

## 21.12.22 복습

### < 프로젝트 생성 방법 >

1. IntelliJ 실행
2. New project
3. No SDK라고 나오는 경우 Download JDK  
Version : 11  
Vendor : Zulu community  
Location: C:\Users\jdk\azul-11.0.13
4. 설치 완료 후 Next
5. 작업할 위치 선택
6. Finish

### <Git, Github>

Git : 분산형 버전 관리 시스템의 한 종류

버전 관리: 파일들을 복사, 백업, 저장 등을 통해 관리하는 것

Github : Git을 지원하는 웹 호스팅 서비스 시스템(클라우드)의 한 종류로 내 컴퓨터에 있는 것의 히스토리를 가져와서 깃허브 웹사이트에 올릴 수 있고 변경된 히스토리를 확인할 수 있다.

Repository: 코드나 문서를 비롯한 리소스를 저장하는 곳을 말하며, 프로젝트 단위로 만든다.

원격 저장소(Remote repository) : 깃허브같은 호스팅 서비스 서버에 올라가 있는 저장소

로컬 저장소(Local repository) : 개인 컴퓨터에 있는 저장소

Fork: 다른 사람의 원격 저장소(원본)를 그대로 복사해 내 계정의 원격 저장소(사본)로 만드는 것

Commit : 현재 변경된 작업 상태를 확정하여 저장소에 저장하는 작업

Merge : Branch의 내용을 현재 Branch로 가져와 합치는 작업

Pull Request: 원본 저장소에 코드를 올릴 권한이 없을 때, merge를 요청하는 것

base는 merge될 저장소, compare은 새롭게 만든 merge할 저장소

### < commit >

1. IntelliJ에서 소스 코드를 작성
2. 상단 Git - commit
3. [이름] 작업 내용 형식으로 commit 메시지 기록
4. commit
5. Git - push - 사본에서 업로드 여부 확인

### <Pull Request to Origin>

1. 사본으로 이동
2. Contribute - open pull request
3. create pull request

### <Fetch Origin to Copy>

1. 사본으로 이동
2. Fetch upstream
3. Fetch and merge
4. 사본에 원본 내용들 갱신되는 것 확인

### <Git 작업 시 주의할 점>

공동 저장소를 사용하기 때문에 다른 사람들의 자료를 지우게 되면 원본 저장소에 문제가 생긴다.

### <변수>

특정 데이터타입의 정보를 저장할 수 있는 메모리 공간 - C

특정 데이터타입의 정보를 저장할 수 있는 공간 - JAVA

데이터타입(자료형) 변수이름 = 3;

데이터타입이란?

정수형: int, 실수형: float, double

int는 4byte = 32bit = 42억9천만 개의 숫자 표현 가능

float은 4byte, double은 8 byte

byte 개념은?

1byte = 8bit

bit 개념은?

컴퓨터에서 가장 작은 저장 단위로 0 혹은 1을 저장할 수 있음

1byte = 8 bit

1bit는 0과1 2개의 숫자 저장 가능

2bit는 00, 01, 10, 11 4개의 숫자 저장 가능

-> n비트는  $2^n$ 개의 숫자를 표현할 수 있다.

데이터타입에 숫자 제한이 걸리는 이유는 결국 비트 시스템 때문이다.

뱅크솔루션은 42억9천만 이상의 숫자 처리 필요

-> BigInteger라는 매우 큰 숫자를 처리할 수 있는 별도의 데이터 타입도 존재

### <float과 double 차이점>

float은 4byte, double은 8byte

포함 관계 때문에 float에 double을 집어넣을 수는 있지만, double의 타입이 float으로 강제로 바뀐다

표기법: float 타입은 f를 붙이고, double은 붙인다.

```
float fnum1 = 3.3f;  
double dnum2 = 7.7;
```

성능은 float이 높고 정밀도는 double이 높다.

### <println과 printf 차이점, 엔터가 있고 없고 차이>

print가 들어가면 전부 출력

ln이 들어가면 자연로그ln이 아닌 자동 엔터(개행)을 의미

f가 들어있으면 포맷 형식을 지정할 수 있음

// 포맷 형식을 지정한다.

: 포맷 == 양식

특정 양식을 지정하고 그 양식에 적합한 출력을 생성해준다는 뜻

&d : 정수 출력

%lf : float, double 소수점 출력

%s : 문자열

//문자열은 반드시 쌍따옴표로 감싸줘야함

ex) "test hi hi hello "

\n은 개행 문자로 엔터와 같은 역할을 한다.

println "문자열" + num -> num이라는 숫자를 문자로 바꿔 합쳐서 출력

printf "%d", num -> num이라는 숫자가 정수형식으로 출력

print와 printf는 엔터가 없고, println만 엔터가 적용된다.

```
System.out.println("num1 = " + num1 + ", num2 = " +  
num2 + ", num3 = " + num3 + ", num4 = " + num4);  
  
System.out.printf(  
"num1 = %d, num2 = %d, num3 = %d, num4 = %d\n",  
num1, num2, num3, num4
```

println은 단순문자열을 출력할 때 편하고 printf는 출력할 내용에 여러 개의 변수가 섞여 있을 때 쓰기 편하다.

<Math.random()을 사용한 랜덤(난수) 만들기>

Math.random()

은 0.0~1.0 미만의 값 사이에서 난수를 발생시킨다.

-Math.random() 함수를 통해 임의의 주사위값(1~6) 만들기

```
int dice = (int)(Math.random() * 6 + 1);
```

(0.0~0.99xxx)의 범위에 6을 곱하면 (0.0~5.99xxx)의 범위로 바뀌게 되고

이 값은 정수형으로 변환이 되어 0~5 값으로 바뀐다.

이에 1을 더하면 1~6 사이의 난수값을 얻을 수 있다.

<사칙연산과 나머지 연산>

더하기(+), 빼기(-), 곱하기(\*), 나누기(/), 나머지(%) 연산이 가능하다.

+) 출력문에 변수1 % 변수2를 나타내기 위해

1. %에 해당하는 아스키 코드값 37을 지정

```
System.out.printf("%d %c %d = %d\n", num1, 37, num2, res);
```

2. %%형식

```
System.out.printf("%d %% %d = %d\n", num1, num2, res);
```

3. 문자열로 나타내기

```
System.out.printf("%d %s %d = %d\n", num1, "%", num2, res);
```

실제 영문자의 경우 아스키코드로 구성되며 이 아스키코드는 숫자와 1대1로 대응된다.

## <관계 연산자>

```
boolean res;
```

Boolean은 true와 false를 정의하는 데이터타입이다.

- 관계연산자

1. 앞의 내용과 뒤의 내용이 같은지 비교하는 ==

```
res = num1 == num2;
```

2. 앞의 내용과 뒤의 내용이 다른지 비교하는 !=

```
res = num1 != num2;
```

3. 앞의 내용이 뒤의 내용보다 작은지 비교하는 <

```
res = num1 < num2;
```

4. 앞의 내용이 뒤의 내용보다 작거나 같은지 비교하는 <=

```
res = num1 <= num2;
```

5. 앞의 내용이 뒤의 내용보다 큰지 비교하는 >

```
res = num1 > num2;
```

6. 앞의 내용이 뒤의 내용보다 크거나 같은지 비교하는 >=

```
res = num1 >= num2;
```

-> 결과값은 Boolean(true or false) 값

## <if문 >

```
if(조건){
```

조건이 만족될 경우 진행할 코드

```
}
```

else // 조건이 만족되지 않을 경우 진행되는 else, else가 없다면 그대로 끝난다.

예시) 짝수와 홀수 구분

```
if (rand % 2 == 0) {  
    System.out.println("나는 짝수: " + rand);  
} else {  
    System.out.println("나는 홀수: " + rand);  
}
```