DIASOFT
всё по-настоящему

DIASOFT
всё по-настоящему

V. Ввод – вывод

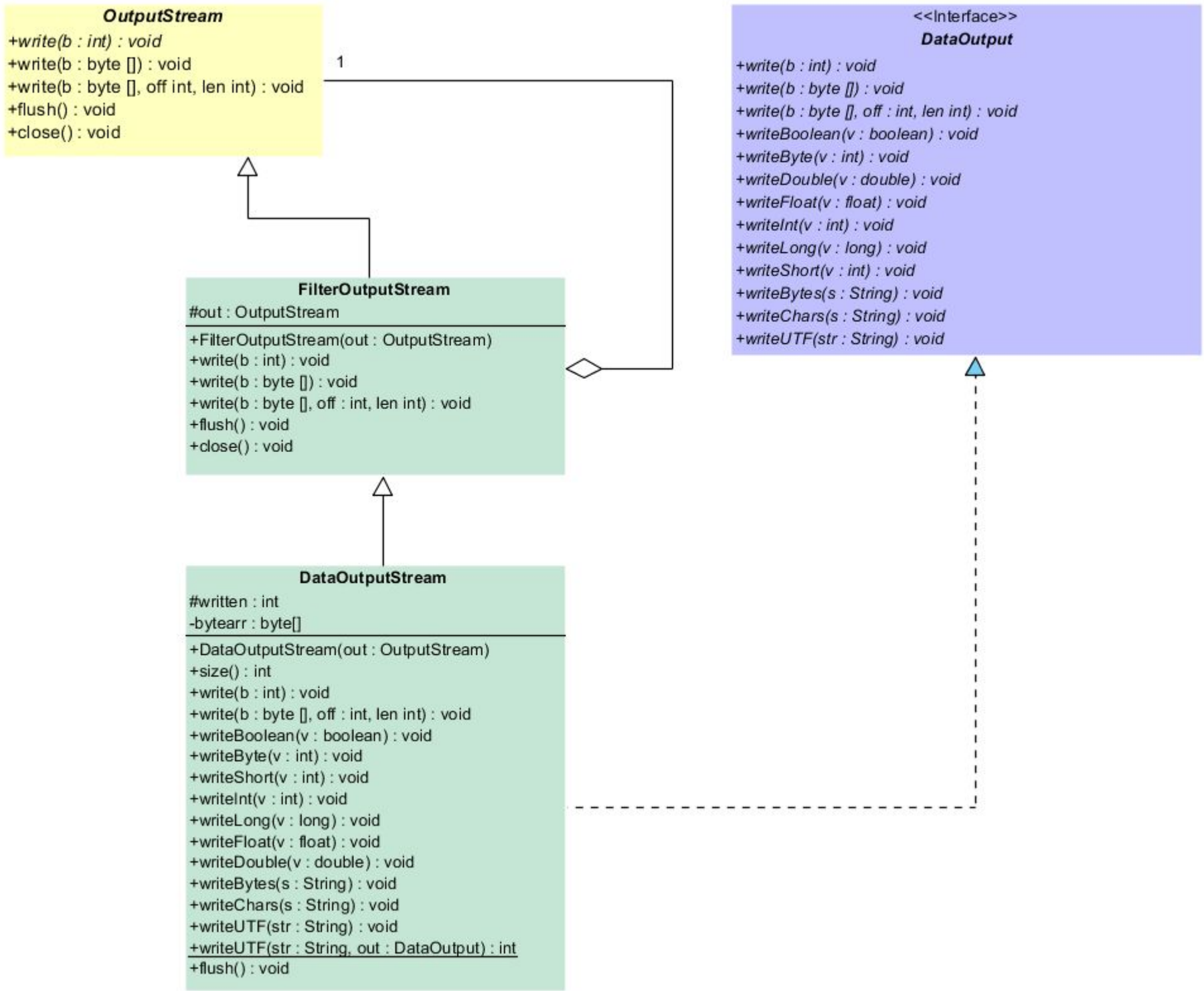
4. Потоки данных

Потоки данных

Объект из которого можно последовательно прочитать значения примитивных типов называется поток ввода данных. Объект в который можно последовательно записать значения примитивных типов называется поток вывода данных. Классом для потоков ввода данных является класс DataInputStream, а потоков вывода данных класс DataOutputStream. Классы потоков данных находятся в пакете java.io.

Потоки вывода данных

Класс DataOutputStream



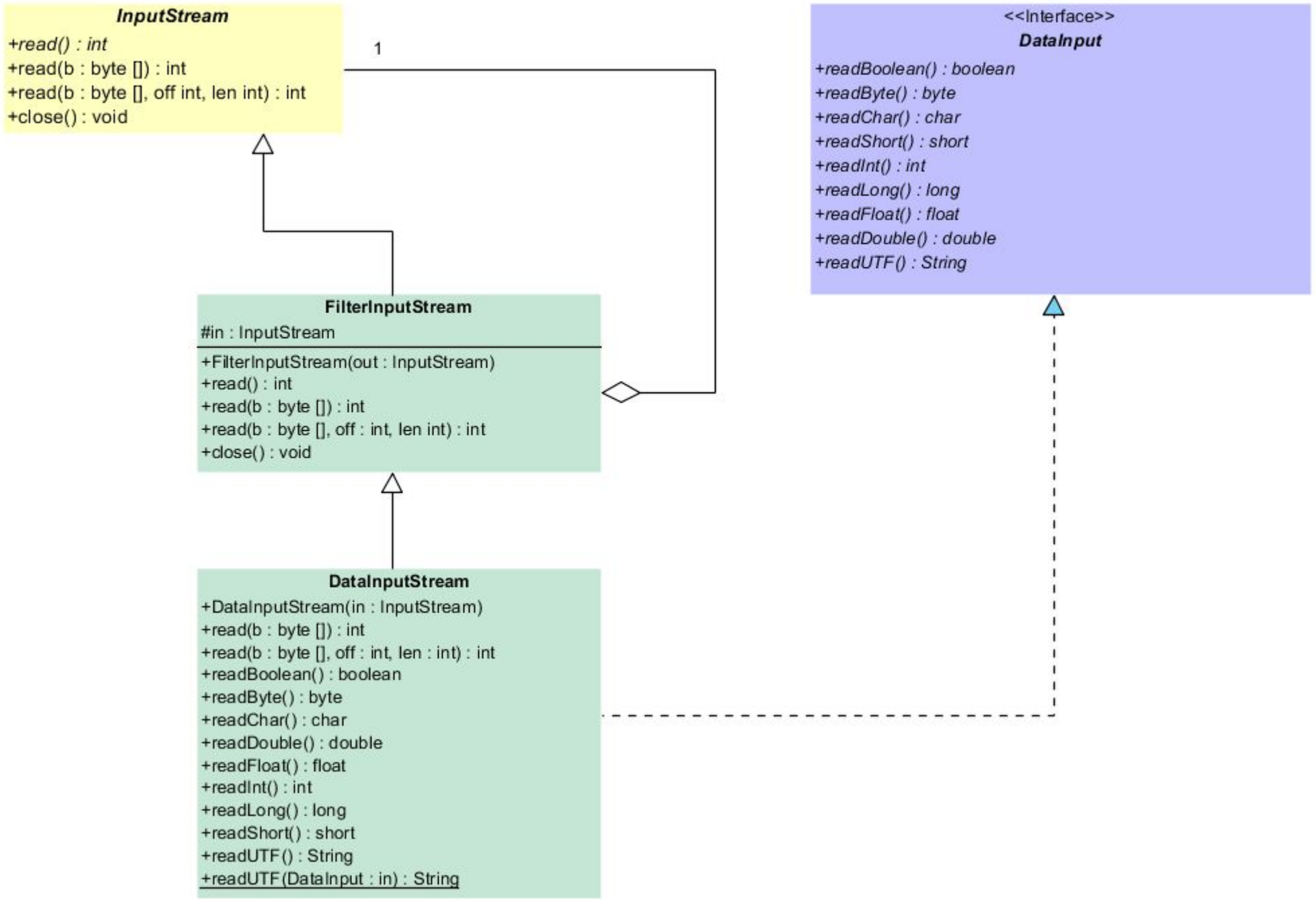
Класс DataOutputStream

```
public class DataOutputStream extends FilterOutputStream {

    protected int written;
    private byte[] bytearr = null;

    public final int size()

    public DataOutputStream(OutputStream out)

    private void incCount(int value)

    public final void writeBoolean(boolean v)
    public final void writeByte(int v)
    public final void writeShort(int v)
    public final void writeChar(int v)
    public final void writeInt(int v)
    public final void writeLong(long v)
    public final void writeFloat(float v)
    public final void writeDouble(double v)

    public final void writeBytes(String s)
    public final void writeChars(String s)
    public final void writeUTF(String str)
    static int writeUTF(String str, DataOutput out)

    public void flush()
}
```

C	Класс DataOutputStream предназначен для превращения примитивов и строк в последовательность байтов и запись этой последовательности в байтовый поток вывода. Байтовый поток вывода задаётся в конструкторе. Класс позволяет хранить и получать число записанных байт.

Потоки ввода данных

Класс DataInputStream



Класс DataInputStream

```
public class DataInputStream extends FilterInputStream implements DataInput {

    public DataInputStream(InputStream in)

    public final int read(byte b[])
    public final int read(byte b[], int off, int len)

    public final boolean readBoolean()
    public final byte readByte()
    public final short readShort()
    public final int readInt()
    public final long readLong()
    public final float readFloat()
    public final double readDouble()
    public final String readUTF()
    public final static String readUTF(DataInput in)
}
```

Класс DataInputStream предназначен для чтения примитивов и строк из байтового потока ввода. Байтовый поток ввода задаётся в конструкторе.

Запись и чтение значений примитива int

Запись int в файл

DIASOFT
всё по-настоящему

```
public class WriteIntDemo {

   public static void main(String[] args) {

      String file = "I:\\FileIO\\writeint.txt";

      FileOutputStream fos = null;
      DataOutputStream dos = null;

      try {

         fos = new FileOutputStream(file);
         dos = new DataOutputStream(fos);

         for (int i = 0; i < 64; i++) {
            dos.writeInt(i);
            System.out.println(Integer.toBinaryString(i) + " ");
         }
      } catch (IOException e) {
         System.out.println("An I/O error occured");
      } finally {
         try {
            if (dos != null) {
               dos.close();
            }
         }
         catch (IOException e) {
            System.out.println("Error closing file");
         }
      }
   }
}
```

Запись int в файл

```
0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
10000
...
110001
110010
110011
110100
110101
110110
110111
111000
111001
111010
111011
111100
111101
111110
111111
```

I:\FileIO\WriteInt.txt - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

WriteInt.txt

Address	0	4	8	c	Dump
00000000	00000000000000000000000000000000	00000000000000000000000000000001	00000000000000000000000000000010	00000000000000000000000000000011	....
00000010	00000000000000000000000000000100	00000000000000000000000000000101	00000000000000000000000000000110	00000000000000000000000000000111	....
00000020	00000000000000000000000000001000	00000000000000000000000000001001	00000000000000000000000000001010	00000000000000000000000000001011	....
00000030	00000000000000000000000000001100	00000000000000000000000000001101	00000000000000000000000000001110	00000000000000000000000000001111	....
00000040	00000000000000000000000000010000	00000000000000000000000000010001	00000000000000000000000000010010	00000000000000000000000000010011	....
00000050	00000000000000000000000000010100	00000000000000000000000000010101	00000000000000000000000000010110	00000000000000000000000000010111	....
00000060	00000000000000000000000000011000	00000000000000000000000000011001	00000000000000000000000000011010	00000000000000000000000000011011	....
00000070	00000000000000000000000000011100	00000000000000000000000000011101	00000000000000000000000000011110	00000000000000000000000000011111	....
00000080	00000000000000000000000000100000	00000000000000000000000000100001	00000000000000000000000000100010	00000000000000000000000000100011	... ...!..."...#
00000090	00000000000000000000000000100100	00000000000000000000000000100101	00000000000000000000000000100110	00000000000000000000000000100111	...$...%...&...'
000000a0	00000000000000000000000000101000	00000000000000000000000000101001	00000000000000000000000000101010	00000000000000000000000000101011	...(...)...*...+
000000b0	00000000000000000000000000101100	00000000000000000000000000101101	00000000000000000000000000101110	00000000000000000000000000101111	...,...-......./
000000c0	00000000000000000000000000110000	00000000000000000000000000110001	00000000000000000000000000110010	00000000000000000000000000110011	...0...1...2...3
000000d0	00000000000000000000000000110100	00000000000000000000000000110101	00000000000000000000000000110110	00000000000000000000000000110111	...4...5...6...7
000000e0	00000000000000000000000000111000	00000000000000000000000000111001	00000000000000000000000000111010	00000000000000000000000000111011	...8...9...:...;
000000f0	00000000000000000000000000111100	00000000000000000000000000111101	00000000000000000000000000111110	00000000000000000000000000111111	...<...=...>...?

Hex Edit View | nb char : 256 | Ln : 1 Col : 1 Sel : 0 | Binary | BigEndian | INS

Чтение int из файла

DIASOFT
всё по-настоящему

```
public class ReadIntDemo {

   public static void main(String[] args) {

      String file = "I:\\FileIO\\WriteInt.txt";

      FileInputStream fis = null;
      DataInputStream dis = null;

      try {

         fis = new FileInputStream(file);
         dis = new DataInputStream(fis);

         int temp;

         for (int i = 0; i < 64; i++) {
            temp = dis.readInt();
            System.out.println(temp);
         }
      } catch (IOException e) {
         System.out.println("An I/O error occured");
      } finally {
         try {
            if (dis != null) {
               dis.close();
            }
         }
         catch (IOException e) {
            System.out.println("Error closing file");
         }
      }
   }
}
```

Чтение int из файла

```
0
1
2
3
4
5
6
7
8
9
10
11
...
50
51
52
53
54
55
56
57
58
59
60
61
62
63
```

Запись и чтение примитивов и строки

Запись данных в файл

DIASOFT
всё по-настоящему

```
public class WriteDataDemo {

   public static void main(String[] args) {

      String file = "I:\\FileIO\\writedata.dat";

      FileOutputStream fos = null;
      DataOutputStream dos = null;

      String name = "Harry Hacker";
      char gender = 'm';
      boolean isMarried = true;
      byte numChildren = 2;
      short yearOfBirth = 1987;
      int salary = 30000;
      long netAsset = 8234567890L;
      double weight = 88.88;
      float gpa = 4.58f;

      System.out.println("Name: " + name);
      System.out.println("Gender: " + gender);
      System.out.println("Is married: " + isMarried);
      System.out.println("Number of children: " + numChildren);
      System.out.println("Year of birth: " + yearOfBirth);
      System.out.println("Salary: " + salary);
      System.out.println("Net Asset: " + netAsset);
      System.out.println("Weight: " + weight);
      System.out.println("GPA: " + gpa);

      ...
   }
}
```

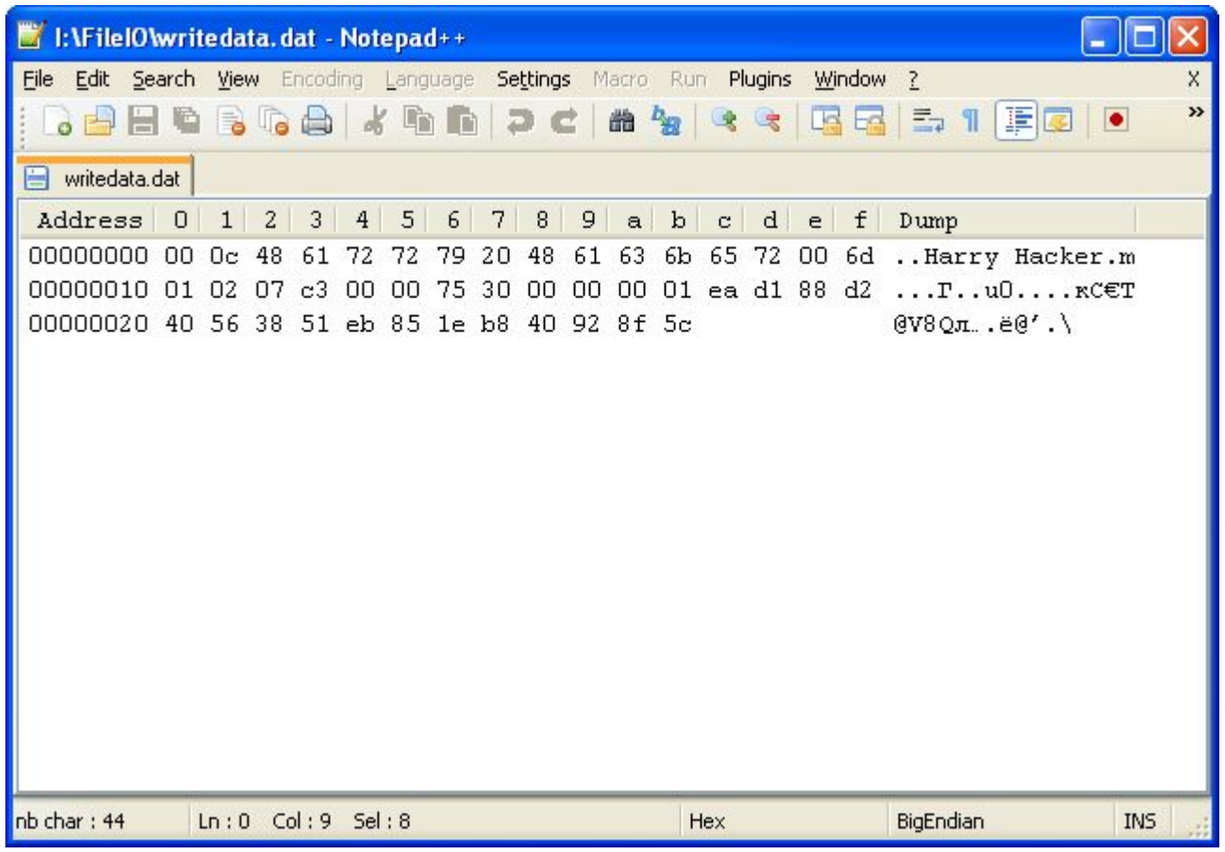
Запись данных в файл

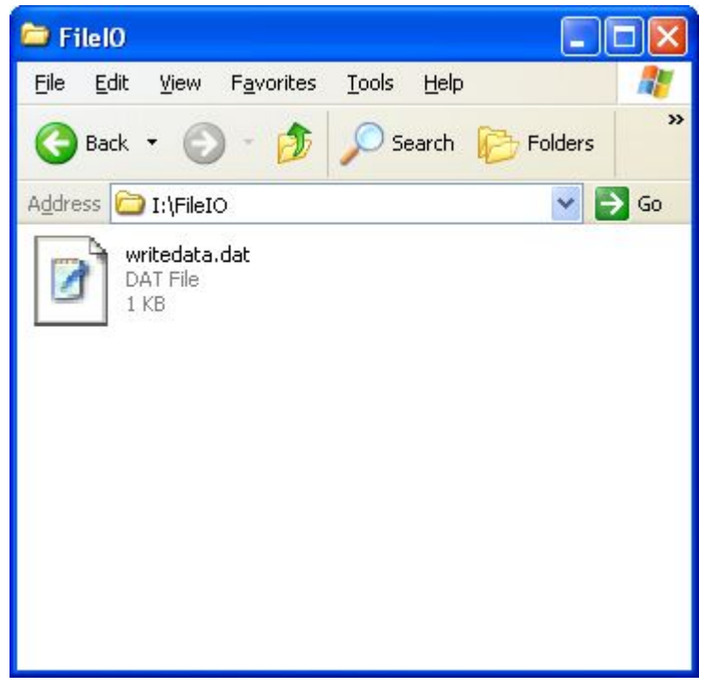
DIASOFT
всё по-настоящему

```
public class WriteDataDemo {

   public static void main(String[] args) {

      ...

      try {

         fos = new FileOutputStream(file);
         dos = new DataOutputStream(fos);

         dos.writeUTF(name);
         dos.writeChar(gender);
         dos.writeBoolean(isMarried);
         dos.writeByte(numChildren);
         dos.writeShort(yearOfBirth);
         dos.writeInt(salary);
         dos.writeLong(netAsset);
         dos.writeDouble(weight);
         dos.writeFloat(gpa);

      } catch (IOException e) {
         System.out.println("An I/O error occured");
      } finally {
         try {
            if (dos != null) {
               dos.close();
            }
         } catch (IOException e) {
            System.out.println("Error closing file");
         }
      }
   }
}
```

Запись данных в файл

```
Name: Harry Hacker
Gender: m
Is married: true
Number of children: 2
Year of birth: 1987
Salary: 30000
Net Asset: 8234567890
Weight: 88.88
GPA: 4.58
```





Чтение данных из файла

DIASOFT
всё по-настоящему

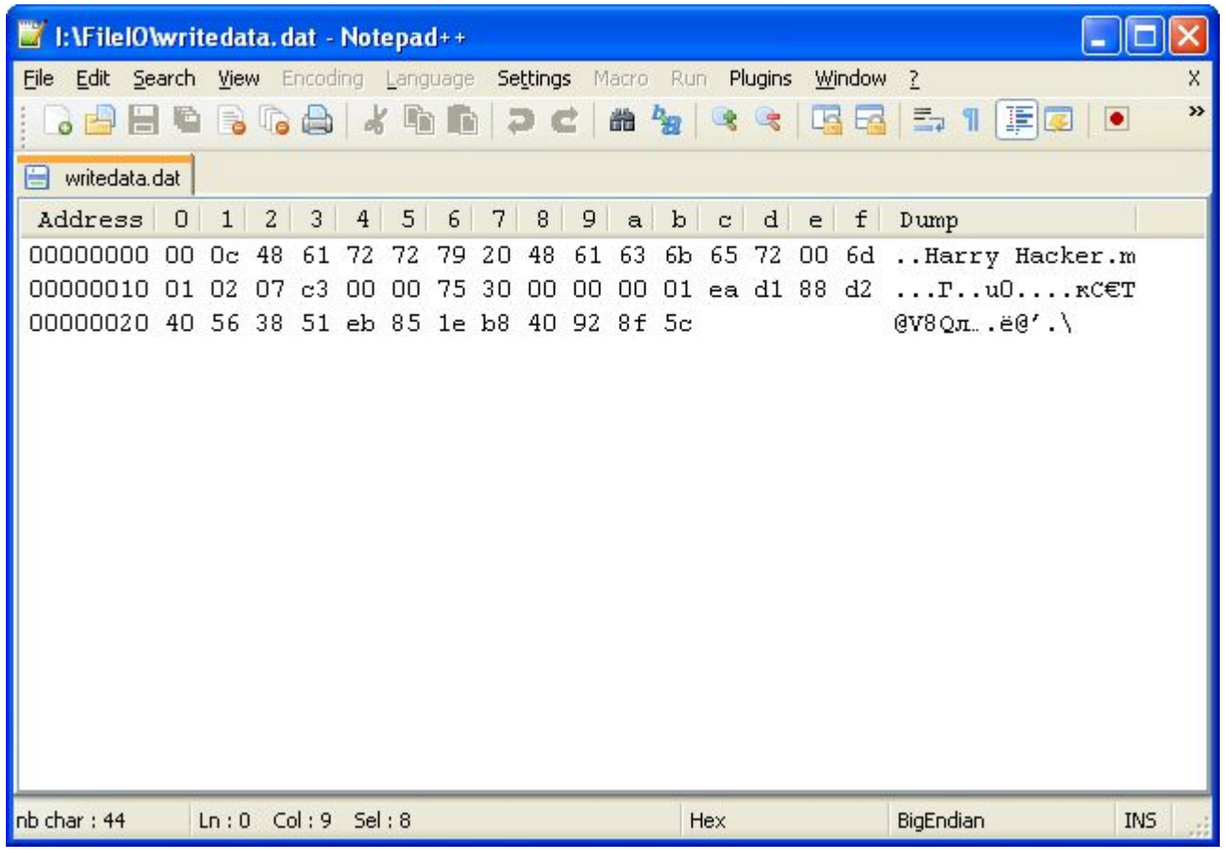
```
public class ReadDataDemo {

   public static void main(String[] args) {

      String file = "I:\\FileIO\\writedata.txt";

      FileInputStream fis = null;
      DataInputStream dis = null;

      try {

         ...

      } catch (IOException e) {
         System.out.println("An I/O error occured");
      } finally {
         try {
            if (dis != null) {
               dis.close();
            }
         } catch (IOException e) {
            System.out.println("Error closing file");
         }
      }
   }
}
```

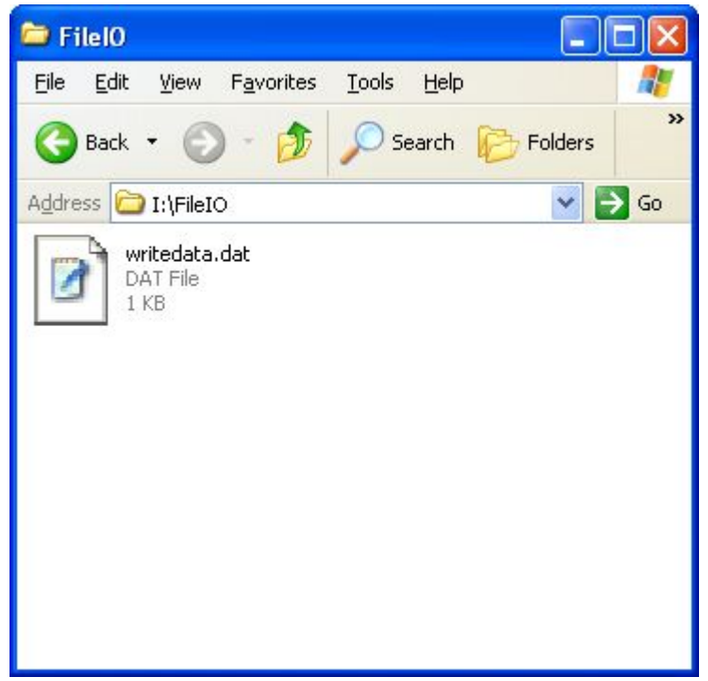
Чтение данных из файла

DIASOFT
всё по-настоящему

```
public class ReadDataDemo {

   public static void main(String[] args) {

   ...
         fis = new FileInputStream(file);
         dis = new DataInputStream(fis);

         String name = dis.readUTF();
         char gender = dis.readChar();
         boolean isMarried = dis.readBoolean();
         byte numChildren = dis.readByte();
         short yearOfBirth = dis.readShort();
         int salary = dis.readInt();
         long netAsset = dis.readLong();
         double weight = dis.readDouble();
         float gpa = dis.readFloat();

         System.out.println("Name: " + name);
         System.out.println("Gender: " + gender);
         System.out.println("Is married: " + isMarried);
         System.out.println("Number of children: " + numChildren);
         System.out.println("Year of birth: " + yearOfBirth);
         System.out.println("Salary: " + salary);
         System.out.println("Net Asset: " + netAsset);
         System.out.println("Weight: " + weight);
         System.out.println("GPA: " + gpa);

   ...

   }
}
```

Чтение данных из файла






```
Name: Harry Hacker
Gender: m
Is married: true
Number of children: 2
Year of birth: 1987
Salary: 30000
Net Asset: 8234567890
Weight: 88.88
GPA: 4.58
```

Запись и чтение состояния объекта

Запись полей класса в файл

DIASOFT
всё по-настоящему

```
class Employee {

   public Employee() {
   }

   public Employee(String name, short yearOfBirth, char gender,
         boolean isMarried, int salary) {

      this.name = name;
      this.yearOfBirth = yearOfBirth;
      this.gender = gender;
      this.isMarried = isMarried;
      this.salary = salary;
   }

   public String toString() {
      return "Employee [name=" + name + ", yearOfBirth=" + yearOfBirth
            + ", gender=" + gender + ", isMarried=" + isMarried + ", salary="
            + salary + "]";
   }
   ...

   private String name;
   private short yearOfBirth;
   private char gender;
   private boolean isMarried;
   private int salary;
}
```

Запись полей класса в файл

DIASOFT
всё по-настоящему

```
class Employee {

   ...

   public void writeState(String file) {

      FileOutputStream fos = null;
      DataOutputStream dos = null;

      try {

         fos = new FileOutputStream(file);
         dos = new DataOutputStream(fos);

         dos.writeUTF(name);
         dos.writeShort(yearOfBirth);
         dos.writeChar(gender);
         dos.writeBoolean(isMarried);
         dos.writeInt(salary);

      } catch (IOException e) {
         System.out.println("An I/O error occured");
      } finally {
         try {
            if (dos != null) {
               dos.close();
            }
         } catch (IOException e) {
            System.out.println("Error closing file");
         }
      }
   }
}
```

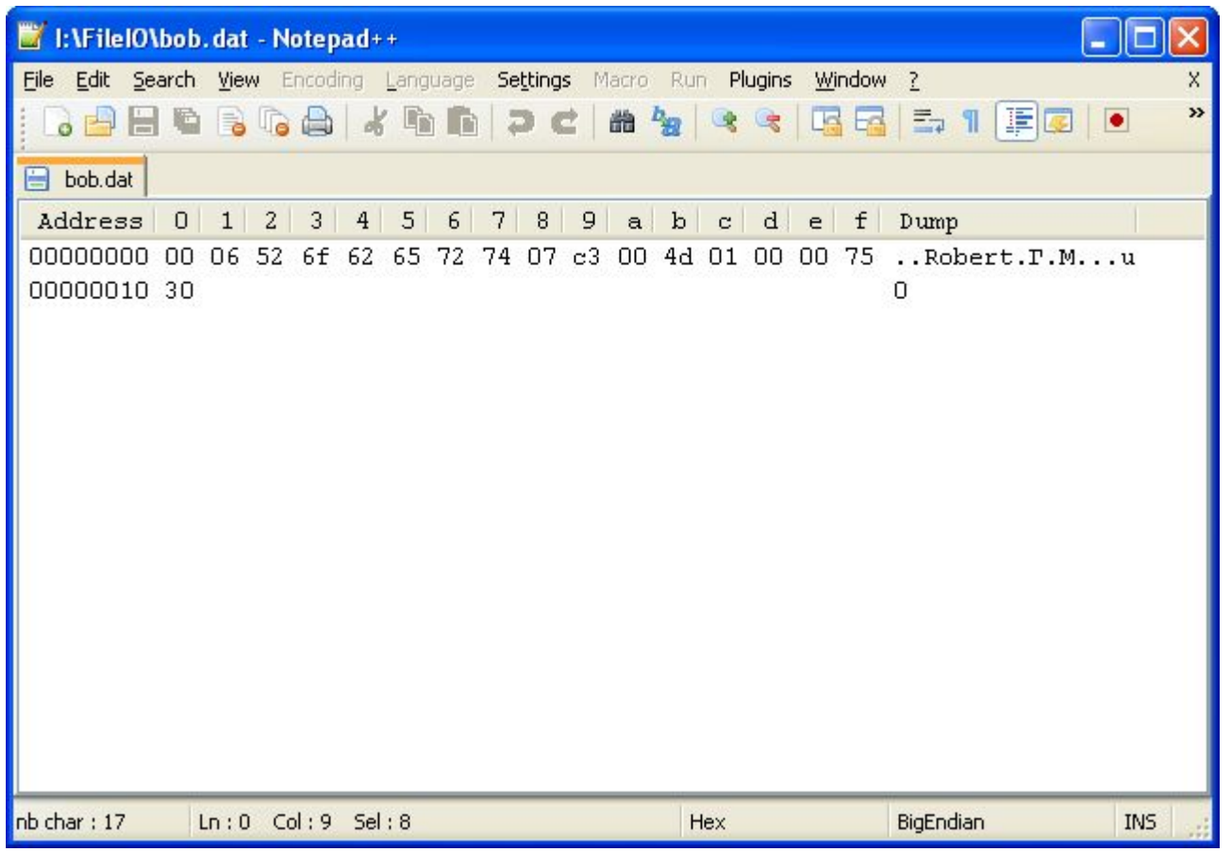
Запись полей класса в файл

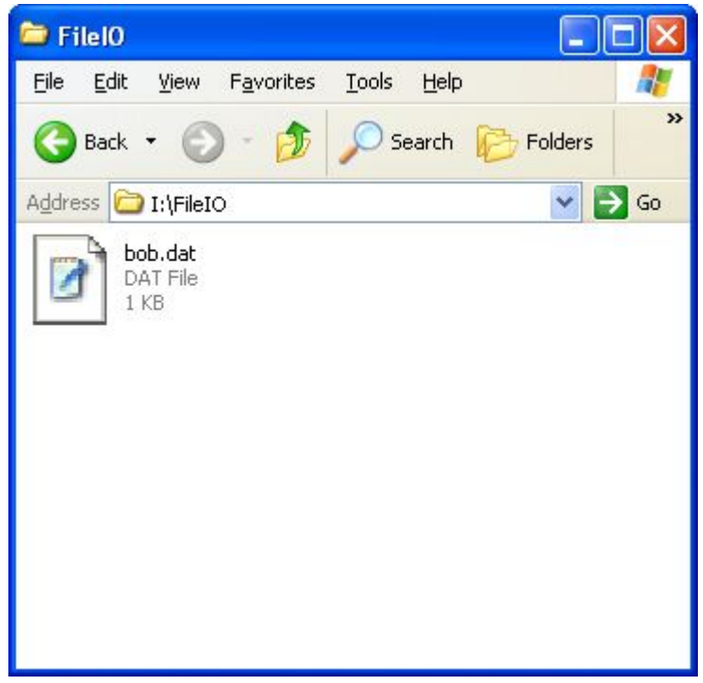
DIASOFT
всё по-настоящему

```
public class DataDemo {

   public static void main(String[] args) {

      Employee bob = new Employee("Robert", (short) 1987, 'M', true, 30000);
      System.out.println(bob);
      bob.writeState("bob.dat");

      ...
   }
}
```

Запись полей класса в файл

```
Employee [name=Robert, yearOfBirth=1987, gender=M, isMarried=true, salary=30000]
```





Чтение полей класса из файла

DIASOFT
всё по-настоящему

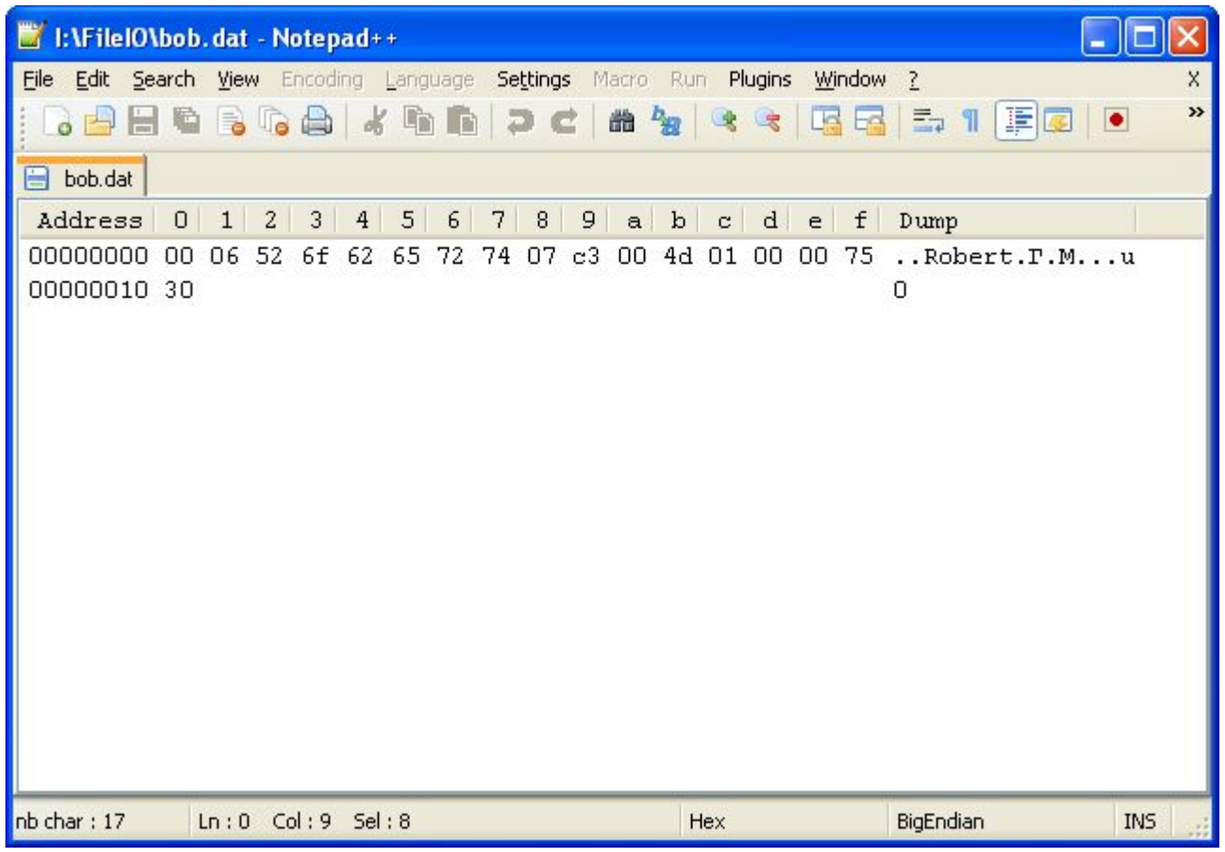
```
class Employee {

   ...

    public void readState(String file) {

      FileInputStream fis = null;
      DataInputStream dis = null;

      try {

         fis = new FileInputStream(file);
         dis = new DataInputStream(fis);

         name = dis.readUTF();
         yearOfBirth = dis.readShort();
         gender = dis.readChar();
         isMarried = dis.readBoolean();
         salary = dis.readInt();

      } catch (IOException e) {
         System.out.println("An I/O error occured");
      } finally {
         try {
            if (dis != null) {
               dis.close();
            }
         } catch (IOException e) {
            System.out.println("Error closing file");
         }
      }
   }
}
```

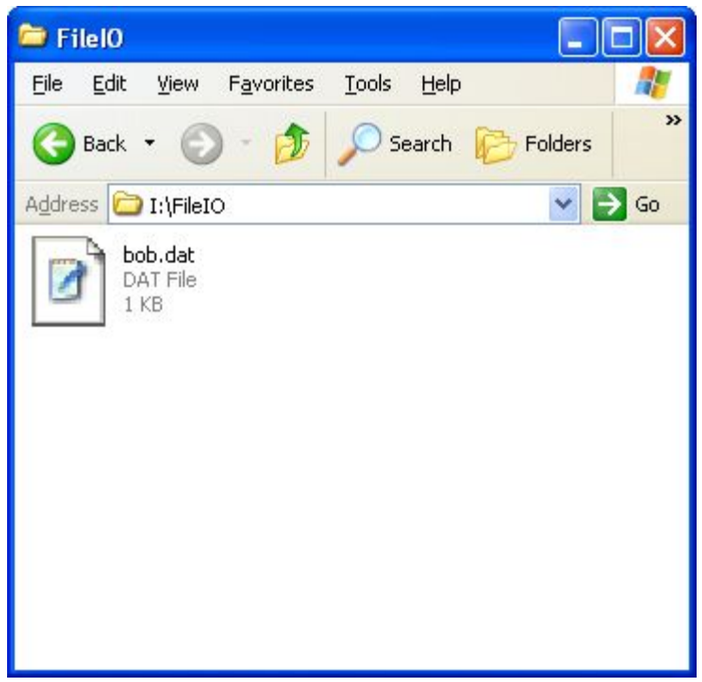
Чтение полей класса из файла

DIASOFT
всё по-настоящему

```
public class DataDemo {

   public static void main(String[] args) {


      ...

      Employee robert = new Employee();
      System.out.println(robert);

      robert.readState("bob.dat");
      System.out.println(robert);
   }
}
```

Чтение полей класса из файла






```
Employee [name=null, yearOfBirth=0, gender= , isMarried=false, salary=0]
Employee [name=Robert, yearOfBirth=1987, gender=M, isMarried=true, salary=30000]
```

Спасибо

Россия, 127018,
Москва, ул. Полковая 3, стр. 14
Тел.: +7(495) 780 7575, 789 9339
Факс: +7(495) 780 7576, 789 9338
info@diasoft.ru, www.diasoft.ru