

## ПРОЕКТ „ДВОИЧНО-ДЕСЕТИЧНО КОДИРАНЕ“ – статична реализация

Всеки полубайт на байт съдържа двоичния запис на една десетична цифра.

### Реализация

```
//Помощен клас: един байт, кодиран двоично-десетично
class ByteBCD{
private int v; //Поддържаеме свойството в int, заради знака
public ByteBCD(){ //Празен конструктор
    v=0;
}
public void setV(int a){ //Сетър с вход int
    v=a & 0b11111111; //само младшите 8 бита
}
public byte getV(){//Гетър-конвертор в тип байт
    return (byte)v;
}
public ByteBCD(int a){//Конструктор с int вход
    setV(a);
}
public ByteBCD(int H,int L){//Конструктор по два полубайта
    setV((H<<4) | (L & 0b1111));
}
public byte lo(){ //Младша десетична цифра
    return (byte) (v & 0b1111);
}
public byte hi(){ //Старша десетична цифра
    return (byte) (v>>>4);
}
//Десетично събиране на два BCD-байта с пренос в CF[0]
//Преносът след събирането остава в CF[0]
public ByteBCD add(ByteBCD b,byte[]CF){
    //Сума на младшите цифри и преноса
    ByteBCD p=new ByteBCD(CF[0]+this.lo()+b.lo());
    if (p.v>9){p.v-=10;//Надхвърля цифра
        CF[0]=1;//Пренос
    }
    else CF[0]=0; //Иначе няма пренос
    //Аналогична сума за старшите цифри
    ByteBCD q=new ByteBCD(CF[0]+this.hi()+b.hi());
    if (q.v>9){q.v-=10;//повече от цифра
        CF[0]=1;
    }
    else CF[0]=0; //Иначе няма пренос
    return new ByteBCD(q.v,p.v);
}
//Форматиран низ (с евентуална водеща нула, ако twoChar е true)
public String format(boolean twoChar){
    String s=Integer.toHexString(v);
    if (twoChar && s.length()<2) s='0'+s;
    return s;
}
public String toString(){
    return format(false);//без водеща нула
}
}
class BCD{
//Стоична бройна система: "цифрите" са числа от 0 до 99
private byte[] d; //Масив от "цифри" (по две цифри
//от най-младша към най-старша двойка)
private int cnt; //Брой значещи байтове
BCD(){//Празен конструктор
    d=new byte[1024]; //Статична реализация:
//не повече от 1024 "цифри" (2024 цифри)
    d[0]=0;//Създава числото 0...
    cnt=1; //... с една "цифра".
}
}
```

```

BCD(String s){//Конструктор от низ
    d=new byte[1024];
    int L=s.length();
    if ((L & 1) == 1){//Проверка за нечетен брой цифри
        //без деление
        s='0'+s;
        L++;
    }
    int i=L-2;
    for (cnt=0;i>=0;cnt++,i-=2){
        d[cnt]=new ByteBCD(s.charAt(i),s.charAt(i+1)).getV();
    }
}
public int getCnt(){
    return cnt;
}
public BCD add(BCD a){
    BCD r=new BCD();//Резултат
    ByteBCD p=new ByteBCD();//Първо събираемо
    ByteBCD q=new ByteBCD();//Второ събираемо
    byte CF[]=new byte[1];//Пренос
    int i;//Номер на разряда
    for (i=0;i<cnt || i<a.cnt;i++){//докато има значеца "цифра"
        //поне в едното събираемо
        p.setV(i<cnt?d[i]:0);//"цифра" от първото събираемо
        q.setV(i<a.cnt?a.d[i]:0);//"цифра" от второто събираемо
        r.d[i]=p.add(q, CF).getV();
    }
    if (CF[0]==1)r.d[i++]=1;//нова цифра 1
    r.cnt=i;//Брой значещи байтове в резултата
    return r;
}
@Override
public String toString(){
    //Най-старшият значещ байт - без водеща нула
    String s=new ByteBCD(d[cnt-1]).toString();
    for (int i=cnt-2;i>=0;i--)
        //останалите са задължително в два символа
        s+=new ByteBCD(d[i]).format(true);
    return s;
}
}
public class Main {
    public static void main(String[] args) {
        BCD a=new BCD("999");
        BCD b=new BCD("11");
        BCD c=a.add(b);
        System.out.println(a + "+"+b+"="+c);
    }
}

```