

PROGRAMTERVEZÉSI MINTÁK KIDOLGOZVA

1. Bridge

```
package mintak;

interface Szin {
    void feltolt();
}

class Piros implements Szin {
    public void feltolt() {
        System.out.println("Feltöltés Piros színnel");
    }
}

class Kek implements Szin {
    public void feltolt() {
        System.out.println("Feltöltés Kék színnel");
    }
}

abstract class Alakzat {
    protected Szin szin;

    public Alakzat(Szin szin) {
        this.szin = szin;
    }

    abstract void rajzol();
}

class Kor extends Alakzat {
    public Kor(Szin szin) {
        super(szin);
    }

    public void rajzol() {
        System.out.print("Kör rajzolása");
        szin.feltolt();
    }
}

class Negyzet extends Alakzat {
    public Negyzet(Szin szin) {
        super(szin);
    }

    public void rajzol() {
        System.out.print("Négyzet rajzolása");
        szin.feltolt();
    }
}

public class Bridge {
    public static void main(String[] args) {
        Alakzat pirosKor = new Kor(new Piros());
        pirosKor.rajzol();

        Alakzat kekNegyzet = new Negyzet(new Kek());
        kekNegyzet.rajzol();
    }
}
```

2. Builder

```
package mintak;

class Kocsi {
    private String model;
    private String szin;
    private int gyartasiEv;

    public Kocsi(String model, String szin, int ev) {
        this.model = model;
        this.szin = szin;
        this.gyartasiEv = ev;
    }

    @Override
    public String toString() {
        return "Kocsi [Model=" + model + ", Szín=" + szin + ",  
Gyártás éve=" + gyartasiEv + "]\n";
    }
}

class KocsiBuilder {
    private String model;
    private String szin;
    private int gyartasiEv;

    public KocsiBuilder setModel(String model) {
        this.model = model;
        return this;
    }

    public KocsiBuilder setSzin(String szin) {
        this.szin = szin;
        return this;
    }

    public KocsiBuilder setGyartasiEv(int gyartasiEv) {
        this.gyartasiEv = gyartasiEv;
        return this;
    }

    public Kocsi build() {
        return new Kocsi(model, szin, gyartasiEv);
    }
}

public class Builder {
    public static void main(String[] args) {
        Kocsi car = new  
KocsiBuilder().setModel("Ford").setSzin("Fekete").setGyartasiEv(2024)  
.build();
        System.out.println(car);
    }
}
```

3. Factory

```
package mintak;

interface Allat {
    void hangKiadas();
}

class Kutya implements Allat {
    public void hangKiadas() {
        System.out.println("Vau!");
    }
}

class Macska implements Allat {
    public void hangKiadas() {
        System.out.println("Meow!");
    }
}

abstract class AllatFactory {
    public abstract Allat letrehoz();
}

class KutyaFactory extends AllatFactory {
    public Allat letrehoz() {
        return new Kutya();
    }
}

class MacskaFactory extends AllatFactory {
    public Allat letrehoz() {
        return new Macska();
    }
}

public class Factory {
    public static void main(String[] args) {
        AllatFactory kutyak = new KutyaFactory();
        Allat kutya = kutyak.letrehoz();
        kutya.hangKiadas();

        AllatFactory macskak = new MacskaFactory();
        Allat macska = macskak.letrehoz();
        macska.hangKiadas();
    }
}
```

4. Iterator

```
package mintak;

import java.util.List;
import java.util.ArrayList;

interface Iterator {
    boolean hasNext();
    Object next();
}

class KocsIterator implements Iterator {
    private List<String> kocsik;
    private int position = 0;

    public KocsIterator(List<String> kocsik) {
        this.kocsik = kocsik;
    }

    public boolean hasNext() {
        return position < kocsik.size();
    }

    public Object next() {
        return hasNext() ? kocsik.get(position++) : null;
    }
}

class KocsiCollection {
    private List<String> kocsik = new ArrayList<>();

    public void hozzaAd(String kocsi) {
        kocsik.add(kocsi);
    }

    public Iterator getIterator() {
        return new KocsIterator(kocsik);
    }
}

public class IteratorMinta {
    public static void main(String[] args) {
        KocsiCollection collection = new KocsiCollection();
        collection.hozzaAd("Tesla");
        collection.hozzaAd("BMW");
        collection.hozzaAd("Audi");

        Iterator iterator = collection.getIterator();
        while (iterator.hasNext()) {
            System.out.println("Kocsi: " + iterator.next());
        }
    }
}
```

5. Prototype

```
package mintak;

interface Prototype {
    Prototype clone();
}

class Jarmu implements Prototype {
    private String model;

    public Jarmu(String model) {
        this.model = model;
    }

    public String getModel() {
        return model;
    }

    @Override
    public Prototype clone() {
        return new Jarmu(this.model);
    }

    @Override
    public String toString() {
        return "Jarmu [Model=" + model + "]";
    }
}

public class PrototypeMinta {
    public static void main(String[] args) {
        Jarmu eredeti = new Jarmu("Tesla Model S");
        Jarmu lemasolt = (Jarmu) eredeti.clone();

        System.out.println(eredeti);
        System.out.println(lemasolt);
    }
}
```