Object oriented programming - Adam Krechowicz

1 Abstract class

Abstract class contains methods that can be implemented in derived classes. We can not create objects based on abstract classes.

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

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

2 Interfaces

Interfaces usually contains only the specifications of methods that needs to be implemented in derived classes.

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

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

Class can implements many interfaces.

3 Anonymous objects

```
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 Tasks to complete

1. Use abstract classes
2. Use interfaces
3. Try to use abstract methods
4. Cast object to interface
5. Create anonymous object
6. Create anonymous class based on interface
7. Create anonymous class based on abstract class
8. Try to extend interfaces
9. Use *Serializable* interface in own class
10. Use *Clonable* interface in own class
11. Use *Comparable* interface in own class
12. Sort array of objects
13. Perform binary search on array of objects