

# Programowanie obiektowe w Javie - interfejs Cloneable

A1. Napisz klasę `Student` z trzema polami: `name` (`String`), `age` (`int`) i `grade` (`double`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Student`, sklonuj go, a następnie zmień ocenę (`grade`) oryginalnego studenta. Wyświetl oceny obu studentów, aby zobaczyć, czy są niezależne.

A2. Napisz klasę `Teacher` z trzema polami: `name` (`String`), `subject` (`String`) i `experience` (`int`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Teacher`, sklonuj go, a następnie zmień doświadczenie (`experience`) oryginalnego nauczyciela. Wyświetl doświadczenie obu nauczycieli, aby zobaczyć, czy są niezależne.

A3. Napisz klasę `Car` z trzema polami: `make` (`String`), `model` (`String`) i `mileage` (`double`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Car`, sklonuj go, a następnie zmień przebieg (`mileage`) oryginalnego samochodu. Wyświetl przebieg obu samochodów, aby zobaczyć, czy są niezależne.

---

B1. Napisz klasę `Smartphone` z trzema polami: `brand` (`String`), `model` (`String`) i `productionDate` (typu `Date`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Smartphone`, sklonuj go, a następnie zmień datę produkcji oryginalnego smartfona. Wyświetl datę produkcji obu smartfonów, aby zobaczyć, czy są niezależne.

B2. Napisz klasę `Laptop` z trzema polami: `brand` (`String`), `model` (`String`) i `purchaseDate` (typu `Date`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Laptop`, sklonuj go, a następnie zmień datę zakupu (`purchaseDate`) oryginalnego laptopa. Wyświetl datę zakupu obu laptopów, aby zobaczyć, czy są niezależne.

B3. Napisz klasę `VideoGame` z trzema polami: `title` (`String`), `genre` (`String`) i `releaseDate` (typu `Date`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `VideoGame`, sklonuj go, a następnie zmień datę wydania (`releaseDate`) oryginalnej gry. Wyświetl datę wydania obu gier, aby zobaczyć, czy są niezależne.

---

C1. Napisz klasę `CreditCard` z trzema polami: `cardNumber` (`String`), `holderName` (`String`) i `expiryDate` (typu `LocalDate`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `CreditCard`, sklonuj go, a następnie zmień datę wygaśnięcia (`expiryDate`) oryginalnej karty kredytowej. Wyświetl datę wygaśnięcia obu kart, aby zobaczyć, czy są niezależne.

C2. Napisz klasę `BankAccount` z trzema polami: `accountNumber` (`String`), `accountHolder` (`String`) i `openingDate` (typu `LocalDate`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `BankAccount`, sklonuj go, a następnie zmień datę otwarcia (`openingDate`) oryginalnego konta bankowego. Wyświetl datę otwarcia obu kont, aby zobaczyć, czy są niezależne.

C3. Napisz klasę `DrivingLicense` z trzema polami: `licenseNumber` (`String`), `holderName` (`String`) i `issueDate` (typu `LocalDate`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `DrivingLicense`, sklonuj go, a następnie zmień datę wydania (`issueDate`) oryginalnego prawa jazdy. Wyświetl datę wydania obu praw jazdy, aby zobaczyć, czy są niezależne.

---

D1. Napisz klasę `Employee` z dwoma polami: `name` (`String`) i `salaries` (tablica 12 zmiennych typu `double`, reprezentująca zarobki za każdy miesiąc). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Employee`, sklonuj go, a następnie zmień zarobki na pozycji 5 (czerwiec) oryginalnego pracownika. Wyświetl zarobki obu pracowników, aby zobaczyć, czy są niezależne.

D2. Napisz klasę `Athlete` z dwoma polami: `name` (`String`) i `times` (tablica 5 zmiennych typu `double`, reprezentująca czas w sekundach potrzebny na przebiegnięcie 100 metrów podczas różnych prób). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Athlete`, sklonuj go, a następnie zmień czas na pozycji 3 oryginalnego sportowca. Wyświetl czasy obu sportowców, aby zobaczyć, czy są niezależne.

D3. Napisz klasę `Teacher` z dwoma polami: `name` (`String`) i `studentsGrades` (tablica 10 zmiennych typu `double`, reprezentująca oceny każdego z 10 uczniów). Zaimplementuj interfejs

`Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Teacher`, sklonuj go, a następnie zmień ocenę na pozycji 10 oryginalnego nauczyciela. Wyświetl oceny obu nauczycieli, aby zobaczyć, czy są niezależne.

---

E1. Napisz klasę `Employee` z dwoma polami: `name` (`String`) i `monthlyHours` (lista tablicowa zmiennych typu `int`, reprezentująca liczbę przepracowanych godzin w każdym miesiącu). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Employee`, sklonuj go, a następnie zmień liczbę godzin na pozycji 5 (czerwiec) oryginalnego pracownika. Wyświetl liczbę godzin obu pracowników, aby zobaczyć, czy są niezależne.

E2. Napisz klasę `Athlete` z dwoma polami: `name` (`String`) i `lapTimes` (lista tablicowa zmiennych typu `int`, reprezentująca czas w sekundach potrzebny na przebiegnięcie okrążenia podczas różnych prób). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Athlete`, sklonuj go, a następnie zmień czas na pozycji 3 oryginalnego sportowca. Wyświetl czasy obu sportowców, aby zobaczyć, czy są niezależne.

E3. Napisz klasę `Teacher` z dwoma polami: `name` (`String`) i `studentsGrades` (lista tablicowa zmiennych typu `int`, reprezentująca oceny każdego z uczniów). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()`, aby móc klonować obiekty tej klasy. W metodzie `main()` utwórz obiekt `Teacher`, sklonuj go, a następnie zmień ocenę na pozycji 10 oryginalnego nauczyciela. Wyświetl oceny obu nauczycieli, aby zobaczyć, czy są niezależne.

---

F1. Napisz klasę `Teacher` z polami `name` (`String`), `age` (`int`) i `salary` (`double`). Następnie napisz klasę `HeadTeacher`, która dziedziczy po klasie `Teacher` i dodaje pole `bonus` (`double`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()` w obu klasach. W metodzie `main()` pokaż przykład prezentujący poprawność klonowania obiektów tych klas.

F2. Napisz klasę `Developer` z polami `name` (`String`), `age` (`int`) i `salary` (`double`). Następnie napisz klasę `SeniorDeveloper`, która dziedziczy po klasie `Developer` i dodaje pole `bonus` (`double`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()` w obu klasach. W metodzie `main()` pokaż przykład prezentujący poprawność klonowania obiektów tych klas.

F3. Napisz klasę `Nurse` z polami `name` (`String`), `age` (`int`) i `salary` (`double`). Następnie napisz klasę `HeadNurse`, która dziedziczy po klasie `Nurse` i dodaje pole `bonus` (`double`). Zaimplementuj interfejs `Cloneable` i nadpisz metodę `clone()` w obu klasach. W metodzie `main()` pokaż przykład prezentujący poprawność klonowania obiektów tych klas.