

## java.util.Date

Класс **Date** хранит время в миллисекундах начиная с 1 января 1970 года.

```
import java.util.Date;
public class Main {
    public static void main(String[] args) {
        Date date = new Date();
        // ВЫВОД ДАТЫ В КОНСОЛЬ
        System.out.println(date.toString());
    }
}
```

С помощью метода **getTime()** можно отобразить количество миллисекунд, прошедших с 1 января 1970 года.

```
import java.util.Date;

public class Main {

    public static void main(String[] args) {
        // С помощью метода getTime()
        // можно отобразить количество миллисекунд,
        // прошедших с 1 января 1970 года.

        Date date = new Date();
        long millis = date.getTime();
        System.out.println(String.valueOf(millis));
    }
}
```

## Вывод времени выполнения программы на экран

```
import java.util.Date;
//... подключение библиотеки для измерения времени

Date date = new Date();
//... действия

Date date1 = new Date();
System.out.println((date1.getTime() - date.getTime()));
```

## Задание

Выведите на экран (и запишите в тетрадь в виде таблицы) время выполнения следующих действий:

1	<pre>long s=0; // задавайте n последовательно 100, 1000, ... 100000 for (long i=1 ;i&lt; n;i++){     s+=i; }</pre>
2	<pre>// задавайте n последовательно 100, 1000, ... 100000 for (long i=1 ;i&lt; n;i++){     System.out.println(i); }</pre>
3	<pre>long a=0; // задавайте n последовательно 100, 1000, ... 100000 for (long i=1 ;i&lt; n;i++){     System.out.println(i);     if (a==i) { break;} }</pre>

## FileOutputStream — запись байтов в файл

```
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException {

        File file = new File("d:\\test_01.txt");
        FileOutputStream fileOutputStream = new FileOutputStream(file);

        String greetings = "Привет!!!";

        fileOutputStream.write(greetings.getBytes());

        fileOutputStream.close();

    }
}
```

```
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException {

        FileOutputStream fileOutputStream = new FileOutputStream("d:\\test_01.txt");

        String greetings = "Привет!!!";

        fileOutputStream.write(greetings.getBytes());

        fileOutputStream.close();

    }
}
```

### Дозапись в файл

```
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException {

        FileOutputStream fileOutputStream = new FileOutputStream("d:\\test_02.txt", true);

        String greetings = "Привет! \n";

        fileOutputStream.write(greetings.getBytes());

        fileOutputStream.close();

    }
}
```

## Задания

1. С помощью потока **FileOutputStream** выполните запись в файл нескольких строк, введенных с клавиатуры.

**Подсказка.** Ввод строки с помощью клавиатуры

```
import java.util.Scanner;

...
Scanner in = new Scanner(System.in);
System.out.print("Input a string: ");
String str = in.nextLine();
```

2. Выполните дозапись двух строк в текстовый файл из предыдущего задания.

## FileInputStream— чтение байтов из файла

```
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException {

        FileInputStream fileInputStream = new FileInputStream("d:\\test_02.txt");

        int i;

        while((i=fileInputStream.read())!= -1){

            System.out.print((char)i);

        }

    }

}
```

### Задание

1. Прочитайте содержимое текстового файла из предыдущего задания, выведите содержимое на экран.

## BufferedInputStream

Считывает данные не по одному байту, а блоками и временно хранит их в специальном буфере. Это позволяет оптимизировать работу программы за счет уменьшения количества обращений к файлу.

```
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException {

        FileInputStream fileInputStream = new FileInputStream("d:\\test_02.txt");

        BufferedInputStream bufferedInputStream = new BufferedInputStream(fileInputStream, 200);

        int i;

        while((i = bufferedInputStream.read())!= -1){

            System.out.print((char)i);

        }

    }

}
```

### Задания

1. Прочитайте содержимое текстового файла из предыдущего задания, выведите содержимое на экран.

2. Выполните в одной программе сначала запись в новый файл нескольких строк, а затем чтение из этого файла.

3. Выведите на экран время записи ста случайных чисел в файл, а затем чтения этих чисел из файла.

## Измерение времени чтения и вывода на экран содержимого текстового файла

```
import java.io.*;
import java.util.Date;

public class Main {

    public static void main(String[] args) throws IOException {

        Date date = new Date();

        FileInputStream fileInputStream = new FileInputStream("d:\\textBook.rtf");

        BufferedInputStream bufferedInputStream = new BufferedInputStream(fileInputStream);

        int i;

        while((i = bufferedInputStream.read()) != -1){

            System.out.print((char)i);

        }

        Date date1 = new Date();

        System.out.println((date1.getTime() - date.getTime()));

    }

}
```

### Задание

1. Исследуйте – как зависит время исполнения программы от размера **BufferedInputStream**.

## Копирование из одного файла в другой

```
import java.io.*;

public class Main {

    public static void main(String args[]) throws IOException {
        FileInputStream fileIn = null;
        FileOutputStream fileOut = null;

        try {
            fileIn = new FileInputStream("D:\\file.txt");
            fileOut = new FileOutputStream("D:\\copied_file.txt");

            int a;
            // Копирование содержимого файла file.txt
            while ((a = fileIn.read()) != -1) {
                fileOut.write(a);
            }
            // Чтение содержимого файла file.txt и запись в файл
            // copied_file.txt
        } finally {
            if (fileIn != null) {
                fileIn.close();
            }
            if (fileOut != null) {
                fileOut.close();
            }
        }
    }

}
```

### Задание

1. Оцените – как зависит время копирования содержимого файла от его размера.