

Platformy technologiczne

Laboratorium 1

Java: Kolekcje

Należy napisać program w Javie SE, który będzie czytał zawartość podanego katalogu na dysku a następnie go wyświetlał w postaci posortowanej lub nieposortowanej (zależnie od wybranej opcji). Dodatkowo program ma umożliwiać zapisanie oraz odczytanie obiektu reprezentującego katalog na dysku. Katalog (lub plik zawierający obiekt) do przeczytania, tryb wyświetlania oraz plik, w którym ma zostać zapisany obiekt mają być podawane jako argumenty wywołania programu. Kryterium sortowania oraz format wyświetlanych danych podaje prowadzący na zajęciach.

Aby zrealizować żadaną funkcjonalność należy utworzyć klasę, która będzie reprezentowała obiekt na dysku. Powinna mieć następujące pola: nazwa pliku, data ostatniej modyfikacji, rozmiar pliku (dla katalogu w tym polu ma znajdować się ilość elementów zawartych w tym katalogu) i typ pliku (plik zwykły lub katalog) będący polem typu enum.

Zawartość katalogów możemy odczytać korzystając z klasy `java.io.File`. Posiada ona metodę, która zwraca tablicę obiektów typu `File`, reprezentującą zawartość podanego katalogu. Odczytane i skopiowane do stworzonych przez siebie obiektów dane wkładamy do zbioru. Zmienna typu zbiór, zależnie od wybranej opcji wywołania programu, ma być zainicjowana klasą `TreeSet` (zbiór posortowany) lub `HashSet` (zbiór nieposortowany).

W celu umożliwienia zapisywania i odczytywania obiektu reprezentującego katalog należy wykorzystać mechanizm serializacji poprzez implementację interfejsu `Serializable` oraz użycie strumieni działających na obiektach `ObjectInputStream` oraz `ObjectOutputStream` udostępniające metody `readObject` oraz `writeObject`.

Format wyświetlanych danych zostanie podany przez prowadzącego na zajęciach, przykładowo może mieć następującą formę: przy wypisywaniu zawartości zbioru najpierw należy wydrukować linijkę zawierającą znaki „>>>” a potem w kolejnych liniijkach pliki zgodnie z poniższym formatem:
nazwa_pliku typ rozmiar data_modyfikacji
gdzie typ to litera „P” lub „K” a data modyfikacji jest sformatowana zgodnie ze wzorcem: „yyyy-MM-dd HH:mm” przy pomocy klasy `java.text.SimpleDateFormat`. Po wypisaniu zawartości katalogu należy wypisać liniijkę zawierającą znaki „<<<”.

Aby program poprawnie działał stworzona klasa przechowujące dane o plikach musi implementować interfejs `Comparable` oraz metody `equals(...)` i `hashCode(...)`. Metody do porównywania obiektów (`equals(...)` i `compareTo()`) oraz metoda `hashCode(...)` powinny być spójne, tzn. powinny uwzględniać te same pola obiektu. Oznacza to, że np. jeśli chcemy by pliki były posortowane wg rozmiaru to ww. metody powinny uwzględniać pola: nazwa pliku i rozmiar, ponieważ mogą istnieć dwa pliki o takim samym rozmiarze ale różnych nazwach.

Należy wykorzystać mechanizm *generics*, tzn. gdy deklarujemy zmienną przechowującą zbiór z obiektami o plikach należy napisać:

```
Set<NazwaNaszejKlasy> nazwaZmiennej;
```

A gdy deklarujemy, że nasza klasa implementuje interfejs `Comparable` należy napisać:

```
public class NazwaNaszejKlasy implements Comparable<NazwaNaszejKlasy> {...}
```

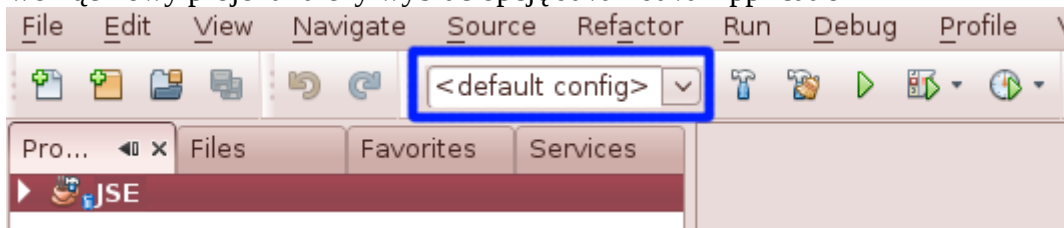
Klasa reprezentująca obiekty plikowe ma być napisana zgodnie z konwencją *JavaBeans*, tzn. pola klasy mają być chronione lub prywatne a dostęp do nich ma następować przez gettery i settery.

Punktacja:

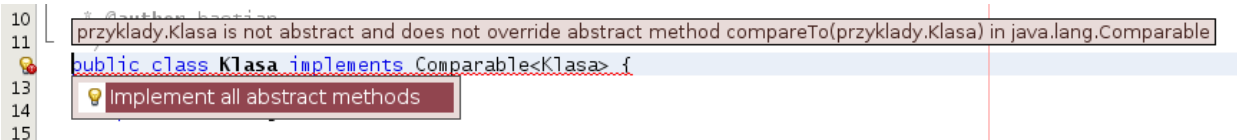
- wypisanie zawartości jednego katalogu (bez rekurencji): 1pkt,
- wypisanie zawartości podkatalogów (rekurencyjnie): 1pkt,
- serializacja (zapis i odczyt obiektu): 1pkt,
- sortowanie: 1pkt,
- poprawne sformatowanie wyjścia programu: 1pkt.

Podpowiedzi:

1. Działanie kolekcji, sposoby tworzenia enumów, itd. były omówione na wykładzie i są zawarte w przykładzie do wykładu.
2. Tworząc nowy projekt należy wybrać opcję Java->Java Application



3. W NetBeansie, argumenty wywołania programu można podawać wybierając opcję „Customize” z listy zaznaczonej na powyższym obrazku.
4. NetBeans potrafi wygenerować poprawne metody `equals(...)` i `hashCode(...)` dla wybranych pól. Opcja dostępna w menu Source -> Insert Code
5. NetBeans potrafi wygenerować gettery i settery dla wybranych pól. Opcja dostępna w menu Source -> Insert Code



6. W NetBeansie pojawiają się podpowiedzi dotyczące kodu źródłowego oznaczone żarówką (patrz rysunek powyżej). Aby uzyskać dostęp do akcji pomocniczych trzeba kliknąć na żarówkę lub nacisnąć `Alt+Enter`.
7. Pisząc implementację metody `compareTo(...)` można posłużyć się istniejącymi implementacjami tej metody dostępnymi w większości standardowych klas (np. `String`, `Date`).