Java IOs

https://github.com/heig-vd-dai-course

Web · PDF

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

Based on the original course by O. Liechti and J. Ehrensberger.

Objectives

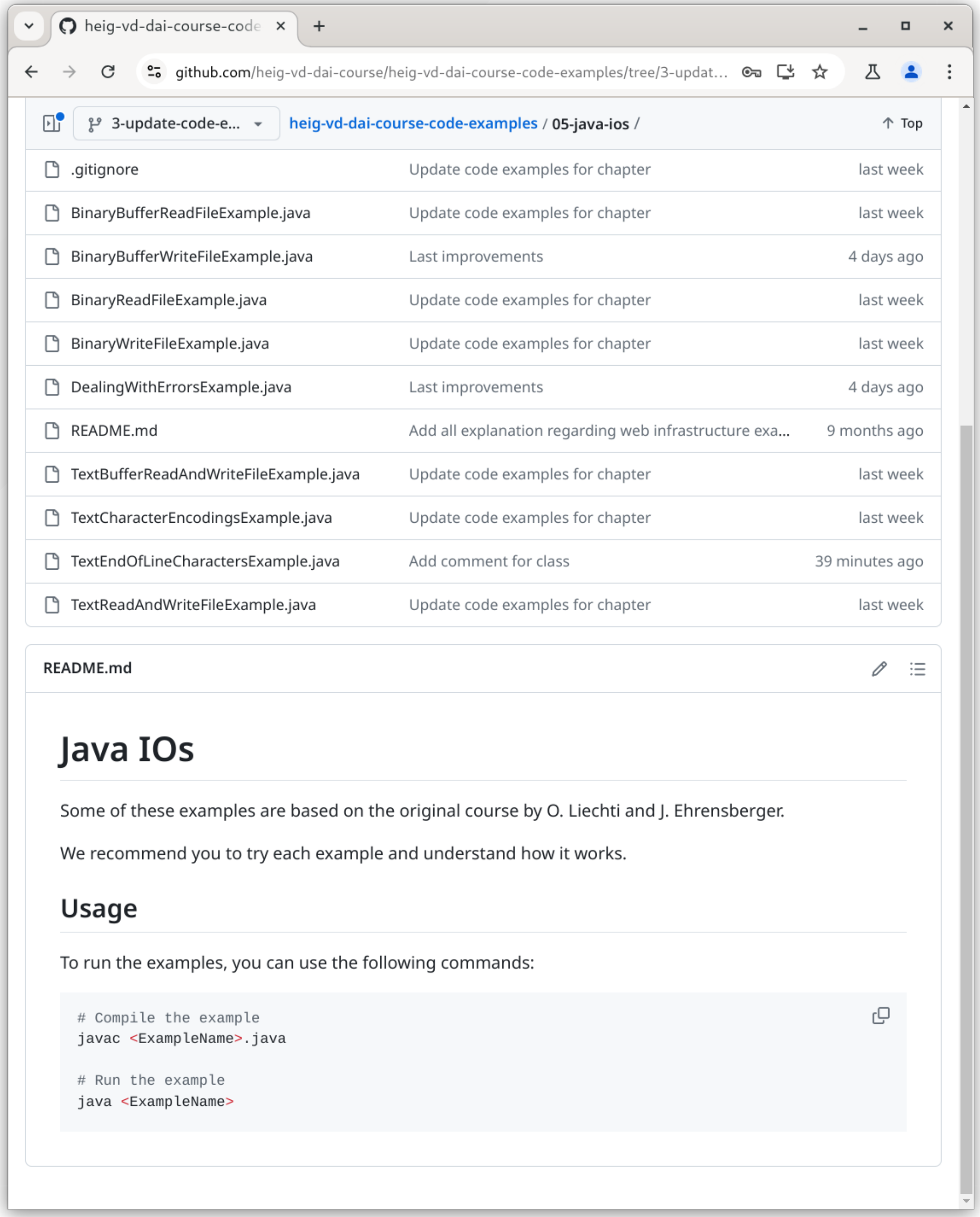
- Know the different types of data (binary vs. text)
- Understand the abstract notion of sources, streams and sink
- Use the different IO types for different use-cases
- Use the Java IO API to read and write files

Prepare and setup your environment

More details for this section in the course material. You can find other resources and alternatives as well.

Check and run the code examples

- Check the code examples
- Run the code examples
- Help to understand the concepts
- Play with the code examples



heig-vd-dai-course-code
github.com/heig-vd-dai-course/heig-vd-dai-course-code-examples/tree/3-updat...
3-update-code-e...
heig-vd-dai-course-code-examples / 05-java-ios /
Top

.gitignore	Update code examples for chapter	last week
BinaryBufferReadFileExample.java	Update code examples for chapter	last week
BinaryBufferWriteFileExample.java	Last improvements	4 days ago
BinaryReadFileExample.java	Update code examples for chapter	last week
BinaryWriteFileExample.java	Update code examples for chapter	last week
DealingWithErrorsExample.java	Last improvements	4 days ago
README.md	Add all explanation regarding web infrastructure exa...	9 months ago
TextBufferReadAndWriteFileExample.java	Update code examples for chapter	last week
TextCharacterEncodingsExample.java	Update code examples for chapter	last week
TextEndOfLineCharactersExample.java	Add comment for class	39 minutes ago
TextReadAndWriteFileExample.java	Update code examples for chapter	last week

README.md

Java IOs

Some of these examples are based on the original course by O. Liechti and J. Ehrensberger.

We recommend you to try each example and understand how it works.

Usage

To run the examples, you can use the following commands:

```
# Compile the example
javac <ExampleName>.java

# Run the example
java <ExampleName>
```

Sources, streams and sinks of data

More details for this section in the course material. You can find other resources and alternatives as well.

Sources, streams and sinks of data

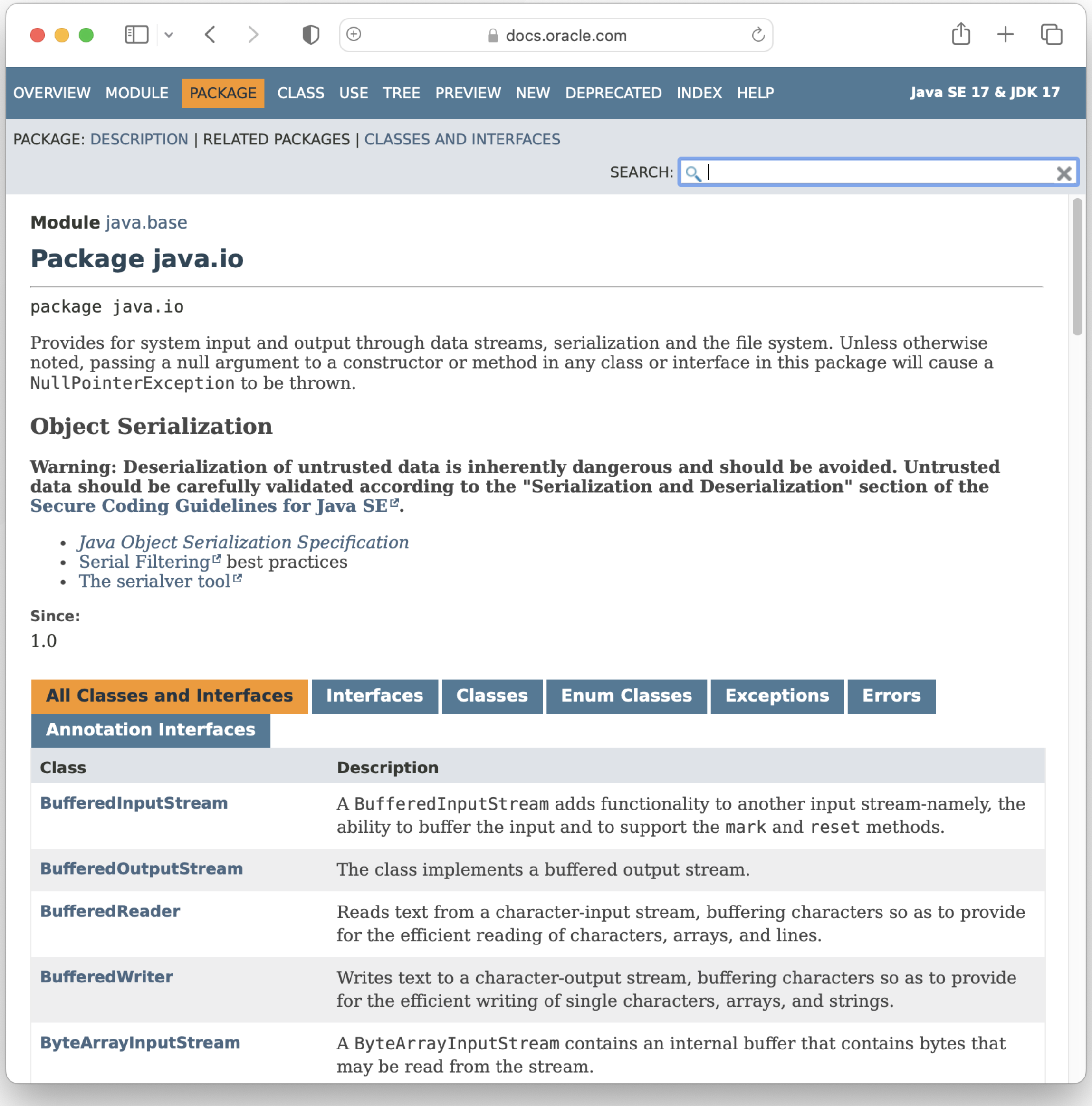
- Abstraction of data flow
- Source: where data comes from (input)
- Sink: where data goes to (output)
- Stream: data flows between source and sink

The Java IO API

More details for this section in the course material. You can find other resources and alternatives as well.

The Java IO API

- Part of `java.base` module
  - `java.io` package
  - `java.nio` package
- Different classes for different IO types:
  - Binary data
  - Text data



docs.oracle.com
OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP
Java SE 17 & JDK 17
PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES
SEARCH:

Module java.base

Package java.io

```
package java.io
```

Provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a `NullPointerException` to be thrown.

Object Serialization

Warning: Deserialization of untrusted data is inherently dangerous and should be avoided. Untrusted data should be carefully validated according to the "Serialization and Deserialization" section of the Secure Coding Guidelines for Java SE.

- *Java Object Serialization Specification*
- Serial Filtering best practices
- The serialver tool

Since:
1.0

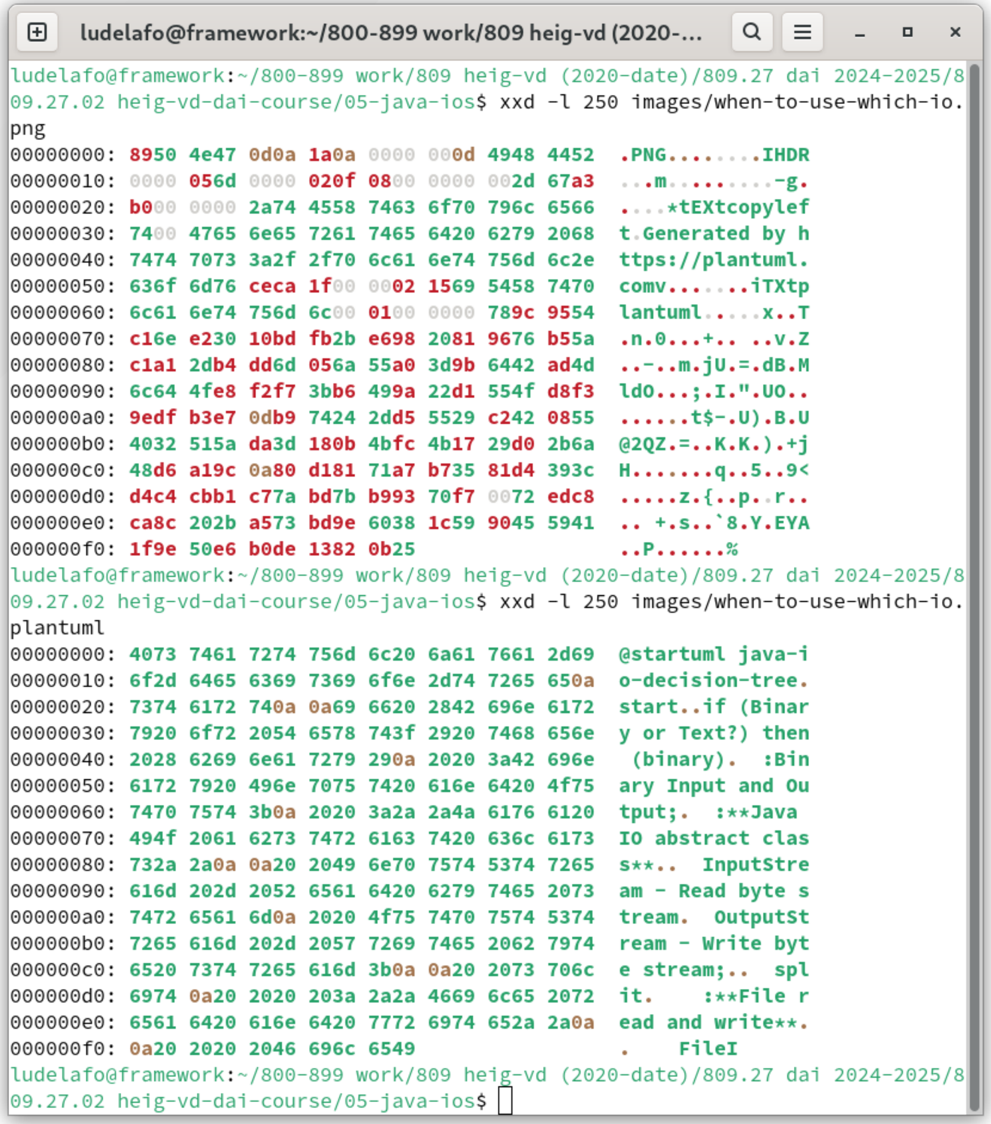
Class	Description
BufferedInputStream	A `BufferedInputStream` adds functionality to another input stream-namely, the ability to buffer the input and to support the `mark` and `reset` methods.
BufferedOutputStream	The class implements a buffered output stream.
BufferedReader	Reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.
BufferedWriter	Writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.
ByteArrayInputStream	A `ByteArrayInputStream` contains an internal buffer that contains bytes that may be read from the stream.

Types of data

More details for this section in the course material. You can find other resources and alternatives as well.

Types of data

- Two types of data:
  - Binary
  - Text
- Both are `0`s and `1`s - the difference is in interpretation:
  - Binary data - raw data
  - Text data - interpretation



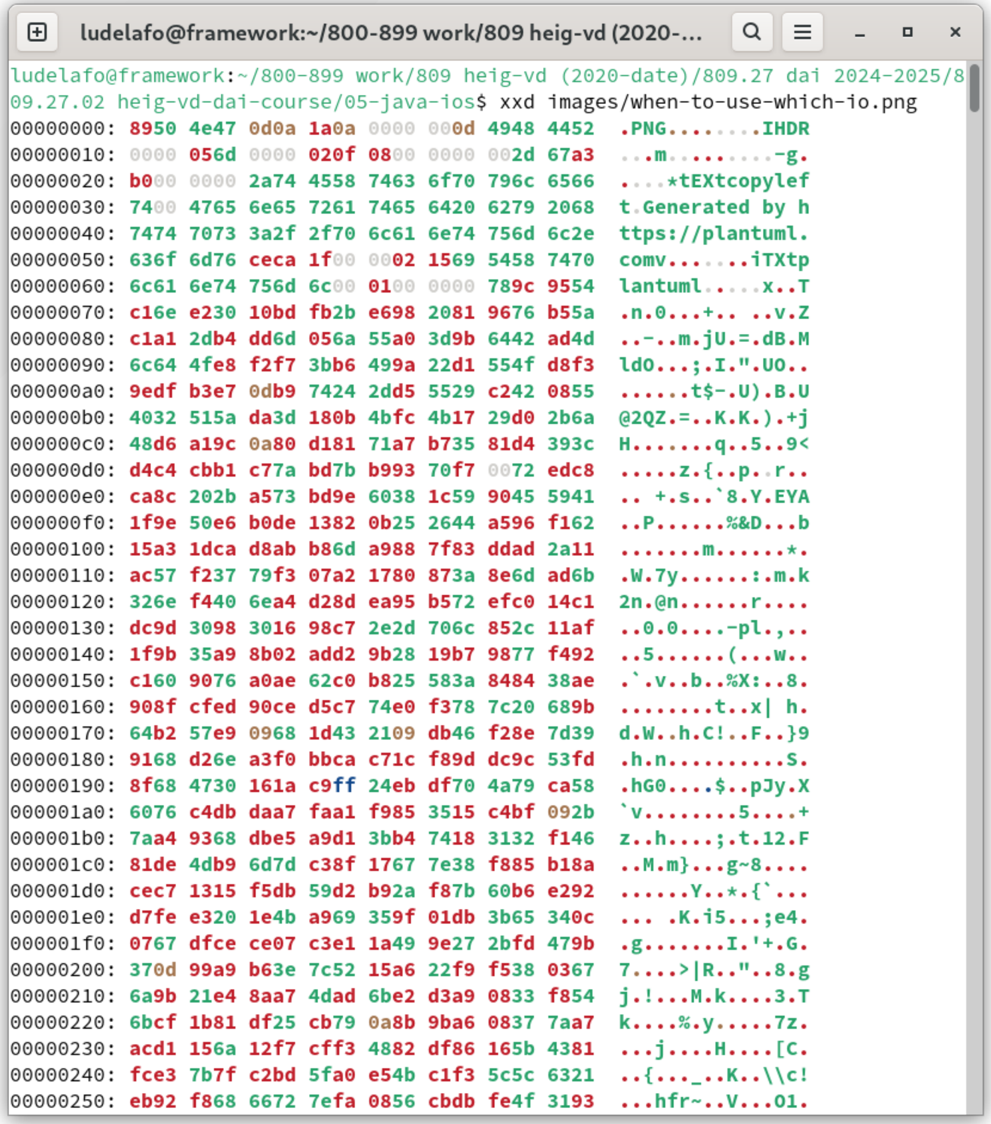
```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd -l 250 images/when-to-use-which-io.
png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25                 ..P......%
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd -l 250 images/when-to-use-which-io.
plantuml
00000000: 4073 7461 7274 756d 6c20 6a61 7661 2d69  @startuml java-i
00000010: 6f2d 6465 6369 7369 6f6e 2d74 7265 650a  o-decision-tree.
00000020: 7374 6172 740a 0a69 6620 2842 696e 6172  start..if (Binar
00000030: 7920 6f72 2054 6578 743f 2920 7468 656e  y or Text?) then
00000040: 2028 6269 6e61 7279 290a 2020 3a42 696e   (binary).  :Bin
00000050: 6172 7920 496e 7075 7420 616e 6420 4f75  ary Input and Ou
00000060: 7470 7574 3b0a 2020 3a2a 2a4a 6176 6120  tput;.  :**Java
00000070: 494f 2061 6273 7472 6163 7420 636c 6173  IO abstract clas
00000080: 732a 2a0a 0a20 2049 6e70 7574 5374 7265  s**..  InputStre
00000090: 616d 202d 2052 6561 6420 6279 7465 2073  am - Read byte s
000000a0: 7472 6561 6d0a 2020 4f75 7470 7574 5374  tream.  OutputSt
000000b0: 7265 616d 202d 2057 7269 7465 2062 7974  ream - Write byt
000000c0: 6520 7374 7265 616d 3b0a 0a20 2073 706c  e stream;..  spl
000000d0: 6974 0a20 2020 203a 2a2a 4669 6c65 2072  it.    :**File r
000000e0: 6561 6420 616e 6420 7772 6974 652a 2a0a  ead and write**.
000000f0: 0a20 2020 2046 696c 6549                 .    FileI
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$
```

Processing binary data with the Java IO API

More details for this section in the course material. You can find other resources and alternatives as well.

Processing binary data with the Java IO API

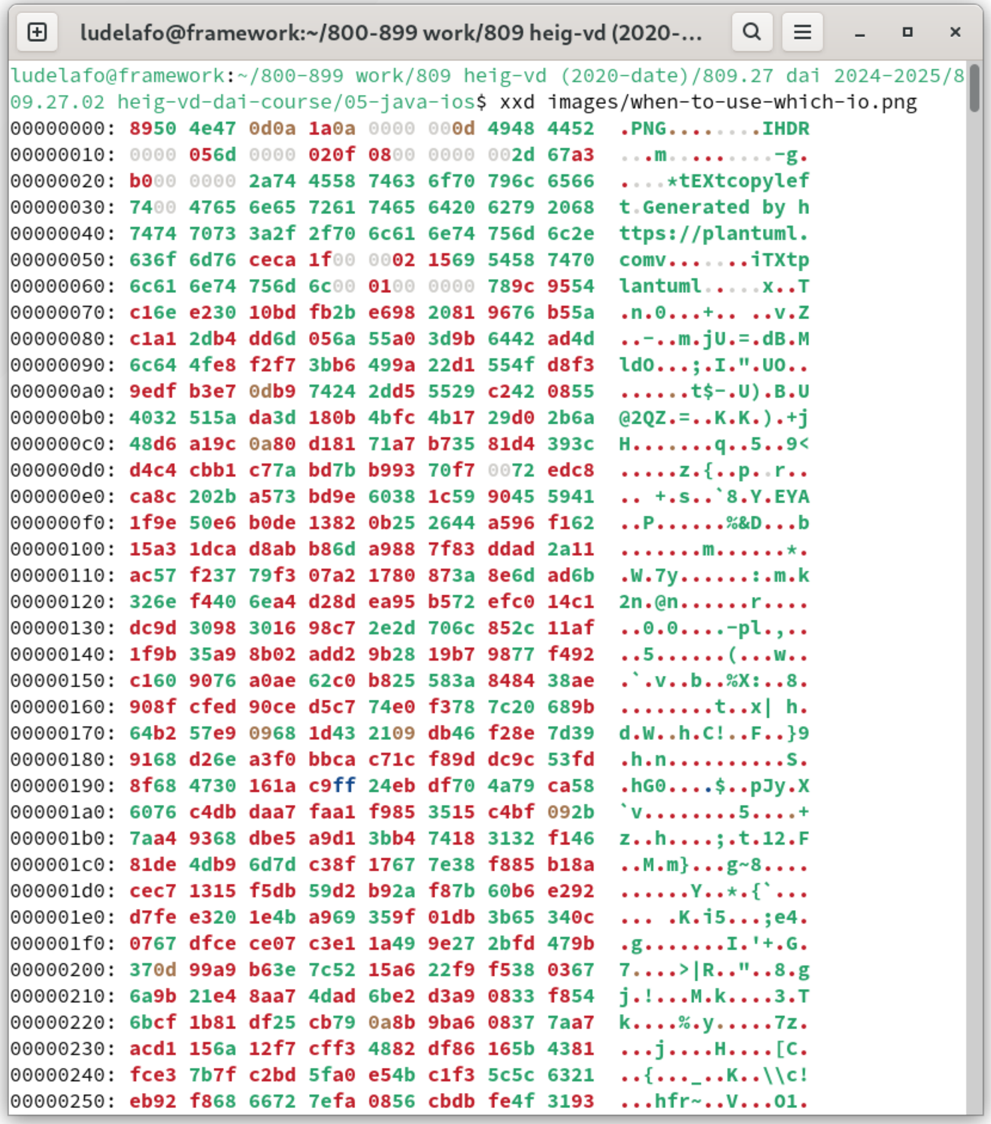
- Most basic type of data processing:
  1. Open a file
  2. Read/write/modify the bytes as they are
  3. Close the file



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R.."..8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

Reading binary data

- Most simple way is to read byte by byte (not efficient)
- `InputStream` and `FileInputStream` classes are used to read binary data
- Let's have a look at the code example `BinaryReadFileExample`

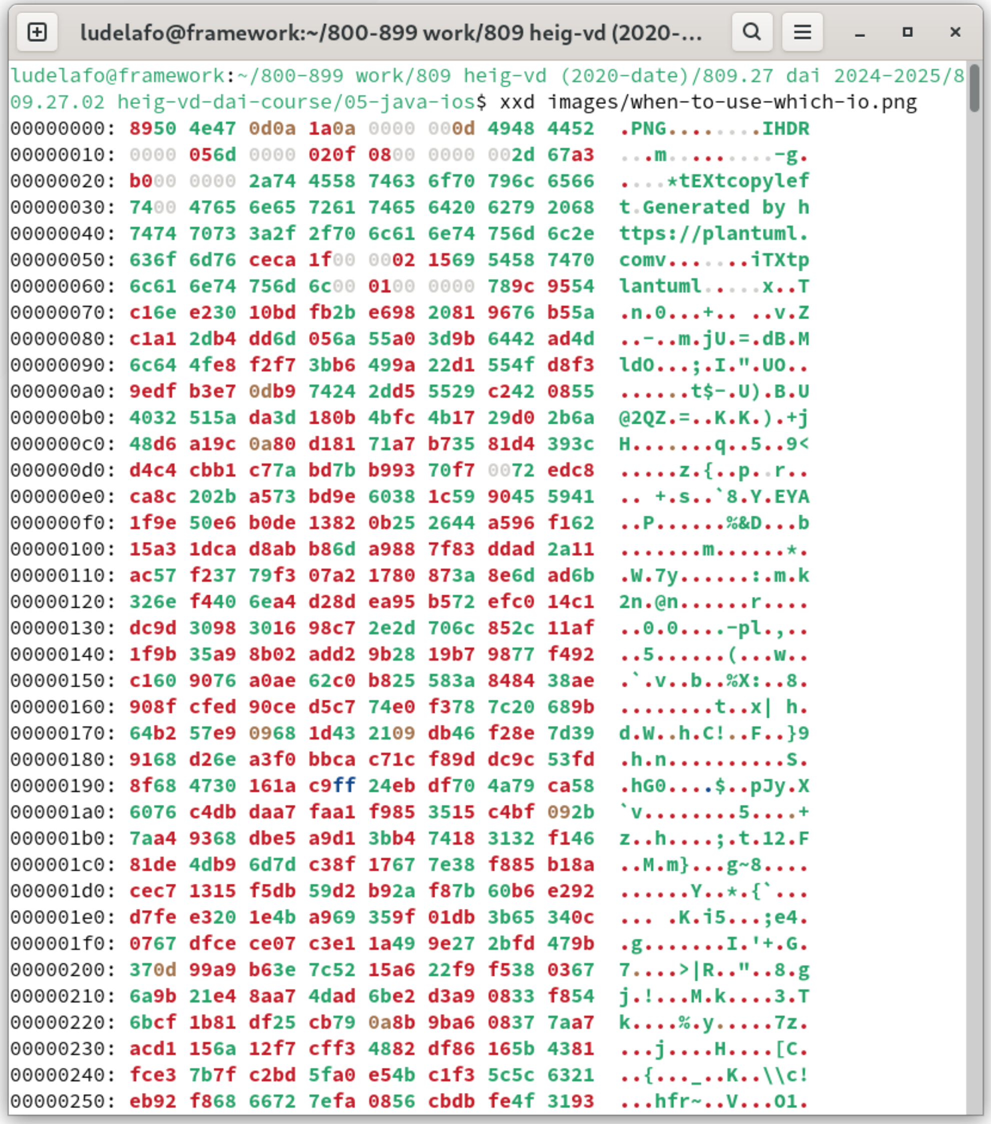


```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R..".8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

```java
class BinaryReadFileExample {

  public static void main(String[] args) throws IOException {
    InputStream fis = new FileInputStream("binary-file.bin");

    // -1 indicates the end of the file
    int b;
    while ((b = fis.read()) != -1) {
      System.out.print(b);
    }

    fis.close();
  }
}
```

Writing binary data

- Most simple way is to write byte by byte (not efficient)
- `OutputStream` and `FileOutputStream` classes are used to write binary data
- Let's have a look at the code example `BinaryWriteFileExample`

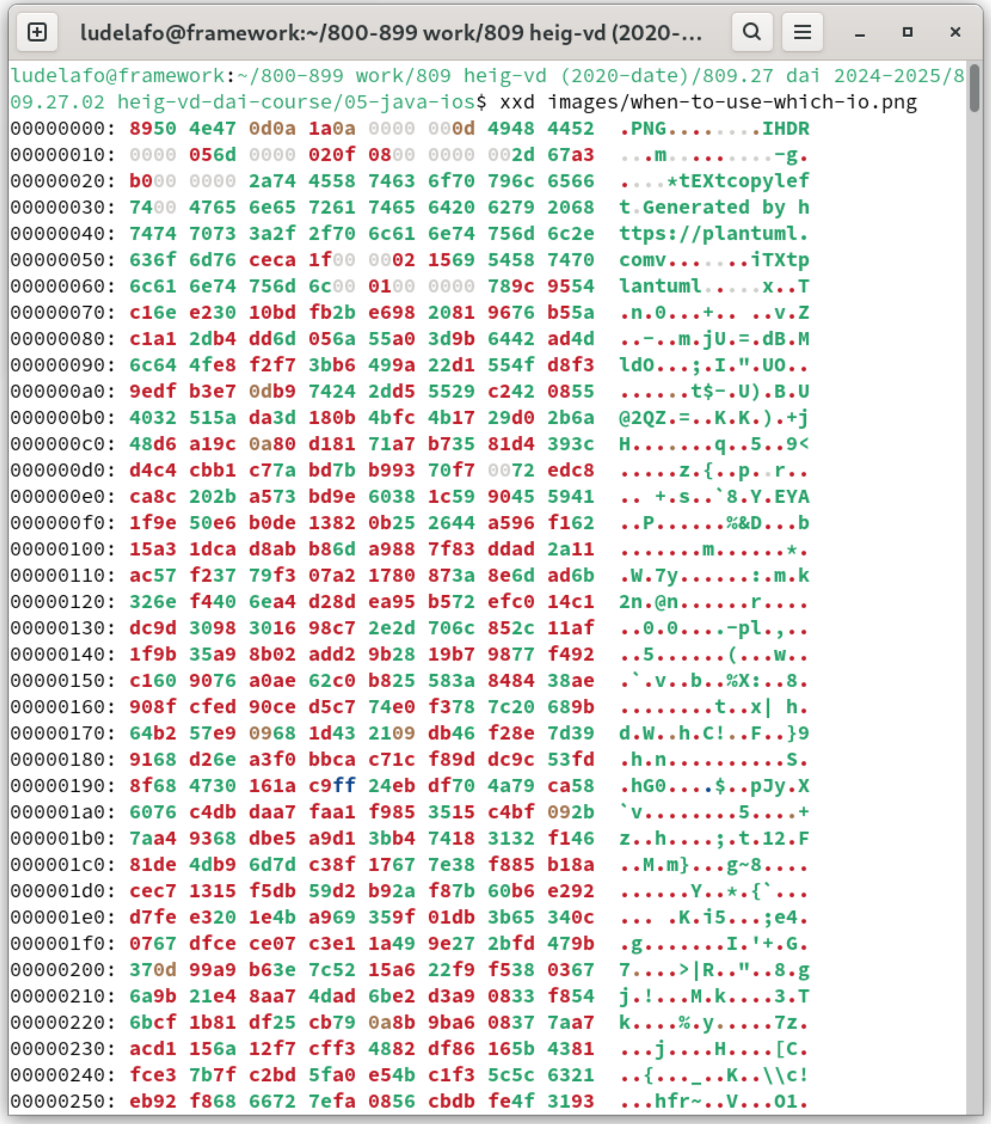


```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R..".8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

```java
class BinaryWriteFileExample {

  public static void main(String[] args) throws IOException {
    OutputStream fos = new FileOutputStream("binary-file.bin");

    for (int i = 0; i < 256; i++) {
      fos.write(i);
    }

    fos.close();
  }
}
```

Reading and writing binary data with buffers

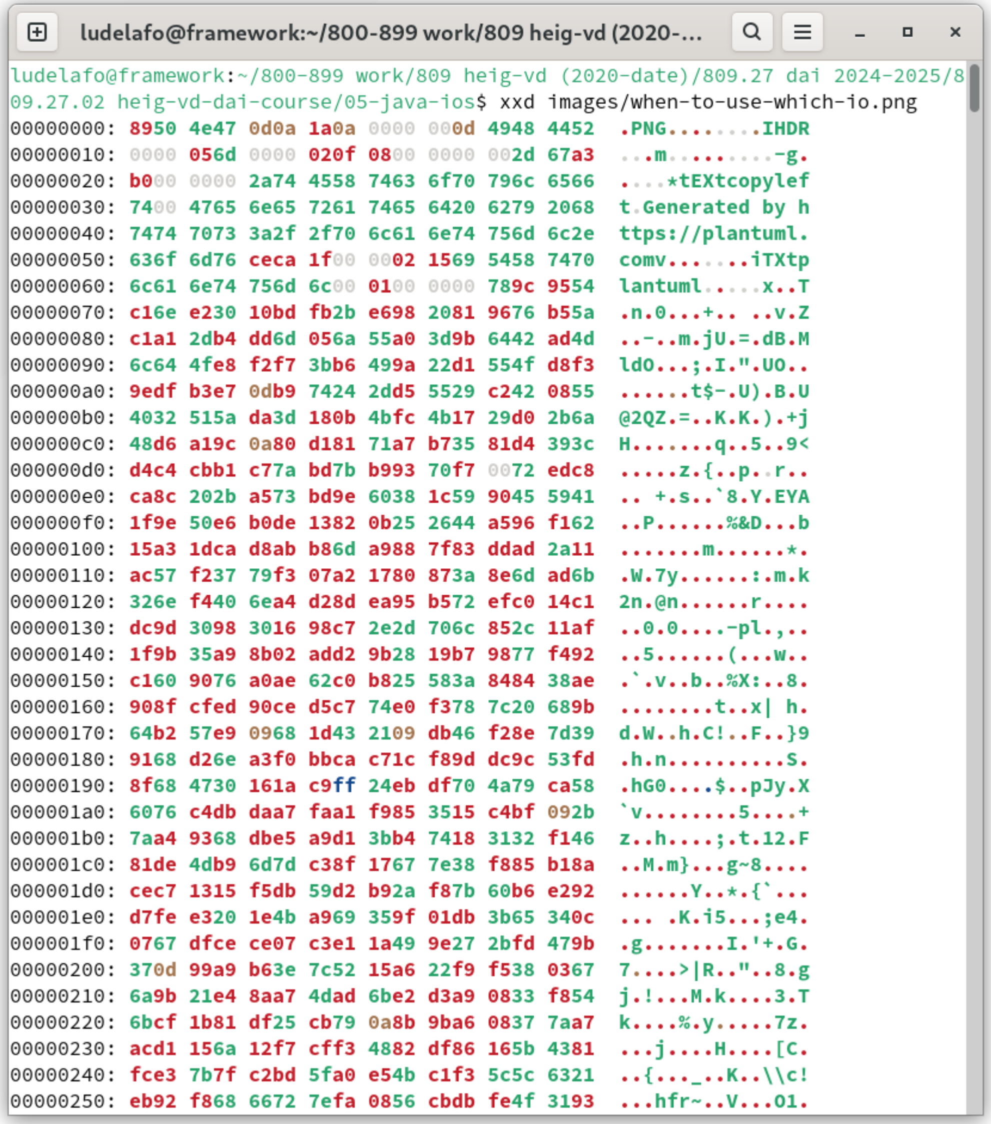
- Reading and writing byte by byte is not efficient: each `read()` or `write()` call results in a system call every time
- Buffers can be used to read write multiple bytes at once



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R.."..8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

Use a buffer to read multiple bytes at once:

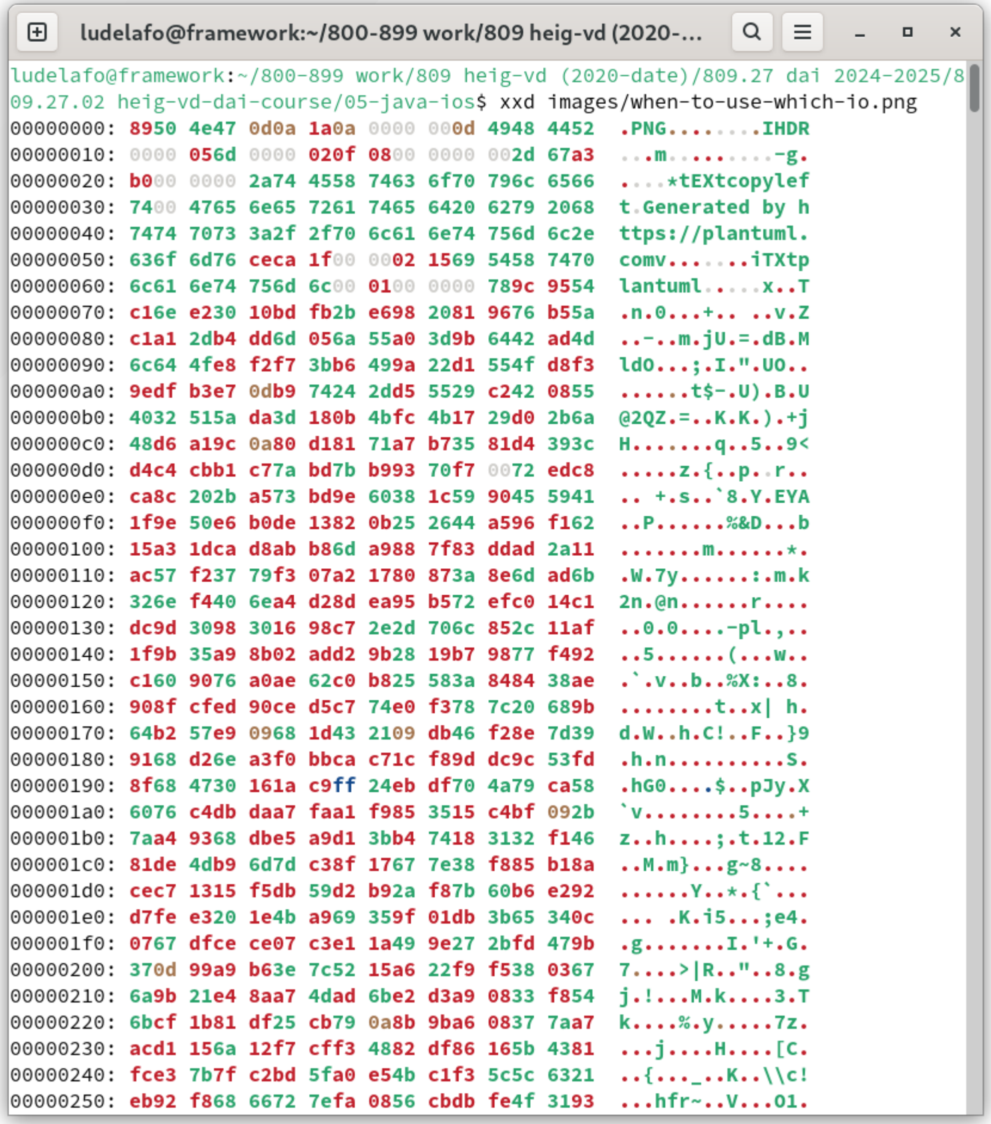
1. First time, a system call is made to read a block of data
2. Subsequent reads are done from the buffer
3. When the buffer is empty, a new block is read



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R..."..8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

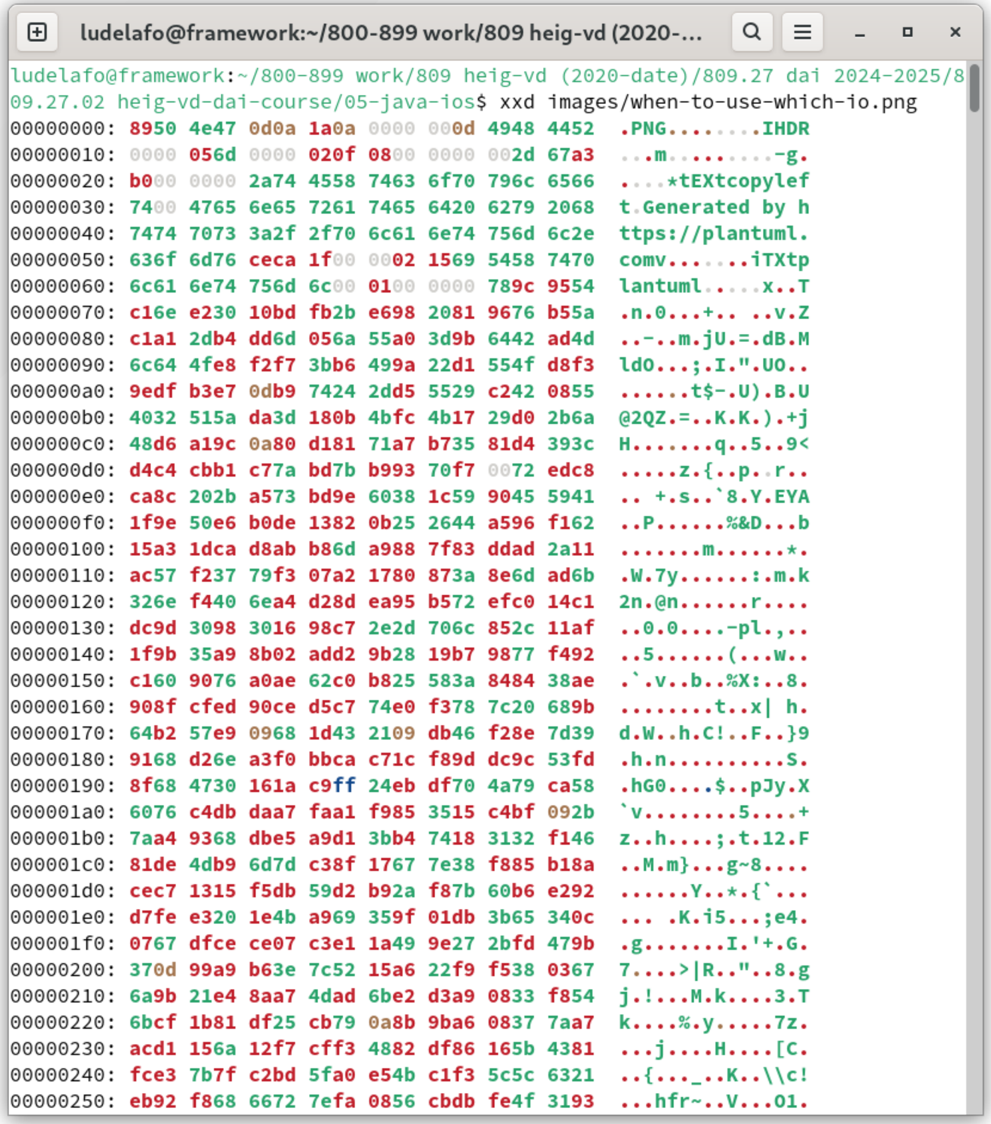
The same applies for writing:

1. First time, a buffer is created
2. Data is written to the buffer
3. When the buffer is full, a system call is made to write the block
4. The buffer is then emptied
5. Bytes can remain in the buffer
   - A flush might be needed to empty the buffer



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R..".8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

- `BufferedInputStream` and `BufferedOutputStream` classes are used to read/write binary data with buffers
- Let's have a look at the code examples `BinaryBufferReadFileExample` and `BinaryBufferWriteFileExample`



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
00000010: 0000 056d 0000 020f 0800 0000 002d 67a3  ...m.........-g.
00000020: b000 0000 2a74 4558 7463 6f70 796c 6566  ....*tEXtcopylef
00000030: 7400 4765 6e65 7261 7465 6420 6279 2068  t.Generated by h
00000040: 7474 7073 3a2f 2f70 6c61 6e74 756d 6c2e  ttps://plantuml.
00000050: 636f 6d76 ceca 1f00 0002 1569 5458 7470  comv.......iTXtp
00000060: 6c61 6e74 756d 6c00 0100 0000 789c 9554  lantuml.....x..T
00000070: c16e e230 10bd fb2b e698 2081 9676 b55a  .n.0...+.. ..v.Z
00000080: c1a1 2db4 dd6d 056a 55a0 3d9b 6442 ad4d  ..-..m.jU.=.dB.M
00000090: 6c64 4fe8 f2f7 3bb6 499a 22d1 554f d8f3  ldO...;.I.".UO..
000000a0: 9edf b3e7 0db9 7424 2dd5 5529 c242 0855  ......t$-.U).B.U
000000b0: 4032 515a da3d 180b 4bfc 4b17 29d0 2b6a  @2QZ.=..K.K.).+j
000000c0: 48d6 a19c 0a80 d181 71a7 b735 81d4 393c  H.......q..5..9<
000000d0: d4c4 cbb1 c77a bd7b b993 70f7 0072 edc8  .....z.{..p..r..
000000e0: ca8c 202b a573 bd9e 6038 1c59 9045 5941  .. +.s..`8.Y.EYA
000000f0: 1f9e 50e6 b0de 1382 0b25 2644 a596 f162  ..P......%&D...b
00000100: 15a3 1dca d8ab b86d a988 7f83 ddad 2a11  .......m......*.
00000110: ac57 f237 79f3 07a2 1780 873a 8e6d ad6b  .W.7y......:.m.k
00000120: 326e f440 6ea4 d28d ea95 b572 efc0 14c1  2n.@n......r....
00000130: dc9d 3098 3016 98c7 2e2d 706c 852c 11af  ..0.0....-pl.,..
00000140: 1f9b 35a9 8b02 add2 9b28 19b7 9877 f492  ..5......(...w..
00000150: c160 9076 a0ae 62c0 b825 583a 8484 38ae  .`.v..b..%X:..8.
00000160: 908f cfed 90ce d5c7 74e0 f378 7c20 689b  ........t..x| h.
00000170: 64b2 57e9 0968 1d43 2109 db46 f28e 7d39  d.W..h.C!..F..}9
00000180: 9168 d26e a3f0 bbca c71c f89d dc9c 53fd  .h.n..........S.
00000190: 8f68 4730 161a c9ff 24eb df70 4a79 ca58  .hG0....$..pJy.X
000001a0: 6076 c4db daa7 faa1 f985 3515 c4bf 092b  `v........5....+
000001b0: 7aa4 9368 dbe5 a9d1 3bb4 7418 3132 f146  z..h....;.t.12.F
000001c0: 81de 4db9 6d7d c38f 1767 7e38 f885 b18a  ..M.m}...g~8....
000001d0: cec7 1315 f5db 59d2 b92a f87b 60b6 e292  ......Y..*.{`...
000001e0: d7fe e320 1e4b a969 359f 01db 3b65 340c  ... .K.i5...;e4.
000001f0: 0767 dfce ce07 c3e1 1a49 9e27 2bfd 479b  .g.......I.'+.G.
00000200: 370d 99a9 b63e 7c52 15a6 22f9 f538 0367  7....>|R.."..8.g
00000210: 6a9b 21e4 8aa7 4dad 6be2 d3a9 0833 f854  j.!...M.k....3.T
00000220: 6bcf 1b81 df25 cb79 0a8b 9ba6 0837 7aa7  k....%.y.....7z.
00000230: acd1 156a 12f7 cff3 4882 df86 165b 4381  ...j....H....[C.
00000240: fce3 7b7f c2bd 5fa0 e54b c1f3 5c5c 6321  ..{..._..K..\\c!
00000250: eb92 f868 6672 7efa 0856 cbdb fe4f 3193  ...hfr~..V...O1.
```

```diff
   public static void main(String[] args) throws IOException {
     InputStream fis = new FileInputStream("binary-file.bin");
+    InputStream bis = new BufferedInputStream(fis);

     // -1 indicates the end of the file
     int b;
-    while ((b = fis.read()) != -1) {
+    while ((b = bis.read()) != -1) {
       System.out.print(b);
     }

-    fis.close();
+    // Closing the BufferedInputStream automatically closes the FileInputStream
+    bis.close();
   }
   public static void main(String[] args) throws IOException {
     OutputStream fos = new FileOutputStream("binary-file.bin");
+    OutputStream bos = new BufferedOutputStream(fos);

     for (int i = 0; i < 256; i++) {
-      fos.write(i);
+      bos.write(i);
     }

-    fos.close();
+    // Flush the buffer to write the remaining bytes
+    bos.flush();
+    bos.close();
   }
```

A quick note on little endian vs. big endian

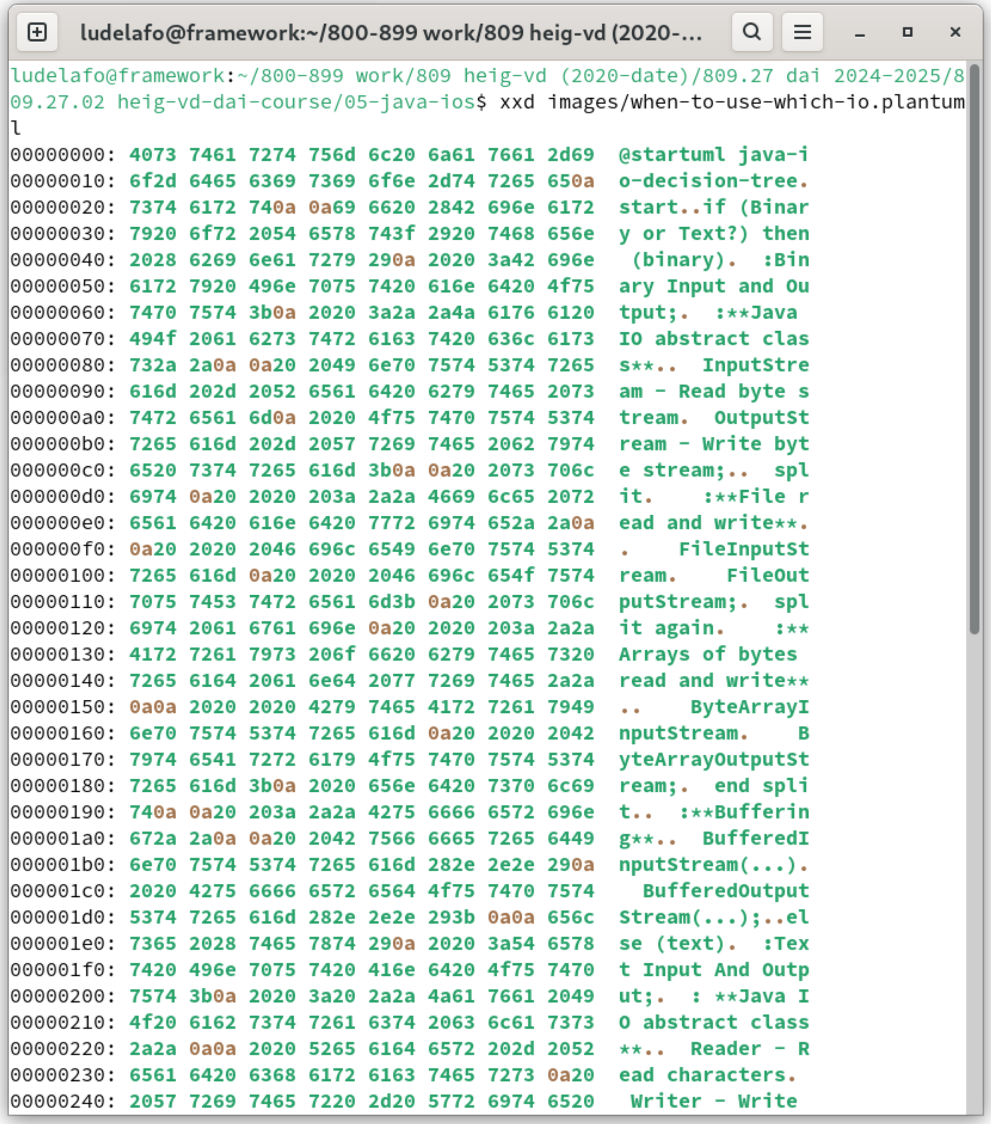
- Little endian: least significant byte first
- Big endian: most significant byte first
- Java uses big endian by default
- The class `ByteBuffer` can be used to convert between the two (not covered in this course)
- Example: `12345678`
  - Little endian: `0x78 0x56 0x34 0x12`
  - Big endian: `0x12 0x34 0x56 0x78`

Processing text data with the Java IO API

More details for this section in the course material. You can find other resources and alternatives as well.

Processing text data with the Java IO API

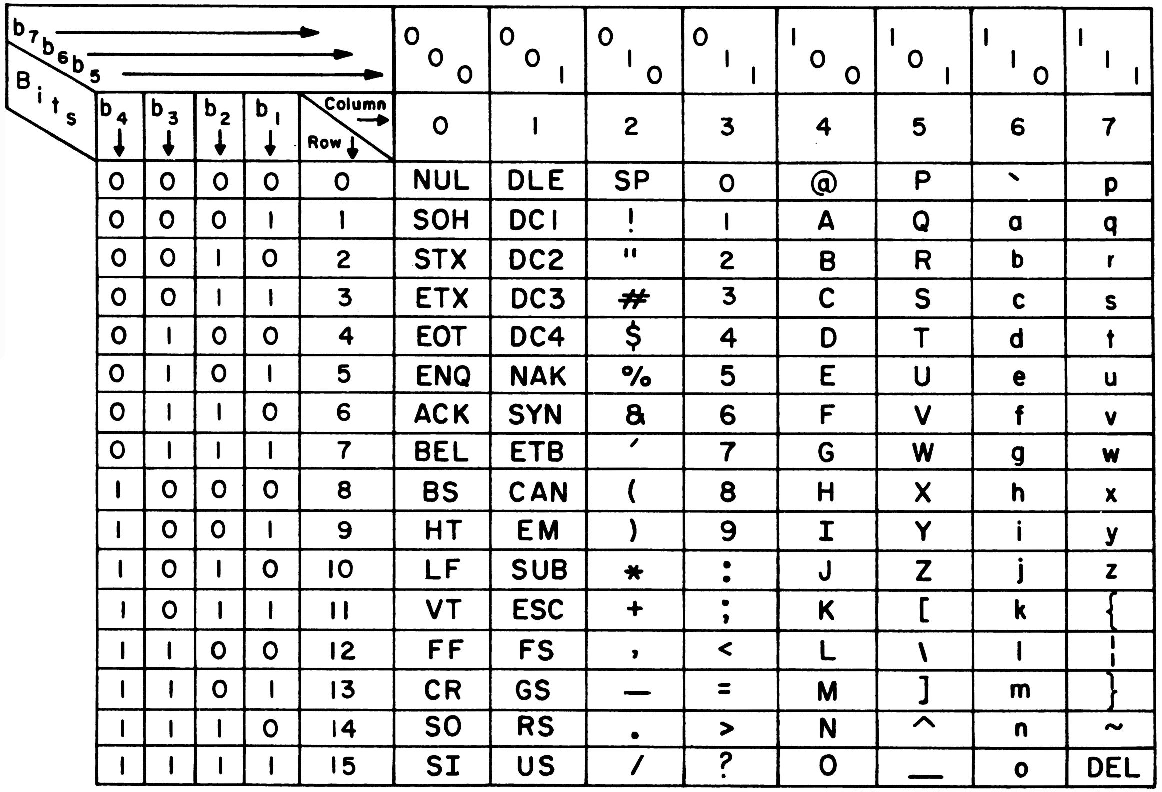
- Text data: interpretation of binary data
- Different character encodings
- Different end of line characters
- Different IO classes for text data



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.plantum
l
00000000: 4073 7461 7274 756d 6c20 6a61 7661 2d69  @startuml java-i
00000010: 6f2d 6465 6369 7369 6f6e 2d74 7265 650a  o-decision-tree.
00000020: 7374 6172 740a 0a69 6620 2842 696e 6172  start..if (Binar
00000030: 7920 6f72 2054 6578 743f 2920 7468 656e  y or Text?) then
00000040: 2028 6269 6e61 7279 290a 2020 3a42 696e   (binary).  :Bin
00000050: 6172 7920 496e 7075 7420 616e 6420 4f75  ary Input and Ou
00000060: 7470 7574 3b0a 2020 3a2a 2a4a 6176 6120  tput;.  :**Java 
00000070: 494f 2061 6273 7472 6163 7420 636c 6173  IO abstract clas
00000080: 732a 2a0a 0a20 2049 6e70 7574 5374 7265  s**..  InputStre
00000090: 616d 202d 2052 6561 6420 6279 7465 2073  am - Read byte s
000000a0: 7472 6561 6d0a 2020 4f75 7470 7574 5374  tream.  OutputSt
000000b0: 7265 616d 202d 2057 7269 7465 2062 7974  ream - Write byt
000000c0: 6520 7374 7265 616d 3b0a 0a20 2073 706c  e stream;..  spl
000000d0: 6974 0a20 2020 203a 2a2a 4669 6c65 2072  it.    :**File r
000000e0: 6561 6420 616e 6420 7772 6974 652a 2a0a  ead and write**.
000000f0: 0a20 2020 2046 696c 6549 6e70 7574 5374  .    FileInputSt
00000100: 7265 616d 0a20 2020 2046 696c 654f 7574  ream.    FileOut
00000110: 7075 7453 7472 6561 6d3b 0a20 2073 706c  putStream;.  spl
00000120: 6974 2061 6761 696e 0a20 2020 203a 2a2a  it again.    :**
00000130: 4172 7261 7973 206f 6620 6279 7465 7320  Arrays of bytes 
00000140: 7265 6164 2061 6e64 2077 7269 7465 2a2a  read and write**
00000150: 0a0a 2020 2020 4279 7465 4172 7261 7949  ..    ByteArrayI
00000160: 6e70 7574 5374 7265 616d 0a20 2020 2042  nputStream.    B
00000170: 7974 6541 7272 6179 4f75 7470 7574 5374  yteArrayOutputSt
00000180: 7265 616d 3b0a 2020 656e 6420 7370 6c69  ream;.  end spli
00000190: 740a 0a20 203a 2a2a 4275 6666 6572 696e  t..  :**Bufferin
000001a0: 672a 2a0a 0a20 2042 7566 6665 7265 6449  g**..  BufferedI
000001b0: 6e70 7574 5374 7265 616d 282e 2e2e 290a  nputStream(...).
000001c0: 2020 4275 6666 6572 6564 4f75 7470 7574    BufferedOutput
000001d0: 5374 7265 616d 282e 2e2e 293b 0a0a 656c  Stream(...);..el
000001e0: 7365 2028 7465 7874 290a 2020 3a54 6578  se (text).  :Tex
000001f0: 7420 496e 7075 7420 416e 6420 4f75 7470  t Input And Outp
00000200: 7574 3b0a 2020 3a20 2a2a 4a61 7661 2049  ut;.  : **Java I
00000210: 4f20 6162 7374 7261 6374 2063 6c61 7373  O abstract class
00000220: 2a2a 0a0a 2020 5265 6164 6572 202d 2052  **..  Reader - R
00000230: 6561 6420 6368 6172 6163 7465 7273 0a20  ead characters. 
00000240: 2057 7269 7465 7220 2d20 5772 6974 6520   Writer - Write 
```

Ancestor of character representations: ASCII

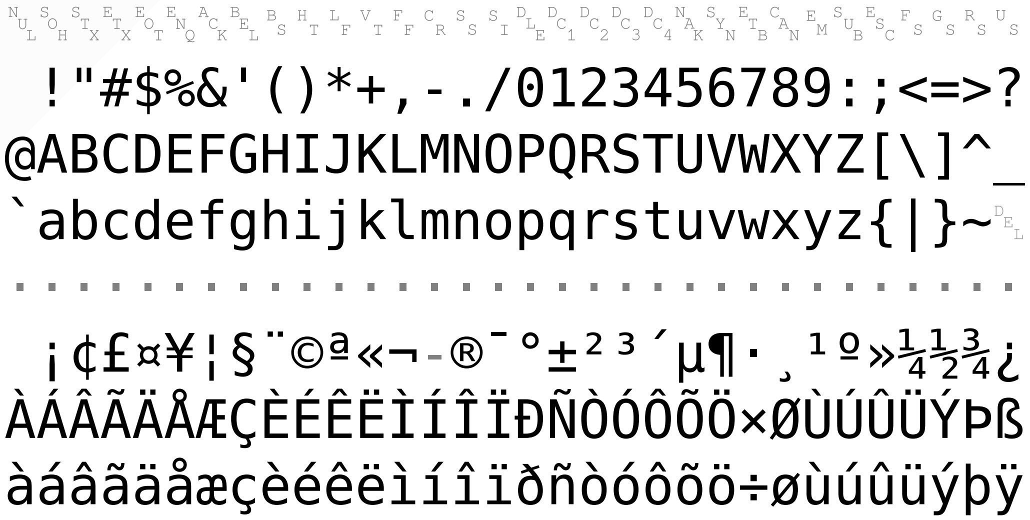
- ASCII: 128 binary values
- Mapping binary to characters
- Published in 1963 and meant for English



b7 b6 b5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits b4 ↓	b3 ↓	b2 ↓	b1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	´	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	¦
1	1	0	1	13	CR	GS	—	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Extended ASCII: codes pages

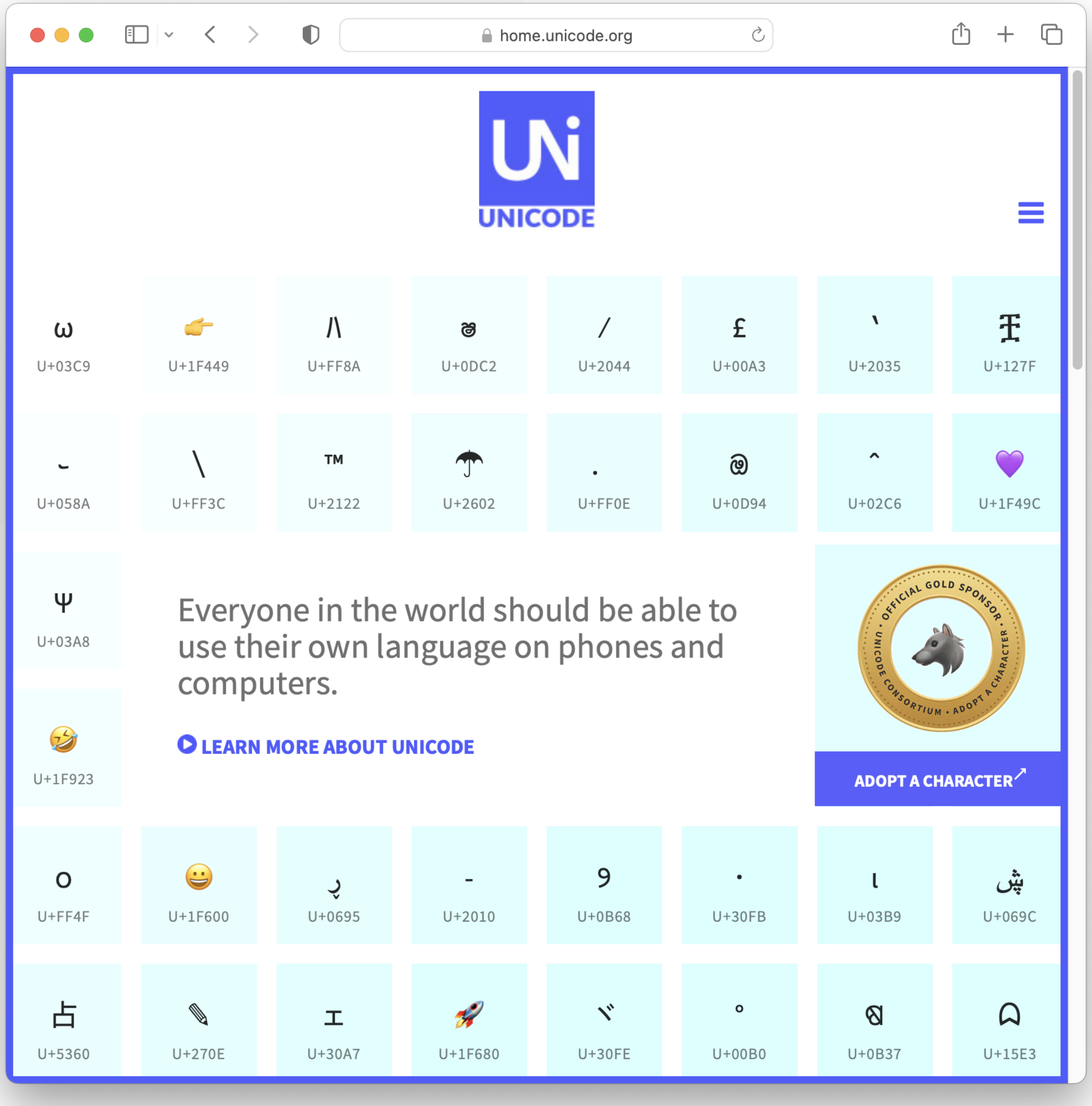
- Extended ASCII (code pages)
- Support for more characters using the remaining 128 values



```
NUL SOH STX ETX EOT ENQ ACK BEL BS HT LF VT FF CR SO SI DLE DC1 DC2 DC3 DC4 NAK SYN ETB CAN EM SUB ESC FS GS RS US
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~DEL
................................
 ¡¢£¤¥¦§¨©ª«¬-®¯°±²³´µ¶·¸¹º»¼½¾¿
ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
àáâãäåæçèéêëìíîïðñòóôõö÷øùúûüýþÿ
```

Unicode

- Unicode: solves ASCII limitations
- Standard to support all languages
- Different implementations:
    - UTF-8
    - UTF-16
    - UTF-32



ω U+03C9	👉 U+1F449	ﾊ U+FF8A	ෂ U+0DC2	⁄ U+2044	£ U+00A3	‵ U+2035	ቿ U+127F
֊ U+058A	＼ U+FF3C	™ U+2122	☂ U+2602	． U+FF0E	ඔ U+0D94	ˆ U+02C6	💜 U+1F49C
Ψ U+03A8							
🤣 U+1F923							
ｏ U+FF4F	😀 U+1F600	ڕ U+0695	‐ U+2010	୨ U+0B68	・ U+30FB	ι U+03B9	ڜ U+069C
占 U+5360	✎ U+270E	ョ U+30A7	🚀 U+1F680	ヾ U+30FE	° U+00B0	ଷ U+0B37	ᗣ U+15E3

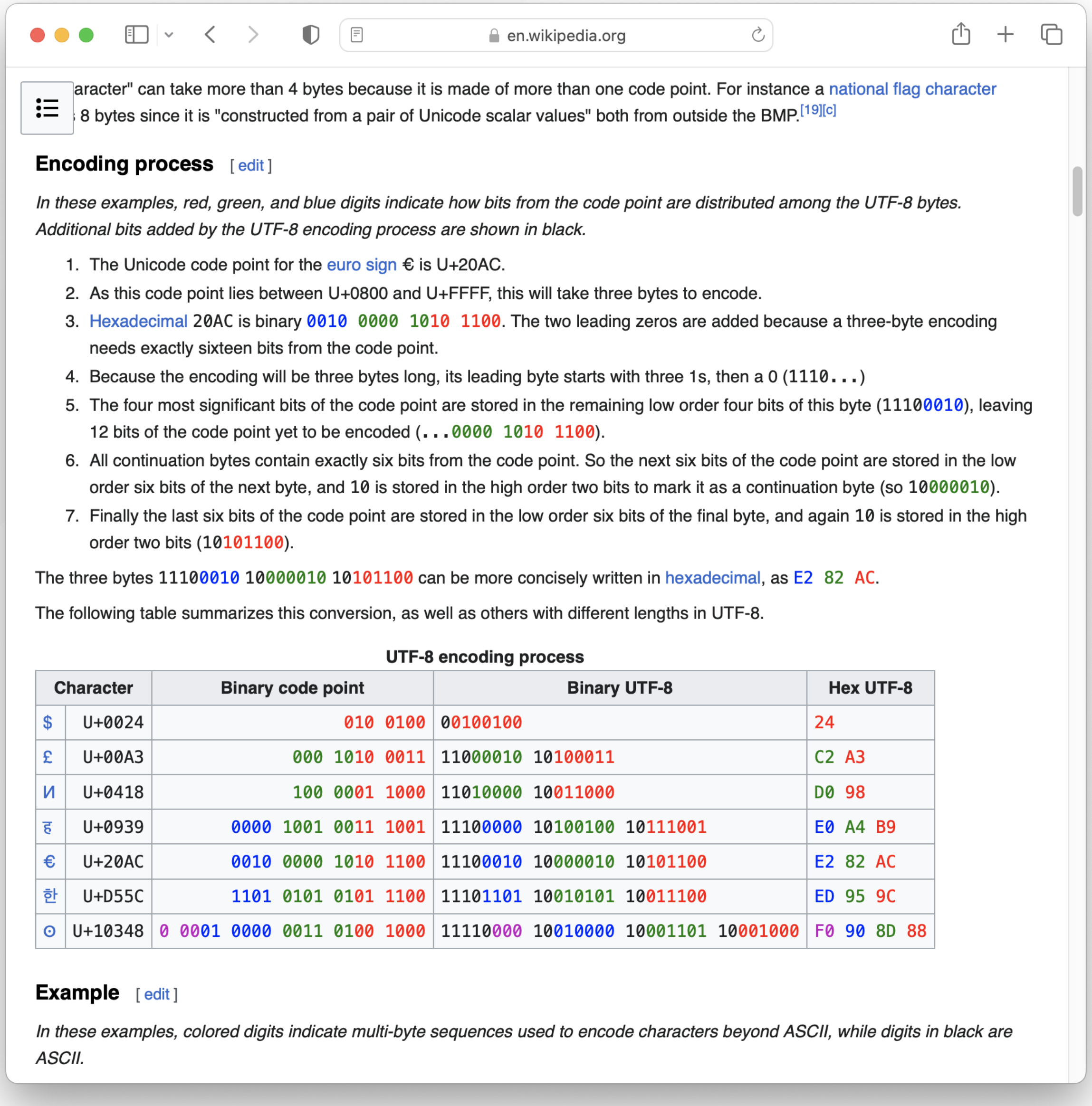
Everyone in the world should be able to use their own language on phones and computers.

LEARN MORE ABOUT UNICODE

OFFICIAL GOLD SPONSOR · ADOPT A CHARACTER · UNICODE CONSORTIUM ·
ADOPT A CHARACTER

UTF-8

- UTF-8: variable-length encoding
- Most common Unicode implementation
- ASCII compatible
- Quite the standard for web and software development



aracter" can take more than 4 bytes because it is made of more than one code point. For instance a national flag character 8 bytes since it is "constructed from a pair of Unicode scalar values" both from outside the BMP.[19][c]

Encoding process [edit]

In these examples, red, green, and blue digits indicate how bits from the code point are distributed among the UTF-8 bytes. Additional bits added by the UTF-8 encoding process are shown in black.

1. The Unicode code point for the euro sign € is U+20AC.
2. As this code point lies between U+0800 and U+FFFF, this will take three bytes to encode.
3. Hexadecimal `20AC` is binary `0010 0000 1010 1100`. The two leading zeros are added because a three-byte encoding needs exactly sixteen bits from the code point.
4. Because the encoding will be three bytes long, its leading byte starts with three 1s, then a 0 (`1110...`)
5. The four most significant bits of the code point are stored in the remaining low order four bits of this byte (`11100010`), leaving 12 bits of the code point yet to be encoded (`...0000 1010 1100`).
6. All continuation bytes contain exactly six bits from the code point. So the next six bits of the code point are stored in the low order six bits of the next byte, and `10` is stored in the high order two bits to mark it as a continuation byte (so `10000010`).
7. Finally the last six bits of the code point are stored in the low order six bits of the final byte, and again `10` is stored in the high order two bits (`10101100`).

The three bytes `11100010 10000010 10101100` can be more concisely written in hexadecimal, as `E2 82 AC`.

The following table summarizes this conversion, as well as others with different lengths in UTF-8.

UTF-8 encoding process

Character		Binary code point	Binary UTF-8	Hex UTF-8
$	U+0024	010 0100	00100100	24
£	U+00A3	000 1010 0011	11000010 10100011	C2 A3
И	U+0418	100 0001 1000	11010000 10011000	D0 98
ह	U+0939	0000 1001 0011 1001	11100000 10100100 10111001	E0 A4 B9
€	U+20AC	0010 0000 1010 1100	11100010 10000010 10101100	E2 82 AC
한	U+D55C	1101 0101 0101 1100	11101101 10010101 10011100	ED 95 9C
𐍈	U+10348	0 0001 0000 0011 0100 1000	11110000 10010000 10001101 10001000	F0 90 8D 88

Example [edit]

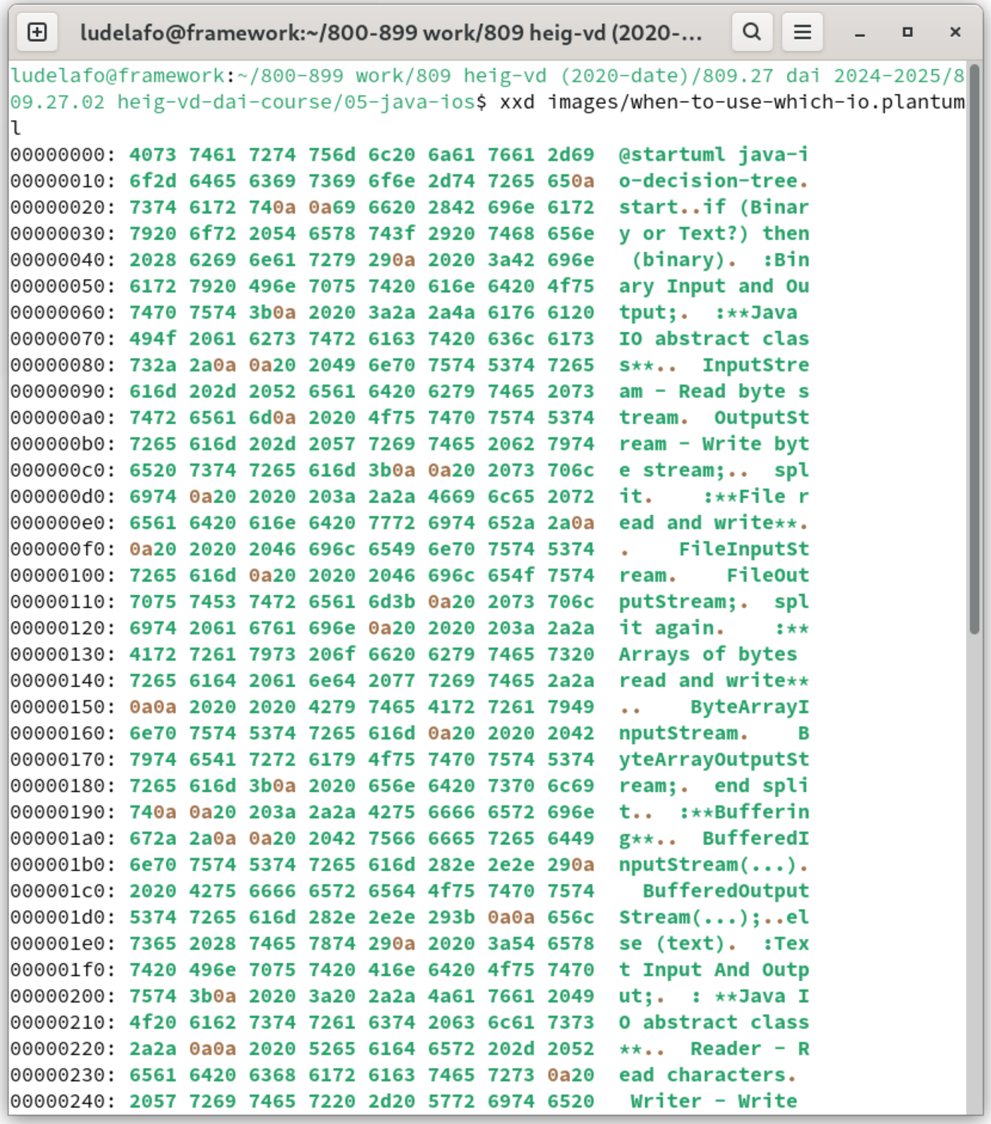
In these examples, colored digits indicate multi-byte sequences used to encode characters beyond ASCII, while digits in black are ASCII.

What happens if you ignore the character encoding?

- Not stored in the file itself
- Misinterpretation leads to issues
- Check, compile and run the `TextCharacterEncodingsExample` code example!

Reading and writing text data

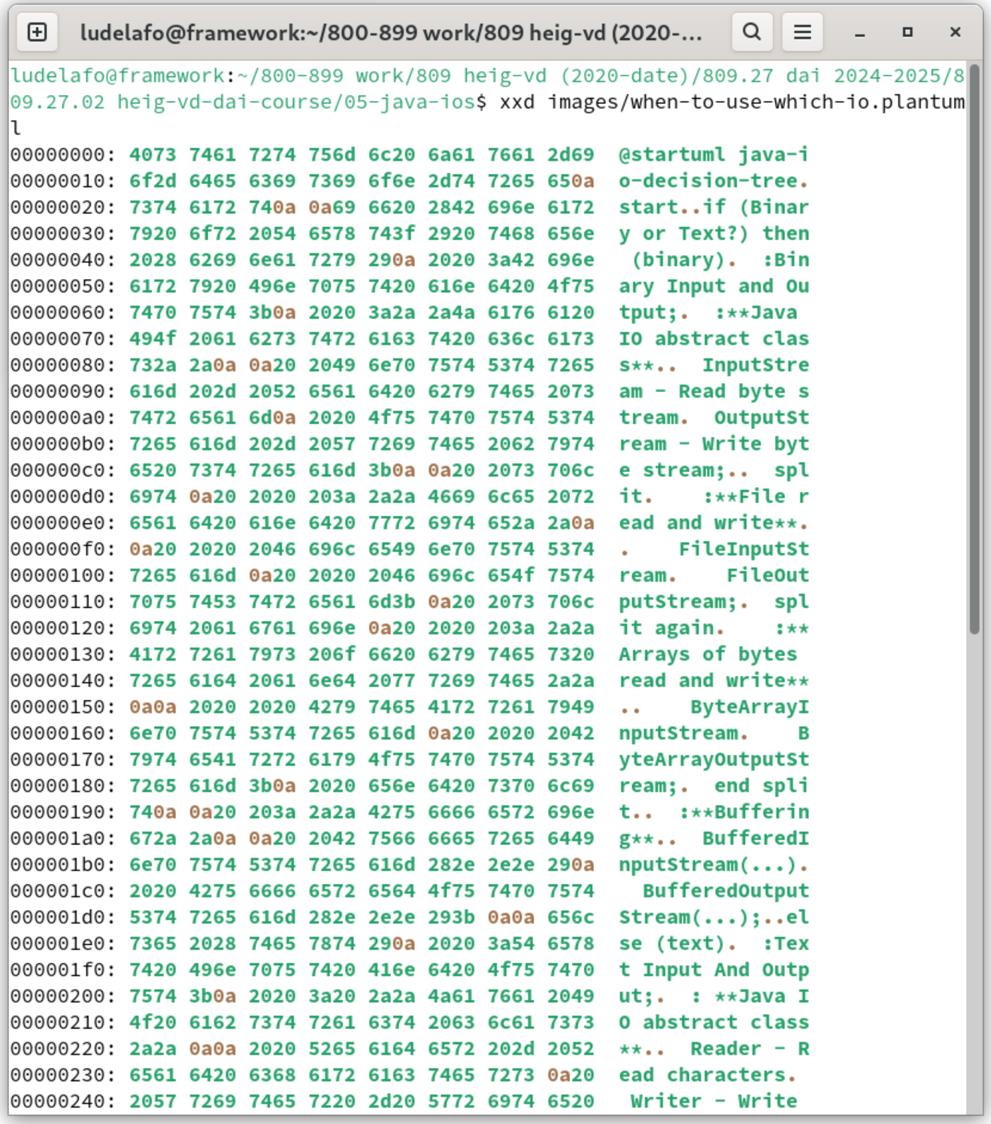
- `Reader` and `Writer` classes are used to read/write text data
- **Always specify the encoding!** If not set, it can be incompatible with other systems
- Let's have a look at the code example `TextReadAndWriteFileExample`



```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.plantum
l
00000000: 4073 7461 7274 756d 6c20 6a61 7661 2d69  @startuml java-i
00000010: 6f2d 6465 6369 7369 6f6e 2d74 7265 650a  o-decision-tree.
00000020: 7374 6172 740a 0a69 6620 2842 696e 6172  start..if (Binar
00000030: 7920 6f72 2054 6578 743f 2920 7468 656e  y or Text?) then
00000040: 2028 6269 6e61 7279 290a 2020 3a42 696e   (binary).  :Bin
00000050: 6172 7920 496e 7075 7420 616e 6420 4f75  ary Input and Ou
00000060: 7470 7574 3b0a 2020 3a2a 2a4a 6176 6120  tput;.  :**Java 
00000070: 494f 2061 6273 7472 6163 7420 636c 6173  IO abstract clas
00000080: 732a 2a0a 0a20 2049 6e70 7574 5374 7265  s**..  InputStre
00000090: 616d 202d 2052 6561 6420 6279 7465 2073  am - Read byte s
000000a0: 7472 6561 6d0a 2020 4f75 7470 7574 5374  tream.  OutputSt
000000b0: 7265 616d 202d 2057 7269 7465 2062 7974  ream - Write byt
000000c0: 6520 7374 7265 616d 3b0a 0a20 2073 706c  e stream;..  spl
000000d0: 6974 0a20 2020 203a 2a2a 4669 6c65 2072  it.    :**File r
000000e0: 6561 6420 616e 6420 7772 6974 652a 2a0a  ead and write**.
000000f0: 0a20 2020 2046 696c 6549 6e70 7574 5374  .    FileInputSt
00000100: 7265 616d 0a20 2020 2046 696c 654f 7574  ream.    FileOut
00000110: 7075 7453 7472 6561 6d3b 0a20 2073 706c  putStream;.  spl
00000120: 6974 2061 6761 696e 0a20 2020 203a 2a2a  it again.    :**
00000130: 4172 7261 7973 206f 6620 6279 7465 7320  Arrays of bytes 
00000140: 7265 6164 2061 6e64 2077 7269 7465 2a2a  read and write**
00000150: 0a0a 2020 2020 4279 7465 4172 7261 7949  ..    ByteArrayI
00000160: 6e70 7574 5374 7265 616d 0a20 2020 2042  nputStream.    B
00000170: 7974 6541 7272 6179 4f75 7470 7574 5374  yteArrayOutputSt
00000180: 7265 616d 3b0a 2020 656e 6420 7370 6c69  ream;.  end spli
00000190: 740a 0a20 203a 2a2a 4275 6666 6572 696e  t..  :**Bufferin
000001a0: 672a 2a0a 0a20 2042 7566 6665 7265 6449  g**..  BufferedI
000001b0: 6e70 7574 5374 7265 616d 282e 2e2e 290a  nputStream(...).
000001c0: 2020 4275 6666 6572 6564 4f75 7470 7574    BufferedOutput
000001d0: 5374 7265 616d 282e 2e2e 293b 0a0a 656c  Stream(...);..el
000001e0: 7365 2028 7465 7874 290a 2020 3a54 6578  se (text).  :Tex
000001f0: 7420 496e 7075 7420 416e 6420 4f75 7470  t Input And Outp
00000200: 7574 3b0a 2020 3a20 2a2a 4a61 7661 2049  ut;.  : **Java I
00000210: 4f20 6162 7374 7261 6374 2063 6c61 7373  O abstract class
00000220: 2a2a 0a0a 2020 5265 6164 6572 202d 2052  **..  Reader - R
00000230: 6561 6420 6368 6172 6163 7465 7273 0a20  ead characters. 
00000240: 2057 7269 7465 7220 2d20 5772 6974 6520   Writer - Write 
```

```java
class TextReadAndWriteFileExample {

  public static void main(String[] args) throws IOException {
    Reader reader = new FileReader("file.java", StandardCharsets.UTF_8);
    Writer writer = new FileWriter("file.txt", StandardCharsets.UTF_8);

    // -1 indicates the end of the file
    int c;
    while ((c = reader.read()) != -1) {
      writer.write(c);
    }

    writer.close();
    reader.close();
  }
}
```

- Just like with binary data, reading and writing text data byte by byte is not efficient
- `BufferedReader` and `BufferedWriter` classes are used to read/write text data with buffers
- Let's have a look at the code example `TextBufferReadAndWriteFileExample`

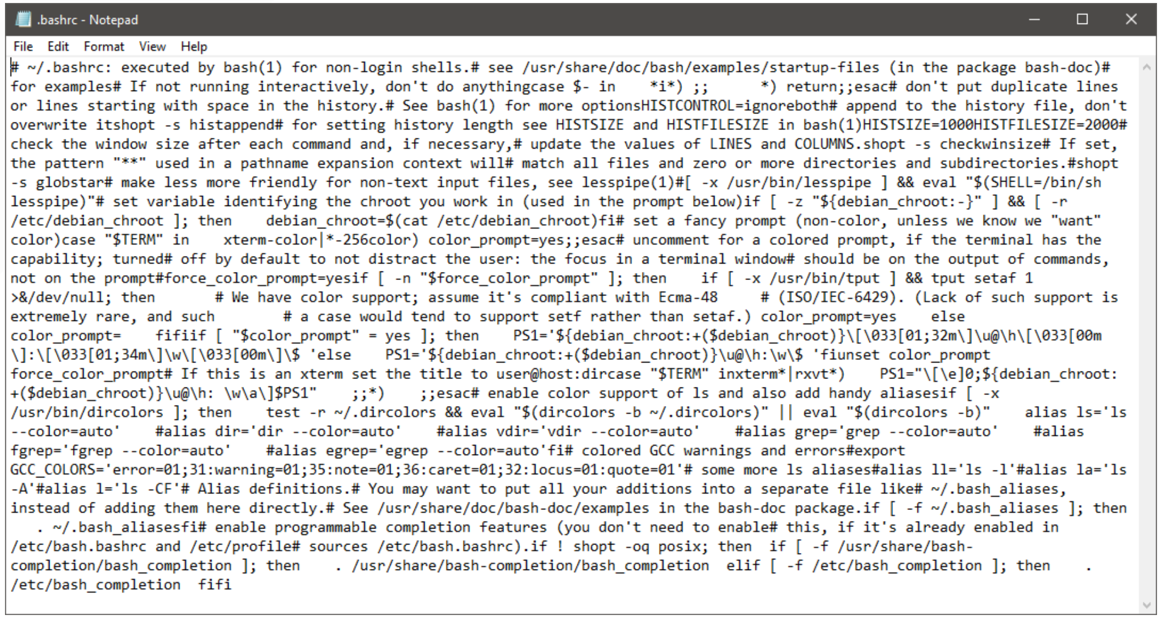


```
ludelafo@framework:~/800-899 work/809 heig-vd (2020-date)/809.27 dai 2024-2025/8
09.27.02 heig-vd-dai-course/05-java-ios$ xxd images/when-to-use-which-io.plantum
l
00000000: 4073 7461 7274 756d 6c20 6a61 7661 2d69  @startuml java-i
00000010: 6f2d 6465 6369 7369 6f6e 2d74 7265 650a  o-decision-tree.
00000020: 7374 6172 740a 0a69 6620 2842 696e 6172  start..if (Binar
00000030: 7920 6f72 2054 6578 743f 2920 7468 656e  y or Text?) then
00000040: 2028 6269 6e61 7279 290a 2020 3a42 696e   (binary).  :Bin
00000050: 6172 7920 496e 7075 7420 616e 6420 4f75  ary Input and Ou
00000060: 7470 7574 3b0a 2020 3a2a 2a4a 6176 6120  tput;.  :**Java 
00000070: 494f 2061 6273 7472 6163 7420 636c 6173  IO abstract clas
00000080: 732a 2a0a 0a20 2049 6e70 7574 5374 7265  s**..  InputStre
00000090: 616d 202d 2052 6561 6420 6279 7465 2073  am - Read byte s
000000a0: 7472 6561 6d0a 2020 4f75 7470 7574 5374  tream.  OutputSt
000000b0: 7265 616d 202d 2057 7269 7465 2062 7974  ream - Write byt
000000c0: 6520 7374 7265 616d 3b0a 0a20 2073 706c  e stream;..  spl
000000d0: 6974 0a20 2020 203a 2a2a 4669 6c65 2072  it.    :**File r
000000e0: 6561 6420 616e 6420 7772 6974 652a 2a0a  ead and write**.
000000f0: 0a20 2020 2046 696c 6549 6e70 7574 5374  .    FileInputSt
00000100: 7265 616d 0a20 2020 2046 696c 654f 7574  ream.    FileOut
00000110: 7075 7453 7472 6561 6d3b 0a20 2073 706c  putStream;.  spl
00000120: 6974 2061 6761 696e 0a20 2020 203a 2a2a  it again.    :**
00000130: 4172 7261 7973 206f 6620 6279 7465 7320  Arrays of bytes 
00000140: 7265 6164 2061 6e64 2077 7269 7465 2a2a  read and write**
00000150: 0a0a 2020 2020 4279 7465 4172 7261 7949  ..    ByteArrayI
00000160: 6e70 7574 5374 7265 616d 0a20 2020 2042  nputStream.    B
00000170: 7974 6541 7272 6179 4f75 7470 7574 5374  yteArrayOutputSt
00000180: 7265 616d 3b0a 2020 656e 6420 7370 6c69  ream;.  end spli
00000190: 740a 0a20 203a 2a2a 4275 6666 6572 696e  t..  :**Bufferin
000001a0: 672a 2a0a 0a20 2042 7566 6665 7265 6449  g**..  BufferedI
000001b0: 6e70 7574 5374 7265 616d 282e 2e2e 290a  nputStream(...).
000001c0: 2020 4275 6666 6572 6564 4f75 7470 7574    BufferedOutput
000001d0: 5374 7265 616d 282e 2e2e 293b 0a0a 656c  Stream(...);..el
000001e0: 7365 2028 7465 7874 290a 2020 3a54 6578  se (text).  :Tex
000001f0: 7420 496e 7075 7420 416e 6420 4f75 7470  t Input And Outp
00000200: 7574 3b0a 2020 3a20 2a2a 4a61 7661 2049  ut;.  : **Java I
00000210: 4f20 6162 7374 7261 6374 2063 6c61 7373  O abstract class
00000220: 2a2a 0a0a 2020 5265 6164 6572 202d 2052  **..  Reader - R
00000230: 6561 6420 6368 6172 6163 7465 7273 0a20  ead characters. 
00000240: 2057 7269 7465 7220 2d20 5772 6974 6520   Writer - Write 
```

```diff
   public static void main(String[] args) throws IOException {
     Reader reader = new FileReader("TextReadAndWriteFileExample.java", StandardCharsets.UTF_8);
+    BufferedReader br = new BufferedReader(reader);
+
     Writer writer = new FileWriter("TextReadAndWriteFileExample.txt", StandardCharsets.UTF_8);
+    BufferedWriter bw = new BufferedWriter(writer);

     // -1 indicates the end of the file
     int c;
-    while ((c = reader.read()) != -1) {
-      writer.write(c);
+    while ((c = br.read()) != -1) {
+      bw.write(c);
     }

-    writer.close();
-    reader.close();
+    // Flush the buffer to write the remaining bytes
+    bw.flush();
+    bw.close();
+    br.close();
   }
```

End of line characters

- Different end of line characters on different systems
  - Unix/Linux/macOS: `\n`, called *"Line feed"* (`LF`)
  - Windows: `\r\n`, called *"Carriage Return + Line feed"* (`CR` + `LF`)



```
# ~/.bashrc: executed by bash(1) for non-login shells.# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)#
for examples# If not running interactively, don't do anythingcase $- in    *i*) ;;      *) return;;esac# don't put duplicate lines
or lines starting with space in the history.# See bash(1) for more optionsHISTCONTROL=ignoreboth# append to the history file, don't
overwrite itshopt -s histappend# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)HISTSIZE=1000HISTFILESIZE=2000#
check the window size after each command and, if necessary,# update the values of LINES and COLUMNS.shopt -s checkwinsize# If set,
the pattern "**" used in a pathname expansion context will# match all files and zero or more directories and subdirectories.#shopt
-s globstar# make less more friendly for non-text input files, see lesspipe(1)#[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh
lesspipe)"# set variable identifying the chroot you work in (used in the prompt below)if [ -z "${debian_chroot:-}" ] && [ -r
/etc/debian_chroot ]; then    debian_chroot=$(cat /etc/debian_chroot)fi# set a fancy prompt (non-color, unless we know we "want"
color)case "$TERM" in    xterm-color|*-256color) color_prompt=yes;;esac# uncomment for a colored prompt, if the terminal has the
capability; turned# off by default to not distract the user: the focus in a terminal window# should be on the output of commands,
not on the prompt#force_color_prompt=yesif [ -n "$force_color_prompt" ]; then    if [ -x /usr/bin/tput ] && tput setaf 1
>&/dev/null; then       # We have color support; assume it's compliant with Ecma-48     # (ISO/IEC-6429). (Lack of such support is
extremely rare, and such        # a case would tend to support setf rather than setaf.) color_prompt=yes    else
color_prompt=    fifiif [ "$color_prompt" = yes ]; then    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m
\]:\[\033[01;34m\]\w\[\033[00m\]\$ 'else    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ 'fiunset color_prompt
force_color_prompt# If this is an xterm set the title to user@host:dircase "$TERM" inxterm*|rxvt*)    PS1="\[\e]0;${debian_chroot:
+($debian_chroot)}\u@\h: \w\a\]$PS1"    ;;*)    ;;esac# enable color support of ls and also add handy aliasesif [ -x
/usr/bin/dircolors ]; then    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"    alias ls='ls
--color=auto'    #alias dir='dir --color=auto'    #alias vdir='vdir --color=auto'    #alias grep='grep --color=auto'    #alias
fgrep='fgrep --color=auto'    #alias egrep='egrep --color=auto'fi# colored GCC warnings and errors#export
GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'# some more ls aliases#alias ll='ls -l'#alias la='ls
-A'#alias l='ls -CF'# Alias definitions.# You may want to put all your additions into a separate file like# ~/.bash_aliases,
instead of adding them here directly.# See /usr/share/doc/bash-doc/examples in the bash-doc package.if [ -f ~/.bash_aliases ]; then
   . ~/.bash_aliasesfi# enable programmable completion features (you don't need to enable# this, if it's already enabled in
/etc/bash.bashrc and /etc/profile# sources /etc/bash.bashrc).if ! shopt -oq posix; then  if [ -f /usr/share/bash-
completion/bash_completion ]; then    . /usr/share/bash-completion/bash_completion  elif [ -f /etc/bash_completion ]; then    .
/etc/bash_completion  fifi
```

```java
class TextEndOfLineCharactersExample {

  public static String END_OF_LINE = "\n";

  public static void main(String[] args) throws IOException {
    InputStream is = new FileInputStream("file.java");
    Reader reader = new InputStreamReader(is, StandardCharsets.UTF_8);
    BufferedReader br = new BufferedReader(reader);

    OutputStream os = new FileOutputStream("file.txt");
    Writer writer = new OutputStreamWriter(os, StandardCharsets.UTF_8);
    BufferedWriter bw = new BufferedWriter(writer);

    String line;
    while ((line = br.readLine()) != null) {
      // Careful: line does not contain end of line characters
      bw.write(line + END_OF_LINE);
    }

    bw.flush();
    br.close();
    is.close();
  }
}
```

Dealing with errors

- All kinds of errors can occur when reading/writing files
- Files must be properly opened and closed
- Ensure no files are corrupted
- Two main ways to handle exceptions:
  - `try-catch-finally` blocks
  - `try-with-resources` blocks

```java
// Bad example: try-catch without finally
public static void tryCatchWithoutFinallyExample() {
  try {
    Reader reader = new FileReader("missing.file");
    Writer writer = new FileWriter("missing.file");

    writer.write(reader.read());
  } catch (IOException e) {
    System.out.println("Exception: " + e);
  }
}
```

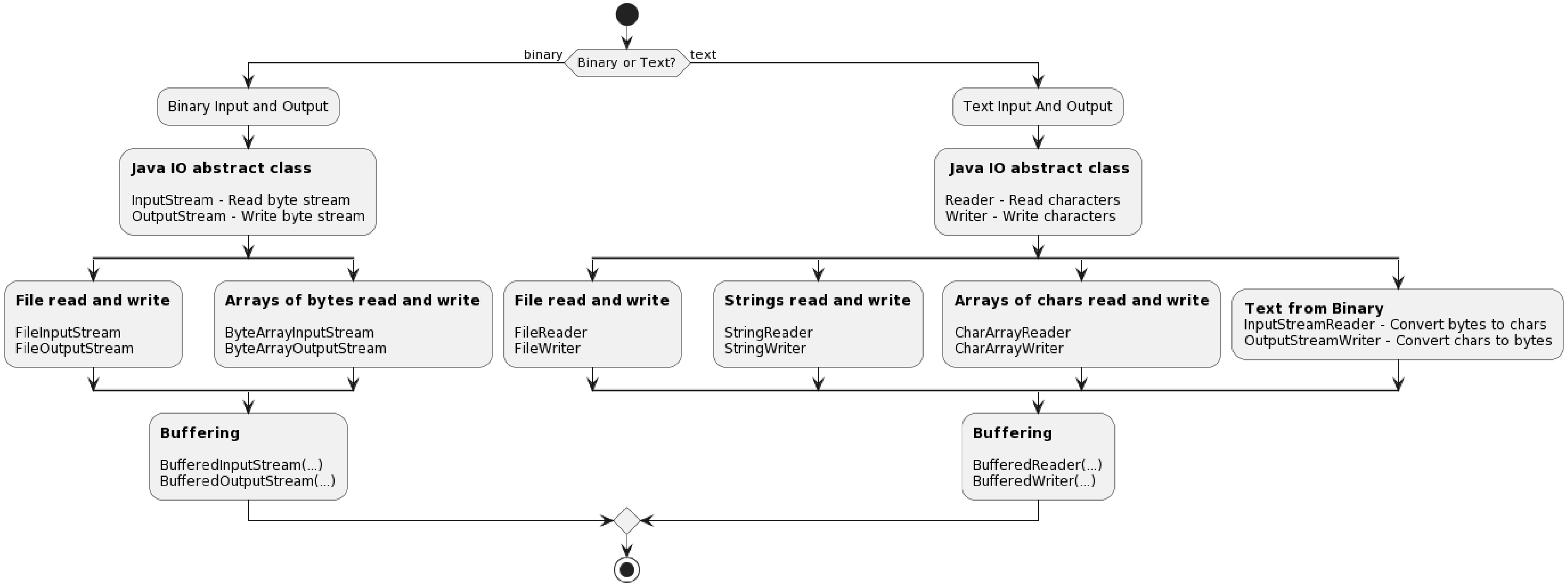
```java
// Better example: try-catch with finally
public static void tryCatchFinallyExample() {
  Reader reader = null;
  Writer writer = null;

  try {
    reader = new FileReader("missing.file");
    writer = new FileWriter("missing.file");

    writer.write(reader.read());
  } catch (IOException e) {
    System.out.println("Exception: " + e);
  } finally {
    if (writer != null) {
      try {
        writer.close();
      } catch (IOException e) {
        System.out.println("Exception in close writer: " + e);
      }
    }

    if (reader != null) {
      try {
        reader.close();
      } catch (IOException e) {
        System.out.println("Exception in close reader: " + e);
      }
    }
  }
}
```

```java
// Best example: try-with-resources
public static void tryWithResourcesExample() {
  try (Reader reader = new FileReader("missing.file");
      Writer writer = new FileWriter("missing.file")) {
    writer.write(reader.read());
  } catch (IOException e) {
    System.out.println("Exception: " + e);
  }
}
```

When to use which IO?

More details for this section in the course material. You can find other resources and alternatives as well.

When to use which IO?



Binary or Text?

binary

text

Binary Input and Output

Java IO abstract class

InputStream - Read byte stream
OutputStream - Write byte stream

File read and write

FileInputStream
FileOutputStream

Arrays of bytes read and write

ByteArrayInputStream
ByteArrayOutputStream

Buffering

BufferedInputStream(...)
BufferedOutputStream(...)

Text Input And Output

Java IO abstract class

Reader - Read characters
Writer - Write characters

File read and write

FileReader
FileWriter

Strings read and write

StringReader
StringWriter

Arrays of chars read and write

CharArrayReader
CharArrayWriter

Text from Binary
InputStreamReader - Convert bytes to chars
OutputStreamWriter - Convert chars to bytes

Buffering

BufferedReader(...)
BufferedWriter(...)

Common pitfalls

More details for this section in the course material. You can find other resources and alternatives as well.

Common pitfalls

- Not using buffers
- Not closing the streams
- Not handling exceptions properly
- Not specifying the character encoding
- Not specifying the end of line characters
- Do not use `PrintWriter` - it swallows exceptions
- Do not use `System.lineSeparator()` - it is platform dependent

Questions

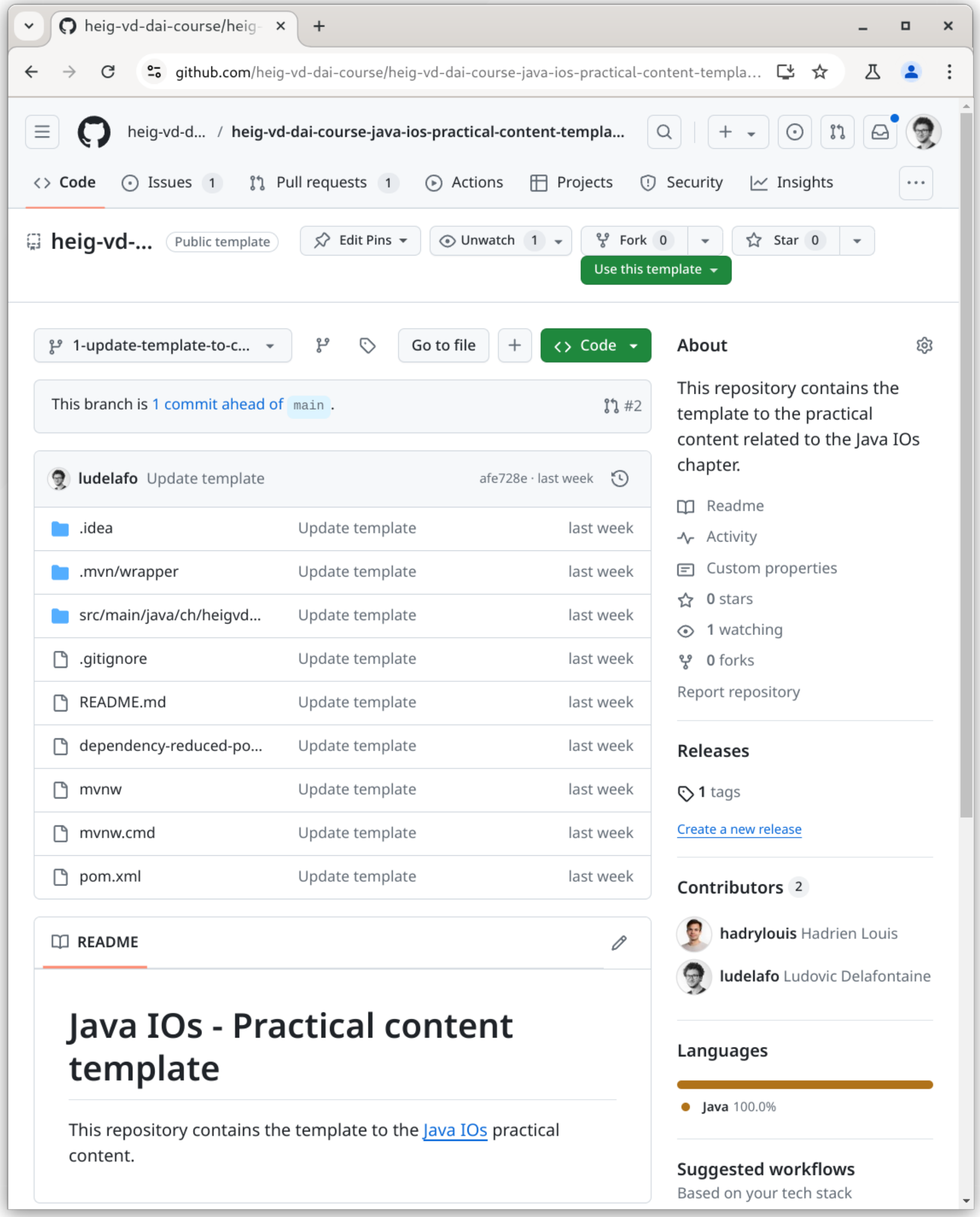
Do you have any questions?

Practical content

What will you do?

Benchmark the different types of streams you have learned:

- Assemble all the code examples to satisfy the use-cases
- Run some benchmarks to determine the best IOs for the given use-cases



This branch is 1 commit ahead of `main`. #2

ludelafo Update template afe728e · last week

.idea	Update template	last week
.mvn/wrapper	Update template	last week
src/main/java/ch/heigvd...	Update template	last week
.gitignore	Update template	last week
README.md	Update template	last week
dependency-reduced-po...	Update template	last week
mvnw	Update template	last week
mvnw.cmd	Update template	last week
pom.xml	Update template	last week

This repository contains the template to the practical content related to the Java IOs chapter.

Java IOs - Practical content template

This repository contains the template to the Java IOs practical content.

Find the practical content

You can find the practical content for this chapter on GitHub.

Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this chapter?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

➡️ GitHub Discussions

You can use reactions to express your opinion on a comment!

What will you do next?

In the next chapter, you will learn the following topics:

- Docker and Docker Compose: how to containerize your applications
  - What is an image?
  - What is a container?
  - How to try out new software without installing it?

Sources

- Main illustration by Martijn Baudoin on Unsplash
- Illustration by Aline de Nadai on Unsplash
- Illustration by T K on Unsplash
- Illustration by Anna Dudkova on Unsplash
- Illustration by Sigmund on Unsplash