백준, 프로그래머스 사이트

\* java가 동작하는 영역은 메소드영역, heap영역, stack영역

:java.exe.로 JVM이 시작되면 JVM은 운영체제에서 할당받은 영역들

-메소드 (Static/class) 영역

JVM이 시작할때 생성

로딩된 클래스 바이트 코드 내용을 분석후 저장 : 런타임상수풀, 필드/메소드 데이터, 메소드코드, 생성자 코드

-스택 stack영역 변수의값이 stack형식으로 저장되고 나오게된다

스레드 별 생성

스택이름, 힙영역의 주소값이 저장

메소드 호출할때마다 Frame을 스택에 추가(push)

메소드 종료하면 Frame 제거 (pop)

-힙 heap영역 : JVM이 시작할 때 생성 배열,객체 실질적인 값이 저장, 사용되지않는 객체는 Garbage Collector에 의해 자동제거

\*null: 참조타입 변수는 힙영역의 객체를 참조하지 않는다는뜻

null값을 가지고 있는 참조 타입변수를 사용하면 NullPointerException이 발생한다

ex1)

int[] intArray = null;

intArray[0] = 10; -----> int[] intArray를 null로 초기화하여 intArray[]인 참조할 객체가없어 NullPointerException가 뜬다 Exception이라뜨면 보통 코드부분오류

\*new연산자는 힙영역에 새로운 객체를 만듬

String name1 = new String(“1”);

String name2 = new String(“1”); //new연산자로인해 다른 힙영역의 신용권 String객체를 만든다 name1 == name2가 되지않는다

객체로생성하여 string값이 같은지 비교할려면

boolean result = 원본문자열.equals(비교문자열);을 사용하면 비교할수있다

String name1 = “신용권”

String name2 = “신용권” //이때는 같은 힙영역의 String객체를 가리킨다->

name1 == name2 가 성립된다

\*new연산자로 배열객체 생성시 향후에 이배열을 사용할것이라고 미리정의

int[] a= null;

a = new int[길이];

해당 길이만큼의 배열객체 a를 생성

\*main메소드의 (String[] args)의 매개변수값에 접근하고싶을 때

run의 configuration에 들어가 program arguments에 값을 입력하여 접근할수있음

-Integer.parseInt(문자열) 문자열을 int값으로 변환

\*배열 중복된 변수 선언을 줄일수있다

배열.length하면 배열의 길이가 나오게된다

\*System.arrayCopy(복사할배열명,시작인덱스,복사될배열,끝인덱스)

\*(변수형지정)Math.random,()\*n 0~n-1의 수중 랜덤값 하나지정(확률이 가각 일정한 랜덤값)

-확률값이 다른 랜덤값은 어떻게? https://skyfe.tistory.com/entry/%ED%99%95%EB%A5%A0%EC%9D%84-%EC%A0%81%EC%9A%A9%ED%95%9C-%EB%9E%9C%EB%8D%A4%EA%B0%92-%EC%84%A0%ED%83%9D%ED%95%98%EA%B8%B0

누적확률값을 랜덤으로 선택하게하면 쉽게 구현가능

|  |  |  |
| --- | --- | --- |
| 값 | 확률 | 누적확률값 |
| 4 | 5% | 5% |
| 3 | 15% | 20% |
| 2 | 15% | 35% |
| 1 | 30% | 65% |
| 0 | 35% | 100% |

double r = rand()/(double)RAND\_MAX;//{0.0-1.0}

double dr = r\* 100.0f//{0.0-100.0}

double p[] = { 5.0f, 15.0f, 15.0f, 30.0f, 35.0f}; // 4,3,2,1,0

double cumulative = 0.0f;

for(int=0; i<5; I++){

cumulative += p[i];

if(dr<= cumulative){

return 4-i;

}

}

-객체.charAt(i)메소드를 통해 객체 I번째 문자열을 가져올 수 있다

-system.exit(0); 시스템종료

-scanner.nextInt(); 입력받은 것을 scan