Laboratorium 2

Celem ćwiczenia jest nabycie umiejętności pisania prostych programów, które komunikują się użytkownikiem w trybie konsolowym i okienkowym. 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, implementacja interfejsu użytkownika działającego w oknie konsoli oraz implementacja interfejsu użytkownika działającego w środowisku okienkowym.

Program przykładowy

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

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

Moduły interfejsu użytkownika pozwalają na przetestowanie wszystkich operacji wykonywanych na obiektach klasy Person. Zawierają one implementację aplikacji testowej działającej w konsoli tekstowej lub w środowisku okienkowym. Oba warianty aplikacji umożliwiają taką samą funkcjonalność:

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 obydwa programy wyświetlają krótki komunikat zawierający informacje o autorze programu.

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

```
java –jar PersonConsoleApp.jar
```

Aplikacja okienkowa zawarta jest w pliku `PersonWindowApp.jar`. Można ją uruchomić w ten sam sposób w konsoli lub poprzez podwójne kliknięcie na pliku `PersonWindowApp.jar`.

Zadanie 1

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

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.

Wzorując się na aplikacji okienkowej z programu przykładowego proszę napisać aplikację z okienkowym interfejsem użytkownika, które umożliwia przetestowane działania wszystkich metod z własnej klasy. Aplikacja powinna tworzyć okno główne programu oraz okno dialogowe umożliwiające wykonywanie operacji edycyjnych.

Zadanie 3 (dla ambitnych)

We własnej klasie proszę dodać pomocnicze metody statyczne umożliwiające zapis i odczyt danych do plików binarnych z wykorzystaniem mechanizmu serializacji obiektów.

Uwaga: we własnej klasie należy zaimplementować interfejs *Serializable*, a do zapisu i odczytu danych należy wykorzystać metody *writeObject* i *readObject*.

Aplikacje konsolowa i okienkowa powinny zostać uzupełnione o możliwość zapisu i odczytu danych do/z plików binarnych.

Zadanie 4 (dla ambitnych)

W aplikacji okienkowej proszę użyć standardowego okna dialogowego do wyboru nazwy pliku, który jest zdefiniowany w klasie *JFileChooser.*

Zadanie 5(dla ambitnych)

W głównym oknie aplikacji okienkowej proszę dołożyć pasek menu, który zawiera opcje umożliwiające wykonywanie tych samych operacji, które umożliwiają w programie przykładowym przyciski rozmieszczone na panelu głównym. W tym celu należy utworzyć obiekt klasy *JMenuBar* stanowiący pasek menu, który należy dodać do okna głównego aplikacji za pomocą metody *setJMenuBar*. Następnie należy utworzyć pomocniczy obiekt klasy *JMenu* stanowiący rozwijane w dół menu oraz dodać ten obiekt do paska menu za pomocą metody *add*. Następnie należy utworzyć kilka obiektów klasy *JMenuItem* stanowiących poszczególne opcje menu i dodać je do rozwijanego menu za pomocą metody *add*. Do obiektów z klasy *JMenuItem* należy dołączyć słuchaczy zdarzeń podobnie jak to było zrobione dla przycisków. W metodzie *actionPerformed* należy uwzględnić dodatkowe opcje menu, które będą źródłem zdarzeń.