Podany jest przykład kodu:

Main.java

```java
// Klasa bazowa Pracownik
class Pracownik {  2 usages  2 inheritors
    String imie;  5 usages
    int wiek;  2 usages
    // Konstruktor
    Pracownik(String imie, int wiek) {    // Konstruktor  2 usages
        this.imie = imie;
        this.wiek = wiek;
    }
    // Metoda wyświetlająca informacje o pracowniku
    void wyswietlInformacje() {  3 usages  1 override
        System.out.println("Imię: " + imie + ", Wiek: " + wiek);
    }
    // Metoda symulująca pracę
    void pracuj() {  2 usages  1 override
        System.out.println(imie + " pracuje.");
    }
}
// Klasa PracownikBudowy dziedzicząca po Pracownik
class PracownikBudowy extends Pracownik {  2 usages
    String specjalizacja;  2 usages
    int doswiadczenie;  2 usages
    // Konstruktor
    PracownikBudowy(String imie, int wiek, String specjalizacja, int doswiadczenie) {  1 usage
        super(imie, wiek);
        this.specjalizacja = specjalizacja;
        this.doswiadczenie = doswiadczenie;
    }
    // Przesłonięta metoda wyświetlająca informacje o pracowniku budowy
    @Override  3 usages
    void wyswietlInformacje() {
        super.wyswietlInformacje();
        System.out.println("Specjalizacja: " + specjalizacja + ", Doświadczenie: " + doswiadczenie + " lat");
    }
    // Metoda specyficzna dla pracownika budowy
    void buduj() {  1 usage
        System.out.println(imie + " buduje.");
    }
}
// Klasa PracownikStazysta dziedzicząca po Pracownik
class PracownikStazysta extends Pracownik {  2 usages
    String mentor;  2 usages
    // Konstruktor
    PracownikStazysta(String imie, int wiek, String mentor) {  1 usage
        super(imie, wiek);
        this.mentor = mentor;
    }
    // Przesłonięta metoda symulująca pracę
    @Override  2 usages
    void pracuj() {
        System.out.println(imie + " uczy się pod okiem mentora " + mentor + ".");
    }
}
// Klasa główna do testowania
public class Main {
    public static void main(String[] args) {
        PracownikBudowy pb = new PracownikBudowy( imie: "Jan",  wiek: 35,  specjalizacja: "Murarz",  doswiadczenie: 10);
        PracownikStazysta ps = new PracownikStazysta( imie: "Anna",  wiek: 22,  mentor: "Kowalski");

        pb.wyswietlInformacje();
        pb.pracuj();
        pb.buduj();
        ps.wyswietlInformacje();
        ps.pracuj();
    }
}
```

W tym przykładzie mamy trzy klasy: **Pracownik**, **PracownikBudowy** i **PracownikStazysta**.
Klasa **Pracownik** jest klasą bazową, z której dziedziczą **PracownikBudowy** i **PracownikStazysta**. Klasa **PracownikBudowy** dodaje nowe pola i metody, a także przesłania metodę wyswietlInformacje. Klasa **PracownikStazysta** przesłania metodę pracuj, aby dostosować ją do specyfiki stażu.

Zadanie

*Dostosuj powyższy kod do trzech różnych scenariuszy: **szkoły**, **szpitala** i **grupy antyterrorystycznej**. W każdym przypadku utwórz odpowiednie klasy bazowe i dziedziczące, dodając specyficzne pola i metody dla każdej z nich.*

1. Szkoła
Klasa bazowa: **PracownikSzkoly**
Pola: **imie, wiek**
Metody: **wyswietlInformacje(), pracuj()**
Klasa dziedzicząca: **Nauczyciel**
Dodatkowe pola: **przedmiot, doswiadczenie**
Dodatkowe metody: **nauczaj()**
Przesłonięta metoda: **wyswietlInformacje()**
Klasa dziedzicząca: **StazystaSzkoly**
Dodatkowe pole: **mentor**
Przesłonięta metoda: **pracuj()**
2. Szpital
Klasa bazowa: **PracownikSzpitala**
Pola: **imie, wiek**
Metody: **wyswietlInformacje(), pracuj()**
Klasa dziedzicząca: **Lekarz**
Dodatkowe pola: **specjalizacja, doswiadczenie**
Dodatkowe metody: **lecz()**
Przesłonięta metoda: **wyswietlInformacje()**
Klasa dziedzicząca: **StazystaSzpitala**
Dodatkowe pole: **mentor**
Przesłonięta metoda: **pracuj()**
3. Grupa Antyterrorystyczna
Klasa bazowa: **PracownikAntyterrorystyczny**
Pola: **imie, wiek**
Metody: **wyswietlInformacje(), pracuj()**
Klasa dziedzicząca: **Agent**
Dodatkowe pola: **specjalizacja, doswiadczenie**
Dodatkowe metody: **operuj()**
Przesłonięta metoda: **wyswietlInformacje()**
Klasa dziedzicząca: **StazystaAntyterrorystyczny**
Dodatkowe pole: **mentor**
Przesłonięta metoda: **pracuj()**

WYMAGANIA DO ZADANIA

Uwaga kod nie spełniający poniższych wymagań zostanie oceniony na ndst:
Klasy oraz ich składowe (pola i metody) udokumentuj w stylu Javadoc.

Pamiętaj aby również w tym styku podpisać kod imieniem nazwiskiem i klasą fragment rozwiązania zadania razem z fragmentem komentarzy Javadoc przedstawia ilustracja poniżej:

```java
/**
 * Klasa reprezentująca pracownika szkoły.
 *
 * @autor Jan Kowalski
 * @version 1.0
 * @since 2024-10-27
 */
class PracownikSzkoly {
    String imie;
    int wiek;

    /**
     * Konstruktor klasy PracownikSzkoly.
     *
     * @param imie Imię pracownika.
     * @param wiek Wiek pracownika.
     */
    PracownikSzkoly(String imie, int wiek) {
        this.imie = imie;
        this.wiek = wiek;
    }

    /**
     * Wyświetla informacje o pracowniku.
     */
    void wyswietlInformacje() {
        System.out.println("Imię: " + imie + ", Wiek: " + wiek);
    }

    /**
     * Symuluje pracę pracownika.
     */
    void pracuj() {
        System.out.println(imie + " pracuje w szkole.");
    }
}
```