

Zadania z programowania w języku Java dla II roku Informatyki.

dr Agnieszka Zbrzezny

Interfejsy

1. Zdefiniuj klasę **Osoba**, która implementuje interfejs **Cloneable** oraz generyczny interfejs **Comparable**. W klasie tej zadeklaruj prywatne pola **nazwisko** typu **String** oraz **dataUrodzenia** typu **LocalDate**. Klasę **Osoba** umieść w pakiecie o nazwie **pl.kmi.ujd.nazwisko**, gdzie **nazwisko** to Twoje nazwisko bez polskich liter.
 - (a) nadpisz metodę **toString** z klasy **Object**. Metoda **toString** powinna zwracać łańcuch zawierający nazwę klasy, a po niej (w nawiasach kwadratowych) nazwisko osoby oraz datę jej urodzenia w formacie **yyyy-MM-dd**.
 - (b) nadpisz metodę **equals** z klasy **Object**, przy czym w porównaniu uwzględnij oba pola klasy.
 - (c) Implementując metodę **compareTo** interfejsu **Comparable** najpierw porównuj nazwiska, a następnie daty urodzenia. Zadbaj o to aby naturalny porządek wyznaczony przez metodę **compareTo** był zgodny z metodą **equals**
 - (d) Napisz program **TestOsoba**, w którym utwórz listę 5 osób o nazwie **grupa** posługując się klasą **ListArray**. W składzie grupy powinny wystąpić przynajmniej dwie osoby o tym samym nazwisku i innej dacie urodzenia oraz dwie osoby o tej samej dacie urodzenia oraz różnych nazwiskach. Ponadto umieść w grupie siebie. Wyświetl zawartość listy **grupa**, po czym posortuj ją za pomocą instancyjnej metody **sort** z klasy **ArrayList** i ponownie wyświetl zawartość tej listy.
2. Zdefiniuj klasę **Student** dziedziczącą po klasie **Osoba** z poprzedniego zadania. Klasa **Student** ma implementować interfejs **Cloneable** oraz generyczny interfejs **Comparable**. W klasie tej zadeklaruj prywatne pole **sredniaOcen** typu **double**. Implementując metodę **compareTo** interfejsu **Comparable** skorzystaj z metody **compareTo** zdefiniowanej w klasie **Osoba** oraz, w razie potrzeby, uwzględnij pole **sredniaOcen**. Klasę **Student** umieść w tym samym pakiecie, w którym znajduje się klasa **Osoba**. Napisz program **TestStudent** w sposób analogiczny do programu **TestOsoba**.
3. Napisz program, który czyta kolejne linie z pliku podanego jako argument wywołania programu i umieszcza je w kolekcji **lines** typu **ArrayList<String>**. Następnie tworzy tablicę **linie** będącą „kopią” listy tablicowej **lines**, po czym:
 - (a) sortuje tę listę tablicową **lines** w porządku naturalnym, po czym wypisuje ją na ekran.
 - (b) sortuje tę listę tablicową **lines** w kolejności od najkrótszych do najdłuższych linii, po czym wypisuje ją na ekran.
 - (c) sortuje tablicę **linie** w porządku naturalnym, po czym wypisuje ją na ekran.
 - (d) sortuje tablicową **linie** w kolejności od najkrótszych do najdłuższych linii, po czym wypisuje ją na ekran.

Uwaga: W powyższych zadaniach przez program rozumiemy klasę zawierającą publiczną i statyczną metodę **main**, która nie zwraca żadnej wartości i której jedyny argument jest typu **String[]**.