Вводный курс в Java
Занятие 5

Александр Русин
e-mail: alexander.rusin@simbirsoft.com

Android Developer
ООО СимбирСофт

String

Представляет константную строку символов. Для строк, значение которых может меняться в процессе выполнения программы следует использовать класс StringBuffer или StringBuilder.

String

public String()

public String(String value)

public String(char[] value)

public String(char[] value, int offset, int count)

public String(byte[] bytes, int off, int len, String enc)

public String(byte[] bytes, String enc)

public String(byte[] bytes, int off, int len)

public String(byte[] bytes)

public String(StringBuffer buffer)

- Позволяют создать новую строку.

String

- public int length() – возвращает длину строки в символах.
- public char charAt(int index) – возвращает символ в заданной позиции (от 0).
- public void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin) – копирует символы строки в массив символов.
- public byte[] getBytes(String enc) или public byte[] getBytes() – переводит строку в массив байт с использованием заданной кодировки.

String

- public boolean equals(Object anObject) – сравнение строк на совпадение.
- public int compareTo(String anotherString) – лексикографическое сравнение строк.
- Public boolean regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len) – проверяет подстроки на совпадение.

SIMBIRSOFT
software development

String

- public int indexOf(int ch)
- public int indexOf(int ch, int fromIndex)
- public int indexOf(String str)
- public int indexOf(String str, int fromIndex)
- public int lastIndexOf(int ch)
- public int lastIndexOf(int ch, int fromIndex)
- Возвращает позицию первого/последнего (или первого/последнего с/до fromIndex) вхождения символа или подстроки в строку.

String

- public String substring(int beginIndex) или
- public String substring(int beginIndex, int endIndex) – возвращает подстроку.
- public String concat(String str) – конкатенация строк.
- public String replace(char oldChar, char newChar) – возвращает строку после подстановки символа.
- public String toLowerCase() или
- public String toUpperCase() – возвращает строку с приведением символов к заданному регистру.

String

- public String trim() – устраняет начальные и конечные пробелы в строке.
- public char[] toCharArray() – создает массив символов.

String

- public static String valueOf(Object obj)
- public static String valueOf(char[] data)
- public static String valueOf(char[] data, int offset, int count)
- public static String valueOf(boolean b)
- public static String valueOf(char c)
- public static String valueOf(int i)
- public static String valueOf(long l)
- Строковое представление аргумента.

StringBuffer

SIMBIRSOFT software development

- Представляет строку символов, длина и содержимое которой может изменяться. Как правило, возвращаемое значение – ссылка на сам объект StringBuffer.
- public StringBuffer() или
- public StringBuffer(int length) или
- public StringBuffer(String str) – создает новый строковый буфер. Исходная длина по умолчанию – 16 символов.

StringBuffer

- public int length() – длина строки в символах.
- public int capacity() – текущая ёмкость буфера для хранения строки.
- public void ensureCapacity(int minimumCapacity) – установить минимальную ёмкость буфера. Новая ёмкость будет не менее заданной величины и не менее удвоенного старого объема + 2.
- public void setLength(int newLength) – установить длину строки. Строка обрезается или дополняется символами с кодом 0 (не пробелами).

SimbirSoft
software development

StringBuffer

- public char charAt(int index) – находит символ в заданной позиции.
- public void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin) – получить подстроку в массив символов.
- public void setCharAt(int index, char ch) – изменить указанный символ.

SIMBIRSOFT
software development

StringBuffer

- public StringBuffer append(Object obj)
- public StringBuffer append(String str)
- public StringBuffer append(char[] str)
- public StringBuffer append(char[] str, int offset, int len)
- public StringBuffer append(boolean b)
- public StringBuffer append(char c)
- public StringBuffer append(int i)
- public StringBuffer append(long l)
- Присоединить строковое значение объекта к строке.

StringBuffer

- public StringBuffer delete(int start, int end)
- public StringBuffer deleteCharAt(int index)
- Удаление подстроки или уменьшением длины строки.

StringBuffer

public StringBuffer insert(int offset, Object obj)

public StringBuffer insert(int offset, String str)

public StringBuffer insert(int offset, char[] str)

public StringBuffer insert(int offset, boolean b)

public StringBuffer insert(int offset, char c)

public StringBuffer insert(int offset, int i)

public StringBuffer insert(int offset, long l)

- Вставка символьного представления объекта в строку.

StringBuffer

- public StringBuffer reverse() – инвертирует положение символов в строке.
- public String toString() – создает новый объект String.

File

- Работа с файлами и каталогами через класс File
- Использование потоков для посимвольного или побайтового вывода.
- Чтение/запись файла
- Форматированный вывод
- Чтение данных с консоли
- Бинарный ввод/вывод в файл

File Class

- Объектом класса File может быть файл или каталог

```
File file1 = new File("data.txt");
File file1 = new File("C:\java");
```

SIMBIRSOFT
software development

Методы

- isFile/isDirectory
- canRead/canWrite
- length – Возвращает длину файла в байтах или 0, если файл не существует.
- list – если объект File директория, то возварщает массив строк из заголовков файлов, находящихся в данной директории; или null
- mkdir – Создает новую директорию.
- delete – Удаляет директорию и возвращает true, если все прошло удачно.
- toURL – Конвертирует путь в URL объект.

SIMBIRSOFT software development

Директория. Пример

```
public class DirListing {
   public static void main(String[] args) {
      File dir = new
File(System.getProperty("user.dir"));
      if (dir.isDirectory()) {
         System.out.println("Directory of " + dir);
         String[] listing = dir.list();
         for (int i = 0; i < listing.length; i++) {
            System.out.println("\t" + listing[i]);
         }
      }
   }
}
```

Директория. Результат

- Directory of /home/alexander/kursJava/lec_5_1
  - src
  - bin
  - .settings
  - .project
  - .classpath

Ввод/Вывод

- В пакете java.io содержится около 60 классов для I/O
- Потоками можно считывать байты или символы
  - Использовать DataStreams для чтения байт I/O
  - Использовать Readers and Writers для посимвольного чтения I/O
    - Для символьный потока I/O можно указать кодировку
- Стоит обратить внимание, что IOException может произойти во время любой операции ввода вывода.

Запись символа

Что	Метод	Конструктор
Character File Ouput	**FileWriter** write(int char) write(byte[] buffer) write(String str)	File file = new File("filename"); FileWriter fout = new FileWriter(file); или FileWriter fout = new FileWriter("filename");

Запись символа

Что	Метод	Конструктор
Buffered Character File Output	**BufferedWriter** write(int char) write(char[] buffer) write(String str) newLine()	File file = new File("filename"); FileWriter fout = new FileWriter(file); BufferedWriter bout = new BufferedWriter(fout); или BufferedWriter bout = new BufferedWriter(new FileWriter(new File("filename")));

Запись символа

Что	Метод	Конструктор

Character Output	**PrintWriter**
	write(int char)
	write(char[] buffer)
	writer(String str)
	print(...)
	println(...)
		FileWriter fout = new FileWriter("filename");
		PrintWriter pout = new PrintWriter(fout);
		или
		PrintWriter pout = new PrintWriter(
		new FileWriter("filename"));
		или
		PrintWriter pout = new PrintWriter(
		new BufferedWriter(new FileWriter("filename")));

FileWriter

- Конструктор
  - – FileWriter(String filename)/FileWriter(File file)
    - Создает выходной поток, используя кодировку по умолчанию
  - – FileWriter(String filename, boolean append)
    - Создает новый поток вывода или добавляет к существующим выходной поток (append = true)
- Методы
  - write(String str)/write(char[] buffer) - Записывает строку или массив символов в файл
  - write(int char) - записывает символ (int) в файл
  - flush - записывает любой буфер символов в файл
  - close - закрывает файловый поток после выполнения flush
  - getEncoding - возвращает кодировку

Кодировка. Пример

```
public class CharacterFileOutput {
    public static void main(String[] args) {
        FileWriter out = null;
        try {
            out = new FileWriter("book.txt");
            System.out.println("Encoding: " + out.getEncoding());
            out.write("Core Programming");
            out.close();
            out = null;
        } catch (IOException ioe) {
            System.out.println("IO problem: " + ioe);
            ioe.printStackTrace();
            try {
                if (out != null) {
                out.close();
            }
            } catch (IOException ioe2) {
            }
        }
    }
}
```

Форматированный вывод

- Создать объект DecimalFormat, описывающий форматирование.
  - DecimalFormat formatter = new DecimalFormat("#,###.##");
- Используя метод format конвертировать значение в строку
  - formatter.format(24.99);

SIMBIRSOFT
software development

Символы форматирования

Символ	Значение
0	Место для цифры.
#	Место для цифры. Если цифра в начале или конце нулю, то не отображаются.
.	Место запятой.
,	Указывает место запятой.
-	Знак минус.
E	Указывает место, чтобы отделить мантиссу от экспоненты.
%	Отображает в виде процентов

SIMBIRSOFT
software development

Пример

```
public class NumFormat {
    public static void main(String[] args) {
    DecimalFormat science = new DecimalFormat("0.000E0");
    DecimalFormat plain = new DecimalFormat("0.0000");
    for (double d = 100.0; d < 140.0; d *= 1.10) {
        System.out.println("Scientific: " + science.format(d)
                                + " and Plain: " + plain.format(d));
    }
    }
}
```

Чтение из файла

Что	Метод	Конструктор
Character File Input	**FileReader** read() read(char[] buffer)	File file = new File("filename"); FileReader fin = new FileReader(file); или FileReader fin = new FileReader("filename");

Чтение из файла

Что	Метод	Конструктор
Buffered Character File Input	**BufferedReader** read() read(char[] buffer) readLine()	File file = new File("filename"); FileReader fin = new FileReader(file); BufferedReader bin = new BufferedReader(fin); или BufferedReader bin = new BufferedReader(new FileReader(new File("filename")));

FileReader

- Конструктор
  - FileReader(String filename)/FileReader(File file)
    - Создает входной поток, используя кодировку по умолчанию
- Методы
  - read/read(char[] buffer) - Читает один символ или массив символов, возвращает -1 когда достигнут конец потока(файла)
  - reset - возврат к началу потока (файла)

SIMBIRSOFT
software development

Пример

```
public class CharacterFileInput {
    public static void main(String[] args) {
        File file = new File("book.txt");
        FileReader in = null;
        if (file.exists()) {
            try {
                in = new FileReader(file);
            System.out.println("Encoding: " + in.getEncoding());
            char[] buffer = new char[(int) file.length()];
            in.read(buffer);
            System.out.println(buffer);
            in.close();
            } catch (IOException ioe) {
                System.out.println("IO problem: " + ioe);
                ioe.printStackTrace();
            }
        }
    }
}
```

Пример

- Построчное считывание.

```
BufferedReader in = new BufferedReader(new FileReader(file));
String lineIn;
while ((lineIn = in.readLine()) != null) {
    System.out.println(lineIn);
}
```

Консольный ввод

- Чтобы читать ввод с консоли, поток должны быть связаны со стандартным вводом, System.in

```
public class IOInput {
    public static void main(String[] args) {
        BufferedReader keyboard;
        String line;
        try {
            System.out.print("Enter value: ");
            System.out.flush();
            keyboard = new BufferedReader(new
        InputStreamReader(System.in));
            line = keyboard.readLine();
            System.out.println("value  = " + line);
        } catch (IOException e) {
            System.out.println("Error reading input!");
        }
    }
}
```

Бинарный ввод/вывод

DataInputStream или DataOutputStream

DataType	DataInputStream	DataOutputStream
byte	readByte	writeByte
int	readInt	writeInt
long	readLong	writeLong
double	readDouble	writeDouble
boolean	readBoolean	writeBoolean
char	readChar	writeChar
String	readUTF	writeUTF

Бинарный вывод

Что	Метод	Конструктор
Binary File Output bytes	**FileOutputStream** write(byte) write(byte[] buffer)	File file = new File("filename"); FileOutputStream fout = new FileOutputStream(file); или FileOutputStream fout = new FileOutputStream("filename");

Бинарный вывод

Что	Метод	Конструктор
Binary File Output byte short int long float double char boolean	**DataOutputStream** writeByte(byte) writeShort(short) writeInt(int) writeLong(long) writeFloat(float) writeDouble(double) writechar(char) writeBoolean(boolean) writeUTF(string) writeChars(string)	File file = new File("filename"); FileOutputStream fout = new FileOutputStream(file); DataOutputStream dout = new DataOutputStream(fout); или DataOutputStream dout = new DataOutputStream(new FileOutputStream(new File("filename")));

Бинарный вывод

Что	Метод	Конструктор
Buffered Binary File Output	**BufferedOutputStream** flush() write(byte) write(byte[] buffer, int off, int len)	File file = new File("filename"); FileOutputStream fout = new FileOutputStream(file); BufferedOutputStream bout = new BufferedOutputStream(fout); DataOutputStream dout = new DataOutputStream(bout); или DataOutputStream dout = new DataOutputStream(new BufferedOutputStream(new FileOutputStream(new File("filename"))));

SIMBIRSOFT
software development

Пример

```
public class BinaryFileOutput {
   public static void main(String[] args) {
      int[] primes = { 1, 2, 3, 5, 11, 17, 19, 23 };
      DataOutputStream out = null;
      try {
      out = new DataOutputStream(new
FileOutputStream("primes.bin"));
         for (int i = 0; i < primes.length; i++) {
            out.writeInt(primes[i]);
      }
      out.close();
      } catch (IOException ioe) {
         System.out.println("IO problem: " + ioe);
         ioe.printStackTrace();
      }
   }
}
```

Бинарный ввод

Что	Метод	Конструктор
Binary File Input bytes	**FileInputStream** read() read(byte[] buffer)	File file = new File("filename"); FileInputStream fin = new FileInputStream(file); или FileInputStream fin = new FileInputStream("filename");
Binary File Input byte short int long float double char boolean	**DataInputStream** readByte() readShort() readInt() readLong() readFloat() readDouble() readchar() readBoolean() readUTF() readFully(byte[] buffer)	File file = new File("filename"); FileInputStream fin = new FileInputStream(file); DataInputStream din = new DataInputStream(fin); или DataInputStream din = new DataInputStream(new FileInputStream(new File("filename")));

Бинарный ввод

Что	Метод	Конструктор
Binary File Input byte short int long float double char boolean	**DataInputStream** readByte() readShort() readInt() readLong() readFloat() readDouble() readchar() readBoolean() readUTF() readFully(byte[] buffer)	File file = new File("filename"); FileInputStream fin = new FileInputStream(file); DataInputStream din = new DataInputStream(fin); или DataInputStream din = new DataInputStream(new FileInputStream(new File("filename")));

Бинарный ввод

Что	Метод	Конструктор
Bufferred Binary File Input	**BufferedInputStream** read() read(byte[] buffer, int off, int len) skip(long)	File file = new File("filename"); FileInputStream fin = new FileInputStream(file); BufferedInputStream bin = new BufferedInputStream(fin); DataInputStream din = new DataInputStream(bin); или DataInputStream din = new DataInputStream(new BufferedInputStream(new FileInputStream(new File("filename"))));

SIMBIRSOFT
software development

Пример

```
public class BinaryFileInput {
   public static void main(String[] args) {
      DataInputStream in = null;
      File file = new File("primes.bin");
      try {
         in = new DataInputStream(new
FileInputStream(file));
         int prime;
         long size = file.length() / 4; // 4 bytes per int
         for (long i = 0; i < size; i++) {
            prime = in.readInt();
            System.out.println(prime);
         }
         in.close();
      } catch (IOException ioe) {
         System.out.println("IO problem: " + ioe);
         ioe.printStackTrace();
      }
   }
}
```

SIMBIRSOFT
software development

Пример

```
public class BinaryFileInput {
   public static void main(String[] args) {
      DataInputStream in = null;
      File file = new File("primes.bin");
      try {
         in = new DataInputStream(new
FileInputStream(file));
         int prime;
         long size = file.length() / 4; // 4 bytes per int
         for (long i = 0; i < size; i++) {
            prime = in.readInt();
            System.out.println(prime);
         }
         in.close();
      } catch (IOException ioe) {
         System.out.println("IO problem: " + ioe);
         ioe.printStackTrace();
      }
   }
}
```

SIMBIRSOFT
software development

Домашняя задача

1 Создать бинарный файл dic. В котором будут хранится слова для поиска.

2 Загрузить словарь (бинарный файл dic).

3 Загрузить текстовый файл in.txt с текстом в котором будет происходить поиск.

4 Вывести в текстовый файл out.txt количество количество совпадений слова и само слово.

SIMBIRSOFT
software development

Спасибо за внимание!