

```
interface CoffeeMachine { 4 usages 4 implementations
    String prepareCoffee(); 3 usages 3 implementations
}
```

Interfejs CoffeeMachine deklaruje metodę `prepareCoffee()`, która ma być zaimplementowana przez każdą klasę reprezentującą maszynę do parzenia kawy. Metoda ta zwraca ciąg znaków (String) opisujący przygotowaną kawę.

```
abstract class BaseCoffee implements CoffeeMachine { 3 usages 3 inheritors
    protected int water; 4 usages
    protected int coffee; 4 usages
    protected int milk; 3 usages

    public BaseCoffee(int water, int coffee, int milk) { 3 usages
        this.water = water;
        this.coffee = coffee;
        this.milk = milk;
    }
}
```

Klasa BaseCoffee implementuje interfejs `CoffeeMachine` i zawiera wspólne właściwości dla wszystkich rodzajów kaw: `water` (ilość wody), `coffee` (ilość kawy), `milk` (ilość mleka). Te właściwości są inicjowane w konstruktorze klasy. Klasa ta nie implementuje metody `prepareCoffee()`, pozostawiając to klasom dziedziczącym po niej.

```
class Espresso extends BaseCoffee { 1 usage
    public Espresso() { 1 usage
        super( water: 50, coffee: 1, milk: 0);
    }

    @Override 3 usages
    public String prepareCoffee() {
        return "Przygotowano Espresso: " + water + " ml wody, " + coffee + " porcja kawy.";
    }
}
```

Klasa Espresso dziedziczy po klasie `BaseCoffee` i ustawia specyficzne wartości dla espresso: 50 ml wody, 1 porcja kawy, 0 ml mleka. Implementuje metodę `prepareCoffee()`, która zwraca tekst opisujący przygotowanie espresso, uwzględniając ilości wody i kawy.

```

class Latte extends BaseCoffee { 1 usage
    public Latte() { 1 usage
        super( water: 30, coffee: 1, milk: 150);
    }

    @Override 3 usages
    public String prepareCoffee() {
        return "Przygotowano Latte: " + water + " ml wody, " + coffee + " porcja kawy, " + milk + " ml mleka.";
    }
}

```

Klasa Latte dziedziczy po klasie BaseCoffee i ustawia specyficzne wartości dla latte: 30 ml wody, 1 porcja kawy, 150 ml mleka. Implementuje metodę prepareCoffee(), która zwraca tekst opisujący przygotowanie latte, uwzględniając ilości wody, kawy i mleka.

```

class Cappuccino extends BaseCoffee { 1 usage
    public Cappuccino() { 1 usage
        super( water: 30, coffee: 1, milk: 100);
    }

    @Override 3 usages
    public String prepareCoffee() {
        return "Przygotowano Cappuccino: " + water + " ml wody, " + coffee + " porcja kawy, " + milk + " ml mleka.";
    }
}

```

Klasa Cappuccino dziedziczy po klasie BaseCoffee i ustawia specyficzne wartości dla cappuccino: 30 ml wody, 1 porcja kawy, 100 ml mleka. Implementuje metodę prepareCoffee(), która zwraca tekst opisujący przygotowanie cappuccino, uwzględniając ilości wody, kawy i mleka.

```

public class Main {
    public static void main(String[] args) {
        CoffeeMachine espresso = new Espresso();
        CoffeeMachine latte = new Latte();
        CoffeeMachine cappuccino = new Cappuccino();

        System.out.println(espresso.prepareCoffee());
        System.out.println(latte.prepareCoffee());
        System.out.println(cappuccino.prepareCoffee());
    }
}

```

W klasie Main w metodzie main() tworzone są obiekty trzech rodzajów kaw: Espresso, Latte i Cappuccino. Dla każdego obiektu wywoływana jest metoda prepareCoffee(), która wypisuje szczegóły dotyczące przygotowania każdej z kaw na ekranie.

DZIAŁANIE

1. Program tworzy obiekty Espresso, Latte i Cappuccino, które są różnymi rodzajami kaw.
2. Każdy obiekt ma swoją implementację metody `prepareCoffee()`, która zwraca tekst z informacjami o ilości wody, kawy i mleka.
3. Program wypisuje na ekranie:
 - Espresso: "Przygotowano Espresso: 50 ml wody, 1 porcja kawy."
 - Latte: "Przygotowano Latte: 30 ml wody, 1 porcja kawy, 150 ml mleka."
 - Cappuccino: "Przygotowano Cappuccino: 30 ml wody, 1 porcja kawy, 100 ml mleka."