

Java Programming assignment #1: Base64 encoding

due by March 23, 2019

Introduction

Coding is a way of representing information. A symbol is mapped on a value. If we have larger symbol set, then we can present more information with the same number of symbols. Bit (binary digit) is a kind of code. We can present 0 or 1 with a single symbol (digit). Decimal number system is also a kind of code. We can present ten different values with a single symbol (digit).

As we all know, ASCII code has a slightly larger symbol set because each symbol has 8 bits, so that it can present 256 different values with single symbol.

Similarly, we can think of hexa-decimal number system, which can present 16 different values with a single symbol (0~9, a~f), each of which requires 4 bits.

Likewise hexadecimal number system, we can think of slightly larger symbol set, such that each symbol can have 64 different values with a single digit. The symbol set S is as follows:

$$S = \{A \sim Z, a \sim z, 0 \sim 9, +, / \}.$$

We call this code as Base64 code. In Base64 code, each symbol requires 6 bits to present a value, similarly to 4 bits to present single hexadecimal value.

Base64 codes are widely used in text-based communication, particularly when text and binary data are mixed together. It is difficult for humans to read because it is different from ASCII code, but computers can comprehend it if it is properly handled.

The homework is to make a Java application that encodes/decodes ASCII text into/from Base64 codes.

To begin with, you can get input from the standard input or input file. We didn't cover how to deal with files, and we know how to use Scanner, so it is quite okay

to use Scanner. You can treat input either as ASCII text or as hexa-decimal numbers.

Then, you need to convert it into bytes array, so that you can put single data value in single array item.

While you're converting your input data in bytes array, you will need to check the alignment. The Base64 encoding requires each symbol has 6 bits aligned, and your data needs to be 3 bytes aligned because LCM (least common multiple) of 6 and 8 is 24 bits (= 3 bytes). To make 3 bytes aligned, you will need to allocate multiple of 3 bytes. If the data is shorter than multiple of 3, you will need to fill in data with padding symbol (=).

After you have aligned the data size, you will need to encode your ASCII or hex value into base64 codes. For every 3 bytes data, you have four base64 symbols (digits or characters), by reading data in six bits unit.

Then, finally prints out the symbol with the value on the mapped symbol. {A-Z, a-z, 0-9, +, /} symbols are mapped on a value {0~63}. You can use the value as index into the symbol set. For example, if your value for the first 6 bits is 0, then you will have the symbol 'A'. If your value for the second 6 bits is 26, then you will have the symbol 'Z'. If your value for the third 6 bits is 27, then you will have the symbol 'a'. If your value for the fourth 6 bits is 63, then you will have the symbol '/'. The final print out, Base64 encoded word, is 'AZa/', if you have data bytes `b[3] = {0x01, 0xa6, 0xff}`

That's Base64 encoding. The decoding is reverse procedure. You should have the same ASCII text or hexadecimal value with the given input.

SGFwcHkgaGFja2luZyB3aXRoIEphdmEgcHJvZ3JhbW1pbmch

Seehwan Yoo