

1 Klasa abstrakcyjna

Klasa abstrakcyjna zawiera metody, które są implementowane w klasach pochodnych. Nie można tworzyć obiektów na podstawie klasy abstrakcyjnej

```
abstract class Abstrakcyjna{
    public abstract void metodaAbstrakcyjna();
}

class Klasa extends Abstrakcyjna{
    @Override
    public void metodaAbstrakcyjna(){
        System.out.println("Juz nie");
    }
}
```

2 Interfejsy

Interfejsy zawierają tylko i wyłącznie metody, które implementowane są w klasach pochodnych.

```
interface Interfejs{
    public void metodaInterfejsu();
}

class Klasa implements Interfejs{
    @Override
    public void metodaInterfejsu(){
        System.out.println("Implementacja interfejsu");
    }
}
```

Klasa może implementować wiele interfejsów.

3 Obiekty anonimowe

```
class Element {
    void print(){
        System.out.println("print");
    }
}

public class Klasa1 {
    public Klasa1() {
```

```

    super();
}

public void metoda(Element e){
    e.print();
}

static void main(String[] args){
    Klasa1 k = new Klasa1();
    k.metoda(new Element());
}
}

```

4 Zadania do wykonania

1. Przetestować klasy abstrakcyjne
2. Przetestować interfejsy
3. Sprawdzić możliwość użycia metod abstrakcyjnych
4. Zrzutować klasę na implementowany interfejs
5. Stworzyć obiekt anonimowy
6. Stworzyć klasę anonimową na podstawie interfejsu
7. Przetestować możliwość dziedziczenia interfejsów
8. Wykorzystać interfejs *Serializable* we własnej klasie
9. Wykorzystać interfejs *Cloneable* we własnej klasie
10. Wykorzystać interfejs *Comparable* we własnej klasie
11. Posortować tablicę obiektów
12. Wyszukać binarnie obiekt w tablicy obiektów