices: [{ID: 3/2021, AMOUNT: 492.10}, {ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]

ACCOUNTANT: PROCESS INVOICE
ACCOUNTANT processed invoice: {ID: 3/2021, AMOUNT: 492.10}
Current invoices amount: 135.81. Invoices: [{ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]

CEO: ADD INVOICE(121.40)
CEO added invoice: {ID: 4/2021, AMOUNT: 121.40}
Current invoices amount: 257.21. Invoices: [{ID: 4/2021, AMOUNT: 121.40}, {ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]

CEO: ADD INVOICE(812.41)
CEO added invoice: {ID: 5/2021, AMOUNT: 812.41}
Current invoices amount: 1069.62. Invoices: [{ID: 5/2021, AMOUNT: 812.41}, {ID: 4/2021, AMOUNT: 121.40}, {ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]
```

```
ACCOUNTANT: PROCESS INVOICE
ACCOUNTANT processed invoice: {ID: 5/2021, AMOUNT: 812.41}
```

Current invoices amount: 257.21. Invoices: [{ID: 4/2021, AMOUNT: 121.40}, {ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]

ACCOUNTANT: PROCESS INVOICE

ACCOUNTANT processed invoice: {ID: 4/2021, AMOUNT: 121.40}

Current invoices amount: 135.81. Invoices: [{ID: 2/2021, AMOUNT: 12.41}, {ID: 1/2021, AMOUNT: 123.40}]

ACCOUNTANT: PROCESS INVOICE

ACCOUNTANT processed invoice: {ID: 2/2021, AMOUNT: 12.41}

Current invoices amount: 123.40. Invoices: [{ID: 1/2021, AMOUNT: 123.40}]

ACCOUNTANT: PROCESS INVOICE

ACCOUNTANT processed invoice: {ID: 1/2021, AMOUNT: 123.40}

Current invoices amount: 0. Invoices: []