CS109 Lab 4

Ben Chen

SUSTech

May 24, 2023

Loop - do while

Do while will execute the code at hand without checking the condition.

- check the condition at the end of each repetition

```
do {
    inputNum = sc.nextInt();
    ...
} while(inputNum != guess);
```

Compared to *while*

- less elegant
- redundant

```
inputNum = sc.nextInt();

while(inputNum != guess) {
    ...
    inputNum = sc.nextInt();
}
```

Loop - for

For is used in countable repetition, written as,

```
for(<init>;<condition>;<step>) { // optional
    <code>
}
```

And writing like this is also allowed,

```
for(;;) { // DON'T forget semi-colon
    <code> // never stop
}
```

Loop - for

For example, output 9 times of the message

```
for(int i = 1; i < 10; i++) {
    System.out.println("This is for loop");
} // counting from 1 to 9
```

`i++` is executed after output, and then $i < 10$ will be checked;

Since when $i = 10$, condition is not satisfied, break the loop.

Break and Continue

Break and continue both terminate the present period of loop.

- **break** will end the loop

```
while(true) {
    ...
    if(i == 9) break;
}
```

- **continue** will execute the next period, skipping the rest.

```
for(int i = 1; i < 10; i++) {
    if(i == 5) continue;
    System.out.println("I am %d", i);
} // skip "I am 5"
```

Tips for OJ

To deal with multiple cases, you mustn't store the input in arrays.

```
int n = input.nextInt(); // number of cases
for(int i = 0; i < n; i++) {
    <initialize variables>
    <input>
    ... // solve each single case
    <output>
}
```

Switch Case

Another form of if-else, but notice that,

- **break** at the end of each Case
- Otherwise, other cases of code will be executed until **break**.
- **default** is not neccessary.

Why we need Command Line?

At restaurant, the experienced with command line is like:

- Customer: I'd like a bottle of coke, hambergur and chip.
- Waiter: OK, right away!

With Graphic Interface:

- Customer: Menu please.
- Waiter: Here's the menu.
  (minutes later)
- Customer: I want this(point to menu), this, and this.
- Waiter: OK, right away!

Command Line Interface

CLI is a software that renders your command.

- In macOS&Linux, it's Bash.
- In Windows, it's PowerShell or cmd.

When people talk about command line, people usually talk about Bash.

Windows Subsystem for Linux

To install WSL, please follow
`https://learn.microsoft.com/en-us/windows/wsl/install`

Path

Computer finds specific files by path.

- Absolute path: Originating from root folder, for example, */Users/chenben/Desktop/CS/baby.java*
- Relative path: Originating from current folder, for example, if I'm in */Users/chenben* , then, the path is *Desktop/CS/baby.java*
- `../` refers to parent folder, `/` refers to root folder, `~` refers to user folder.

Some commands about path,

- `pwd` checks the current path
- `ls` shows files and folders in the current path
- `cd` changes the current path

Execute program

There're three ways of executing program in command line,

- `./<program>` executes external program.
- `<command>` executes internal program. If you added the path of your program to PATH, you can directly use the name of it as command.
- `curl` executes remote program.

Flag

Flag is options for the command, present like -<flag>.

```
ls -al # -al is a flag asking to show full list
```

Flag might need arguments, a full pattern of command is like,

```
<command> arg1 arg2 -flag arg3 --Flag=arg4
# arg1 and arg2 are required by command
# arg3 is for -flag and arg4 can also be written like this.
```

Basic Command

You may need these commands,

```
mv <file> <path> # move file to path
cp <file> <path> # make a copy of file to path
mkdir <name> # create a folder
rm <file> # remove file
rm -rf <folder> # remove folder and its files
touch <file> # create a new file
cat <file> # show the content of file
vim <file> # modify the content of file
man <cmd> # show manual of command
```