

Practice 5

Deadline: 2 weeks from now. Should be checked onsite (during labs).

Task 1

Write a Java program to process a game map with Unicode terrain symbols. First, read the input map from a UTF-8 encoded file (`input.txt`). Then, process the symbols by the following rules:

- Water (~, U+007E): Convert all water tiles to ice (❄️, U+2744).
- Forest (🌲, U+1F332): Convert all forest tiles to barren land (□, U+2B1C).
- Mountain (⛰️, U+26F0): Mountains remain unchanged.

(You may search for unicode and directly copy-paste it from [here](#).)

Finally, write the processed map to a UTF-16 encoded file (`output.txt`).

Sample input:

```
~🌲🌲⛰️⛰️
~🌲~🌲⛰️⛰️
🌲~🌲~🌲~
```

Sample output (should be in UTF-16):

```
❄️❄️□□⛰️⛰️
❄️□❄️⛰️⛰️□
□❄️⛰️❄️□❄️
```

Hint

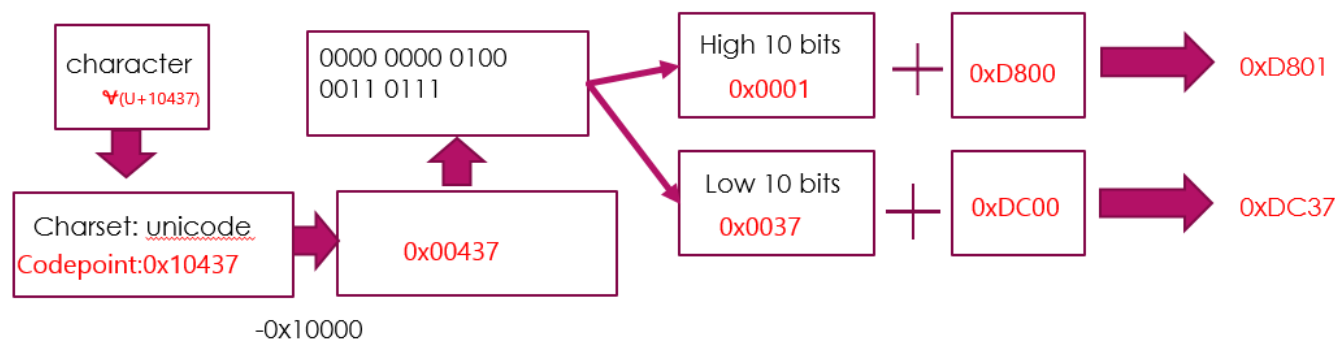
In Java, a `char` is a 16-bit data type that can represent characters in the Basic Multilingual Plane (BMP), which includes Unicode code points from U+0000 to U+FFFF. Characters outside this range, like 🌲 (U+1F332), require more than one `char` to be represented. These characters are stored as two `char` values, called a *surrogate pair*. To handle such characters, you may use special methods like `codePoints()` in the `String` class to properly process them, instead of treating them as individual `char` values.

Steps to Convert Code Point to Surrogate Pair:

1. **Check if the code point is in the range U+10000 to U+10FFFF.**
2. **Subtract 0x10000** from the code point: This reduces the code point range to 20 bits.
3. **Divide the result into high and low 10-bit values.**
4. **Add 0xD800 to the high 10 bits** to form the **high surrogate**.

5. Add 0xDC00 to the low 10 bits to form the **low surrogate**.

As shown in the figure below:



Task 2

Write a program that determines the type of a given file.

Note that we can't always rely on file extensions to determine its type, since file extensions may be missing, or incorrect (e.g., an attacker may change a malicious `game.exe` into a seemingly innocent `game.txt`).

Instead, we may check the file header to help determine its type. Below are file headers for three different file types. See [here](#) for more details on file format and magic numbers.

File Type	File Header (Hex)
png	89504e47
zip or jar	504b0304
class	cafebabe

Please write a `FileTypeParser.java` to determine the type of a given file. Executing the program

```
java FileTypeParser 1
```

will give the following output

```
Filename: 1
File Header(Hex): [ca, fe, ba, be]
File Type: class
```

You should run the code on the given input files 1,2,3.

Evaluation

The practice will be checked by teachers or SAs. What will be tested:

1. That you understand every line of your own code, not just copy from somewhere
2. That your program compiles correctly (javac)
3. Correctness of the program logic
4. That the result is obtained in a reasonable time

Late submissions after the deadline will incur a 20% penalty, meaning that you can only get 80% of this practice's score.