

Programowanie obiektowe w Javie - interfejs Comparable oraz Comparator

A1. Napisz klasę `Student`, która zawiera pola: `imie` (typu `String`), `sredniaOcen` (typu `double`) i `rokUrodzenia` (typu `int`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Student` były sortowane malejąco według średniej ocen. Stwórz listę 5 obiektów klasy `Student` i posortuj ją według sprecyzowanego kryterium.

A2. Napisz klasę `Pracownik`, która zawiera pola: `imie` (typu `String`), `pensja` (typu `double`) i `dataZatrudnienia` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Pracownik` były sortowane rosnąco według pensji. Stwórz listę 5 obiektów klasy `Pracownik` i posortuj ją według sprecyzowanego kryterium.

A3. Napisz klasę `Klient`, która zawiera pola: `imie` (typu `String`), `nrKlienta` (typu `int`) i `ostatnieLogowanie` (typu `Date`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Klient` były sortowane malejąco według daty ostatniego logowania. Stwórz listę 5 obiektów klasy `Klient` i posortuj ją według sprecyzowanego kryterium.

A4. Napisz klasę `Produkt`, która zawiera pola: `nazwa` (typu `String`), `cena` (typu `double`) i `dataProdukcji` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Produkt` były sortowane rosnąco według daty produkcji. Stwórz listę 5 obiektów klasy `Produkt` i posortuj ją według sprecyzowanego kryterium.

A5. Napisz klasę `Osoba`, która zawiera pola: `imie` (typu `String`), `wzrost` (typu `int`) i `dataUrodzenia` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Osoba` były sortowane malejąco według wzrostu. Stwórz listę 5 obiektów klasy `Osoba` i posortuj ją według sprecyzowanego kryterium.

B1. Napisz klasę `Ksiazka`, która zawiera pola: `tytul` (typu `String`), `liczbaStron` (typu `int`) i `dataWydania` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Ksiazka` były sortowane malejąco według liczby stron. Stwórz tablicę 4 obiektów klasy `Ksiazka` i posortuj ją według sprecyzowanego kryterium.

B2. Napisz klasę `Samochod`, która zawiera pola: `marka` (typu `String`), `przebieg` (typu `int`) i `rokProdukcji` (typu `int`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Samochod` były sortowane rosnąco według przebiegu. Stwórz tablicę 4 obiektów klasy `Samochod` i posortuj ją według sprecyzowanego kryterium.

B3. Napisz klasę `ProduktSpozywczy`, która zawiera pola: `nazwa` (typu `String`), `cena` (typu `double`) i `dataWaznosci` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `ProduktSpozywczy` były sortowane rosnąco według daty ważności. Stwórz tablicę 4 obiektów klasy `ProduktSpozywczy` i posortuj ją według sprecyzowanego kryterium.

B4. Napisz klasę `Muzyka`, która zawiera pola: `tytul` (typu `String`), `artysta` (typu `String`) i `rokWydania` (typu `int`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Muzyka` były sortowane malejąco według roku wydania. Stwórz tablicę 4 obiektów klasy `Muzyka` i posortuj ją według sprecyzowanego kryterium.

B5. Napisz klasę `Przedmiot`, która zawiera pola: `nazwa` (typu `String`), `waga` (typu `double`) i `cena` (typu `double`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Przedmiot` były sortowane rosnąco według wagi. Stwórz tablicę 4 obiektów klasy `Przedmiot` i posortuj ją według sprecyzowanego kryterium.

C1. Napisz klasę `Student`, która zawiera pola: `imie` (typu `String`), `sredniaOcen` (typu `double`) i `rokStudiow` (typu `int`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Student` były sortowane według jednego kryterium: malejąco według średniej ocen, a przy równości sortowane były rosnąco według roku studiów. Stwórz tablicę 4 obiektów klasy `Student` i posortuj ją według sprecyzowanego kryterium.

C2. Napisz klasę `Zamowienie`, która zawiera pola: `nazwaProduktu` (typu `String`), `ilosc` (typu `int`) i `cenaJednostkowa` (typu `double`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Zamowienie` były sortowane według jednego kryterium: malejąco według ceny jednostkowej, a przy równości sortowane były rosnąco według ilości. Stwórz listę tablicową 4 obiektów klasy `Zamowienie` i posortuj ją według sprecyzowanego kryterium.

C3. Napisz klasę `Klient`, która zawiera pola: `imie` (typu `String`), `saldo` (typu `double`) i `ostatnieZakupy` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Klient` były sortowane według jednego kryterium: malejąco według salda, a przy równości sortowane były rosnąco według daty ostatnich zakupów. Stwórz listę tablicową 4 obiektów klasy `Klient` i posortuj ją według sprecyzowanego kryterium.

C4. Napisz klasę `Kurs`, która zawiera pola: `nazwa` (typu `String`), `liczbaGodzin` (typu `int`) i `cena` (typu `double`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Kurs` były sortowane według jednego kryterium: rosnąco według liczby godzin, a przy równości

sortowane były malejąco według ceny. Stwórz tablicę 4 obiektów klasy `Kurs` i posortuj ją według sprecyzowanego kryterium.

C5. Napisz klasę `Produkt`, która zawiera pola: `nazwa` (typu `String`), `cena` (typu `double`) i `dataWaznosci` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Produkt` były sortowane według jednego kryterium: malejąco według daty ważności, a przy równości sortowane były rosnąco według ceny. Stwórz listę obiektów klasy `Produkt` i posortuj ją według sprecyzowanego kryterium. Następnie wyświetl posortowaną listę na ekranie.

D1. Napisz klasę `Samochód`, która zawiera pola: `marka` (typu `String`), `model` (typu `String`) i `numerRejestracyjny` (typu `String`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Samochód` były sortowane według jednego kryterium: rosnąco według długości numeru rejestracyjnego. Stwórz tablicę 4 obiektów klasy `Samochód` i posortuj ją według sprecyzowanego kryterium.

D2. Napisz klasę `Pracownik`, która zawiera pola: `imie` (typu `String`), `nazwisko` (typu `String`) i `stanowisko` (typu `String`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Pracownik` były sortowane według jednego kryterium: rosnąco według długości nazwiska. Stwórz listę tablicową 4 obiektów klasy `Pracownik` i posortuj ją według sprecyzowanego kryterium.

D3. Napisz klasę `Film`, która zawiera pola: `tytuł` (typu `String`), `reżyser` (typu `String`) i `gatunek` (typu `String`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Film` były sortowane według jednego kryterium: rosnąco według długości tytułu. Stwórz listę tablicową 4 obiektów klasy `Film` i posortuj ją według sprecyzowanego kryterium.

D4. Napisz klasę `Książka`, która zawiera pola: `tytuł` (typu `String`), `autor` (typu `String`) i `dataWydania` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Książka` były sortowane według jednego niestandardowego kryterium: rosnąco według roku wydania. Stwórz tablicę 4 obiektów klasy `Książka` i posortuj ją według sprecyzowanego kryterium.

D5. Napisz klasę `Produkt`, która zawiera pola: `nazwa` (typu `String`), `cena` (typu `double`) i `dataProdukcji` (typu `LocalDate`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Produkt` były sortowane według jednego niestandardowego kryterium: malejąco według roku produkcji. Stwórz listę tablicową 4 obiektów klasy `Produkt` i posortuj ją według sprecyzowanego kryterium.

E1. Zdefiniuj klasę Klient, która będzie implementować generyczny interfejs Comparable. W klasie tej zadeklaruj prywatne pola nazwisko typu String oraz saldo typu double. Implementując metodę compareTo interfejsu Comparable, porównuj klientów na podstawie ich salda, a w przypadku takiego samego salda - na podstawie nazwiska. Następnie zdefiniuj klasę Firma dziedziczącą po klasie Klient. Klasa Firma ma dodatkowo posiadać prywatne pole liczbaPracowników typu int. Implementując metodę compareTo interfejsu Comparable w klasie Firma, skorzystaj z metody compareTo zdefiniowanej w klasie Klient oraz, w razie potrzeby, uwzględnij pole liczbaPracowników. Napisz program TestKlient, w którym utwórz listę 5 klientów i firm o nazwie listaKlientów posługując się klasą ArrayList. W składzie listy powinny wystąpić przynajmniej dwóch klientów o takim samym saldzie i różnym nazwisku oraz dwie firmy o takiej samej liczbie pracowników i różnym saldzie. Wyświetl zawartość listy listaKlientów, posortuj ją za pomocą instancyjnej metody sort z klasy ArrayList i ponownie wyświetl zawartość tej listy.

E2. Zdefiniuj klasę Zwierzę, która będzie implementować generyczny interfejs Comparable. W klasie tej zadeklaruj prywatne pola gatunek typu String oraz wiek typu int. Implementując metodę compareTo interfejsu Comparable, porównuj zwierzęta na podstawie ich wieku, a w przypadku takiego samego wieku - na podstawie gatunku. Następnie zdefiniuj klasę Pies dziedziczącą po klasie Zwierzę. Klasa Pies ma dodatkowo posiadać prywatne pole rasa typu String. Implementując metodę compareTo interfejsu Comparable w klasie Pies, skorzystaj z metody compareTo zdefiniowanej w klasie Zwierzę oraz, w razie potrzeby, uwzględnij pole rasa. Napisz program TestZwierzę, w którym utwórz listę 5 zwierząt i psów o nazwie listaZwierząt posługując się klasą ArrayList. W składzie listy powinny wystąpić przynajmniej

F1. Napisz klasę Osoba z polami imię (String), wiek (int) i wzrost (double). Napisz klasę implementującą interfejs Comparator, która porównuje osoby na podstawie wieku. Stwórz tablicę 5 osób i posortuj ją według wieku.

F2. Napisz klasę Produkt z polami nazwa (String), cena (double) i dataWaznosc (LocalDate). Napisz klasę implementującą interfejs Comparator, która porównuje produkty na podstawie daty ważności. Stwórz listę 5 produktów i posortuj ją według daty ważności.

F3. Napisz klasę Samochód z polami marka (String), rokProdukcji (int) i cena (double). Napisz klasę implementującą interfejs Comparator, która porównuje samochody na podstawie roku produkcji. Stwórz tablicę 5 samochodów i posortuj ją według roku produkcji.

F4. Napisz klasę Pracownik z polami imię (String), pensja (double) i dataZatrudnienia (LocalDate). Napisz klasę implementującą interfejs Comparator, która porównuje pracowników na podstawie pensji. Stwórz tablicę 5 pracowników i posortuj ją według pensji.

F5. Napisz klasę Książka z polami tytuł (String), cena (double) i dataWydania (Date). Napisz klasę implementującą interfejs Comparator, która porównuje książki na podstawie daty wydania. Stwórz listę 5 książek i posortuj ją według daty wydania.

G1. Napisz klasę “Product” z polami “id” (typu int), “name” (typu String) oraz “price” (typu double). Zaimplementuj generyczny interfejs “Comparator” do porównywania obiektów po polu “price” (od najniższej do najwyższej ceny), a w przypadku równości po polu “id”. Stwórz listę 5 obiektów klasy “Product” i posortuj ją zgodnie z opisanym kryterium.

G2. Napisz klasę “Person” z polami “firstName” (typu String), “lastName” (typu String) oraz “birthDate” (typu LocalDate). Zaimplementuj generyczny interfejs “Comparator” do porównywania obiektów po polu “lastName” (alfabetycznie od A do Z), a w przypadku równości po polu “firstName”. Stwórz tablicę 5 obiektów klasy “Person” i posortuj ją zgodnie z opisanym kryterium.

G3. Napisz klasę “Order” z polami “id” (typu int), “customerName” (typu String) oraz “orderDate” (typu LocalDate). Zaimplementuj generyczny interfejs “Comparator” do porównywania obiektów po polu “orderDate” (od najwcześniejszej do najpóźniejszej daty), a w przypadku równości po polu “id”. Stwórz listę 5 obiektów klasy “Order” i posortuj ją zgodnie z opisanym kryterium.

G4. Napisz klasę “Song” z polami “title” (typu String), “artist” (typu String) oraz “duration” (typu int). Zaimplementuj generyczny interfejs “Comparator” do porównywania obiektów po polu “duration” (od najkrótszej do najdłuższej piosenki), a w przypadku równości po polu “title”. Stwórz tablicę 5 obiektów klasy “Song” i posortuj ją zgodnie z opisanym kryterium.

G5. Napisz klasę “Student” z polami “id” (typu int), “name” (typu String) oraz “averageGrade” (typu double). Zaimplementuj generyczny interfejs “Comparator” do porównywania obiektów po polu “averageGrade” (od najwyższej do najniższej średniej ocen), a w przypadku równości po polu “name”. Stwórz listę 5 obiektów klasy “Student” i posortuj ją zgodnie z opisanym kryterium.

H1. Napisz klasę “Product” z polami “id” (typu int), “name” (typu String) oraz “price” (typu double). Zaimplementuj dwie klasy implementujące generyczny interfejs “Comparator”: “PriceComparator” do porównywania obiektów po polu “price” (od najniższej do najwyższej ceny) oraz “NameComparator” do porównywania obiektów po polu “name” (alfabetycznie od A do Z). Stwórz listę 5 obiektów klasy “Product” i posortuj ją zgodnie z oboma kryteriami (najpierw po cenie, a następnie po nazwie).

H2. Napisz klasę “Person” z polami “firstName” (typu String), “lastName” (typu String) oraz “birthDate” (typu LocalDate). Zaimplementuj dwie klasy implementujące generyczny interfejs “Comparator”: “LastNameComparator” do porównywania obiektów po polu “lastName” (alfabetycznie od A do Z) oraz “BirthDateComparator” do porównywania obiektów po polu “birthDate” (od najstarszej do najmłodszej osoby). Stwórz tablicę 5 obiektów klasy “Person”

i posortuj ją zgodnie z oboma kryteriami (najpierw po nazwisku, a następnie po dacie urodzenia).

H3. Napisz klasę “Order” z polami “id” (typu int), “customerName” (typu String) oraz “orderDate” (typu LocalDate). Zaimplementuj dwie klasy implementujące generyczny interfejs “Comparator”: “OrderDateComparator” do porównywania obiektów po polu “orderDate” (od najwcześniejszej do najpóźniejszej daty) oraz “CustomerNameComparator” do porównywania obiektów po polu “customerName” (alfabetycznie od A do Z). Stwórz listę 5 obiektów klasy “Order” i posortuj ją zgodnie z oboma kryteriami (najpierw po dacie zamówienia, a następnie po nazwie klienta).

H4. Napisz klasę “Song” z polami “title” (typu String), “artist” (typu String) oraz “duration” (typu int). Zaimplementuj dwie klasy implementujące generyczny interfejs “Comparator”: “DurationComparator” do porównywania obiektów po polu “duration” (od najkrótszej do najdłuższej piosenki) oraz “ArtistTitleComparator” do porównywania obiektów po polu “artist” (alfabetycznie od A do Z) i w przypadku równości po polu “title”. Stwórz tablicę 5 obiektów klasy “Song” i posortuj ją zgodnie z oboma kryteriami (najpierw po długości utworu, a następnie po artyście i tytule).

H5. Napisz klasę “Student” z polami “id” (typu int), “name” (typu String) oraz “averageGrade” (typu double). Zaimplementuj dwie klasy implementujące generyczny interfejs “Comparator”: “AverageGradeComparator” do porównywania obiektów po polu “averageGrade” (od najwyższej do najniższej średniej ocen) oraz “IdComparator” do porównywania obiektów po polu “id” (od najniższego do najwyższego identyfikatora). Stwórz listę 5 obiektów klasy “Student” i posortuj ją zgodnie z oboma kryteriami (najpierw po średniej ocen, a następnie po identyfikatorze).