

```
abstract class Marks{ 2 usages 2 inheritors
    public abstract double getPercentage();
}
```

Klasa Marks: Abstrakcyjna klasa z metodą getPercentage() do obliczania procentów punktów.

```
class A extends Marks{ 2 usages
    private int subject1, subject2, subject3; 2 usages

    public A(int subject1, int subject2, int subject3){ 1 usage
        this.subject1 = subject1;
        this.subject2 = subject2;
        this.subject3 = subject3;
    }

    @Override 2 usages
    public double getPercentage() {
        int totalPoints = subject1 + subject2 + subject3;
        return (totalPoints / 300.0) * 100;
    }
}
```

Klasa A: Dziedziczy po Marks, przechowuje punkty z 3 przedmiotów i oblicza procent na podstawie maks. 300 punktów.

```
class B extends Marks{ 2 usages
    private int subject1, subject2, subject3, subject4; 2 usages

    public B(int subject1, int subject2, int subject3, int subject4){ 1 usage
        this.subject1 = subject1;
        this.subject2 = subject2;
        this.subject3 = subject3;
        this.subject4 = subject4;
    }

    @Override 2 usages
    public double getPercentage() {
        int totalPoints = subject1 + subject2 + subject3;
        return (totalPoints / 400.0) * 100;
    }
}
```

Klasa B: Dziedziczy po Marks, przechowuje punkty z 4 przedmiotów i oblicza procent na podstawie maks. 400 punktów.

```
public class Main {
    public static void main(String[] args) {
        A studentA = new A(57, 25, 86);
        B studentB = new B(50, 24, 35, 100);

        System.out.println("Procent zdobytych punktów przez studenta A: " + studentA.getPercentage() + "%");
        System.out.println("Procent zdobytych punktów przez studenta B: " + studentB.getPercentage() + "%");
    }
}
```

Klasa Main: Tworzy obiekty klas A i B, wywołuje getPercentage() i wypisuje wyniki na konsolę.