

Programowanie obiektowe II

Laboratorium nr 4

Stefan Sokołowski

Zadanie 1:

W katalogu

`iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Prog04/P3_Prac`

zdefiniowana jest klasa `Pracownik` oraz pomocniczo klasa `Test` służąca tylko do przetestowania własności klasy `Pracownik` i do wyrzucenia po sprawdzeniu.

Proszę napisać własną klasę `Szef` rozszerzającą klasę `Pracownik` (każdy szef jest również pracownikiem) o spis podwładnych (szef musi wiedzieć, kto mu podlega)¹. Konstruktor tej klasy powinien jako argumenty brać

- już istniejącego pracownika,
- zmienioną pensję na nowym stanowisku;

powinien nowemu szefowi przypisywać (na razie) pusty spis podwładnych; i powinien posiadać

- metodę dodającą mu nowego podwładnego, oraz
- metodę sprawdzającą, czy dany pracownik jest jego podwładnym.

Do tego dopisać własny program testujący nową klasę — ale istotna jest tylko klasa `Szef`, a ten program testujący jest do wyrzucenia, kiedy tylko zostanie sprawdzone, że ta klasa jest w porządku.

Uwaga:

W mojej klasie `Pracownik` **nie wolno** niczego zmieniać.

Zadanie 2:

Napisać program wykorzystujący klasę `Szef` z zad. 1, który

- „zatrudnia” 15 różnych pracowników (wymyślić im nazwiska, imiona, wykształcenie i pensje),
- dwóch z nich „mianuje” szefami,
- jednemu szefowi przydziela 5 podwładnych a drugiemu 8,
- drukuje w jakimś przejrzystym formacie (np. w tabelce) spis pracowników z zaznaczeniem, kto jest czym szefem.

Zadanie 3:

Przerobić konstruktor klasy `Pracownik` z zad. 1 tak, żeby miał pustą listę argumentów, a o potrzebne informacje pytał użytkownika. Więc skutkiem wywołania

¹Podobny (ale nieidentyczny) przykład z pracownikiem i szefem został zaprezentowany w wykładzie 2 (`iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Slajdy/02a.pdf`) na str. 19.

```
new Pracownik()
```

w programie testującym powinna być seria pytań do użytkownika programu:

```
NOWY PRACOWNIK
```

```
nazwisko:
```

```
imie:
```

```
wykształcenie (wybrać z listy)
```

```
0 — nieistotne
```

```
1 — srednie
```

```
2 — wyższe ekonomiczne
```

```
3 — wyższe techniczne
```

```
wybor:
```

```
pensja:
```

i nowy pracownik powinien być tworzony z odpowiedzi na te pytania.

Zadanie 4:

Napisać metody porządkowania spisu pracowników z zad. 1 według

- (a) nazwisk i imion (czyli: wg nazwisk alfabetycznie, ale jeśli dwa nazwiska są takie same, to decyduje alfabetyczna kolejność ich imion),
- (b) wysokości pensji i nazwisk (czyli: od najwyższej pensji, ale jeśli dwaj pracownicy mają te same pensje, to decyduje alfabetyczna kolejność ich nazwisk).

Zrobić to tak, żeby samo sortowanie było tylko jedno i mogło działać dla wszelkich uporządkowań, nie tylko tych wymienionych wyżej. Program powinien pytać o liczbę pracowników do posortowania, następnie o potrzebne dane o pracownikach; a po wczytaniu odpowiedzi drukować wynik sortowania w postaci eleganckiej tabelki.

Wskazówka:

Por. klasę `MaxMin` z wykładu 4:

`iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Slajdy/04a.pdf`
str. 19–20.

Uwaga:

Metoda sortowania powinna być własnej roboty — **nie wolno** stosować standardowego sortowania Jawy.

Zadanie 5 (domowe na punkty):

Poniższy fragment genealogii dawnych Piastów

```
Piast
|
'-Siemowit
|
'-Lestek
|
```

```

'-Siemomysl
|
|-Mieszko I
| |
| |-Mieszko
| |
| |-Lambert
| |
| |-Swietopelk
| |
| '-Boleslaw I Chrobry
| |
| |-Bezprym
| |
| |-MieszkoII
| | |
| | '-Kazimierz I Odnowiciel
| | |
| | |-Wladyslaw I Herman
| | | |
| | |-Boleslaw III Krzywousty
| | |
| | '-Boleslaw II Szczodry
| |
| '-Otto
|
'-Czcibor

```

wydrukowałem programem

```
iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biIII/Prog04/P4.Genealogia/Piastowie.java .
```

Ale potem z tego programu usunąłem konstruktor klasy `Osoba` oraz wszystkie jej prywatne własności. Proszę uzupełnić te dwa braki tak, żeby program wydrukował taką właśnie genealogię Piastów.

Uwaga:

Nie wolno niczego innego zmieniać ani dopisywać — ale oczywiście trzeba starannie przeczytać, żeby ustalić, co brakuje do działania istniejącym metodom.

Uwaga:

Czas na przysłanie (na s.sokolowski@ans-elblag.pl): 2 tygodnie od zajęć z lab.4 — potem już nie przyjmuję rozwiązań. Liczba punktów zależy od oryginalności i elegancji rozwiązania.

Zadanie 6 (domowe specjalne):

W programie, napisanym do zadania 4, zastosowaną tam metodę sortowania pracowników proszę zastąpić przez znacznie szybsze **sortowanie przez scalanie** (*merge sort*, por. [https://pl.wikipedia.org/wiki/Sortowanie](https://pl.wikipedia.org/wiki/Sortowanie_przez_scalanie)).

Uwaga:

Pierwszej osobie, która przyśle mi (na s.sokolowski@ans-elblag.pl) poprawne rozwiązanie tego zadania, przysługują punkty do zaliczenia laboratorium. Liczba tych punktów zależy od oryginalności i elegancji rozwiązania.