

Laboratorium 1

Celem ćwiczenia jest nabycie umiejętności pisania prostych programów, które komunikują się użytkownikiem w trybie konsolowym. W ćwiczeniu należy zwrócić szczególną uwagę na takie zagadnienia jak: podział programu na model danych i interfejs użytkownika, hermetyzacja danych, generowanie i obsługa wyjątków, definiowanie i wykorzystywanie typów wyliczeniowych oraz implementacja interfejsu użytkownika działającego w oknie konsoli.

Program przykładowy

Program przykładowy składa się z kilku plików źródłowych i został podzielony na dwa bloki:

- **model danych:** plik `Person.java`
- **konsolowy interfejs użytkownika:** pliki `PersonConsoleApp.java` oraz `ConsoleUserDialog.java`

W skład modelu danych wchodzi typ wyliczeniowy *enum* `PersonJob`, klasa pomocnicza `PersonException` oraz główna klasa modelu danych `Person`. Typ wyliczeniowy *enum* `PersonJob` zawiera definicję kilku wybranych stanowisk, które może zajmować osoba reprezentowana przez obiekt klasy `Person`. Typ został tak zdefiniowany, by mógł być rozszerzany o kolejne stanowiska, bez potrzeby modyfikacji pozostałych klas w programie.

Klasa `PersonException` jest rozszerzeniem klasy `Exception` i stanowi typ wyjątków dedykowany do obsługi błędów w aplikacjach wykonujących operacje na obiektach klasy `Person`.

Klasa `Person` jest główną klasą modelu danych. Obiekty tej klasy reprezentują osoby związane z uczelnią. Klasa zawiera cztery prywatne atrybuty:

- `String firstName` – imię osoby,
- `String lastName` – nazwisko osoby
- `int birthYear` – rok urodzenia osoby,
- `PersonJob job` – stanowisko zajmowane przez osobę.

Dla atrybutów powyższych atrybutów przyjęto następujące założenia:

- 1) pola `firstName` oraz `lastName` są unikalnym identyfikatorem danej osoby i muszą zawierać niepuste ciągi znaków,
- 2) pole `birthYear` musi zawierać liczbę z przedziału [1900-2030] lub wartość 0, która oznacza niezdefiniowany rok urodzenia.
- 3) pole `job` musi zawierać wyłącznie jedną z pozycji zdefiniowanych w typie wyliczeniowym *enum* `PersonJob`.

Ponadto klasa `Person` zawiera:

- publiczny konstruktor z dwoma parametrami umożliwiającymi określenie imienia i nazwiska osoby,
- publiczne gettery umożliwiające odczyt wartości prywatnych atrybutów,
- publiczne settery, umożliwiające nadawanie wartości prywatnym atrybutom,
- publiczną metodę `toString`, która zwraca imię i nazwisko osoby,

- publiczne statyczne metody umożliwiające zapis i odczyt obiektów klasy *Person* do/z plików tekstowych.

Klasa została zdefiniowana w ten sposób by nie zawierała jakichkolwiek operacji związanych implementacją interfejsu użytkownika. Konstruktor oraz wszystkie settery kontrolują, czy nadawane wartości spełniają powyższe założenia i w razie potrzeby zgłaszają wyjątek klasy *PersonException* z komunikatem tekstowym wskazującym przyczynę błędu. Komunikat ten będzie wyświetlany w modułach realizujących dialog z użytkownikiem w programie konsolowym lub okienkowym.

Moduł interfejsu użytkownika pozwala na przetestowanie wszystkich operacji wykonywanych na obiektach klasy *Person*. Zawiera on implementację aplikacji testowej działającej w konsoli tekstowej, która umożliwia wykonanie poniższych operacji:

- 1) tworzenie nowej osoby,
- 2) modyfikowanie danych aktualnej osoby,
- 3) usuwanie danych aktualnej osoby,
- 4) zapis danych aktualnej osoby do pliku tekstowego o podanej nazwie,
- 5) odczyt danych osoby z pliku tekstowego o podanej nazwie,
- 6) zakończenie działania aplikacji.

Ponadto program wyświetla krótki komunikat zawierający informacje o autorze programu.

Wersja konsolowa interfejsu użytkownika to klasa *PersonConsoleApp* i jest zawarta w pliku *PersonConsoleApp.java*. Przy jej implementacji wykorzystano w charakterze pomocniczym klasę *ConsoleUserDialog*, która zawiera pomocnicze metody do wyświetlania komunikatów tekstowych w oknie konsoli oraz metody do wczytywania danych różnych typów z klawiatury. Plik *ConsoleUserDialog.java* ma charakter uniwersalny i może być wykorzystywany w innych programach konsolowych.

Zadanie 1

Proszę uruchomić i przetestować działanie aplikacji. Następnie proszę przeanalizować kod źródłowy zawarty w plikach z rozszerzeniem **.java*.

Aplikacja konsolowa jest zawarta w pliku *PersonConsoleApp.jar*. Uruchomienie tej aplikacji następuje poprzez wpisanie następującego polecenia w oknie konsoli:

```
java -jar PersonConsoleApp.jar
```

Zadanie 2

Wzorując się na klasie *Person* z programu przykładowego proszę zdefiniować własną klasę reprezentującą dowolnie wybrane obiekty, które będą mogły być w przyszłości łączone w różne grupy (np. książki, programy komputerowe, znaczki pocztowe itp.).

Klasa powinna zawierać co najmniej 4 atrybuty, w tym jeden atrybut typu *String*, jeden atrybut typu numerycznego (typ *int*, *float* lub *double*) oraz jeden atrybut typu wyliczeniowego. Pozostałe atrybuty mogą być dowolnych typów według uznania autora.

Dla wszystkich atrybutów proszę określić dodatkowe założenia dotyczące wartości, które mogą przyjmować.

Proszę zdefiniować co najmniej 1 konstruktor oraz publiczne gettery, settery i przedefiniowaną metodę *toString*. Ponadto proszę przygotować pomocnicze metody statyczne umożliwiające zapis i odczyt obiektów do plików tekstowych.

Proszę przygotować prostą aplikację konsolową umożliwiającą przetestowanie działania wszystkich metod własnej klasy. Uwaga: w programie można wykorzystać plik źródłowy `ConsoleUserDialog.java` z programu przykładowego.

Wszystkie pliki źródłowe proszę wysłać do sprawdzenia (**Uwaga:** Ten program będzie rozwijany na następnych zajęciach i zostanie oceniony po drugich zajęciach)

Uwaga:

Wszystkie pliki źródłowe z rozszerzeniem **.java* muszą na początku zawierać komentarz, w którym będą zawarte następujące informacje:

- zawartość pliku,
- imię i nazwisko oraz numer indeksu autora,
- data powstania programu.

Ponadto, każdy program powinien na początku drukować dane autora, lub posiadać opcję w menu lub przycisk, który powoduje wyświetlenie tych danych.

Programy nie podpisane w ten sposób będą miały obniżoną ocenę!!!!