**Java 프로그래밍 3주차 과제**

**컴퓨터공학과 20244003 안예건**

Week1

1. Prime

Int 형식의 변수 num에 29를 저장했습니다. 소수 판별을 위해 boolean 형식의 flag를 생성하고 false를 저장했습니다. 2에서부터 num에 저장된 정수까지 나누고 만약 나머지가 0이라면 flag는 false이고 이때 소수임이 출력됩니다. Flag가 true라면 소수가 아님이 출력됩니다.

1. GuessNumber

우선 스캐너를 scanner라는 이름으로 초기화합니다. 이후 랜덤 정수 하나를 1부터 최대 숫자인 maxNumber, 즉 100 까지의 범위 중에서 생성하여 randomNumber에 저장하고 while 무한반복 내에서 스캐너의 nextInt()를 통하여 userNumber를 입력받습니다. 랜덤 숫자와 입력받은 숫자를 if else문으로 비교하여 결과를 출력합니다. 유저가 숫자를 맞추었다면 break를 사용하여 종료합니다.

Week2

1) Todo

우선 스캐너를 정의하고 어레이 리스트 스트링형 task와 불리안형 isDone을 생성했습니다. 무한반복 내에서 메뉴가 출력되고 userInput에 사용자가 선택한 메뉴를 받습니다.

userInput 처리 :

1번이 추가되었다면 task에 nextLine스캐너와 add()를 이용하여 할 일을 저장하고, 아직은 추가만 된 상태이므로 isDone은 false로 add()하여둡니다.

2번을 입력받으면 isEmpty()를 통해 리스트가 빈 상태인지 먼저 확인하고, 비었다면 비었다는 문자열을 출력합니다. 그렇지 않다면 우선 목록을 출력합니다.

목록 출력 0부터 task의 사이즈까지 반복하는 for 문 내에서 task 번호(출력시에는 i + 1), task 리스트 내의 i번째 항목, isDone 리스트 내의 i번째 항목은 물음표 연산자(if-else 간소화) 를 사용하여 true일시 완료 혹은 false일 때 미완료를 출력합니다. 실습 때 말씀하신 대로 완료시에는 []내에 완료 미완료를 출력하도록 했습니다.

( 즉 출력 시 “번호. Task이름 [완/미완료]” 형식).

이 for문을 실행하기 위해서는 몇 번째의 task를 완료할지 유저로부터 int하나를 받아와야 하므로, 스캐너의 nextInt를 사용하여 index에 -1을 처리하여 저장합니다. 이때 -1을 하는 이유는 인간과는 다르게 java에서는 리스트의 첫 번째 인덱스가 0(번째)이고 두 번째가 1이기 때문입니다. 하지만 숫자를 입력받고 난 이후 할 일을 입력도 하지 않았는데 자동으로 넘어가버리는 현상이 발생했습니다. 너무나 당황스러웠지만 교수님께서 공지사항에 적어주신 대로 sc.nextLine()을 추가하여 개행 문자를 제거하니 깔끔하게 해결되었습니다. 감사합니다.

이후에는 완료하기를 희망하는 task 번호가 저장된 index를 사용하여 task를 완료 처리합니다. 이를 위해 index가 0 이상이며 && task size 미만일 때 ( 이 조건을 추가한 이유는 task에 없는데도 잘못 처리하는 경우가 있을 수 있기 때문입니다.) && isDone의 index를 get()해줍니다. 오류가 발생하지 않았다면 잘 완료되었다고 출력하고, 오류가 발생했다면 ( ex : index 번호가 유저의 실수로 잘못 입력된 경우) 오류가 발생했다는 문자열을 출력합니다.

3번을 입력받으면 우선 tasks.isEmpty()를 사용하여 리스트가 빈 상태인지 검사하고, 아니라면 2번에서 목록을 출력한 것과 동일한 방식으로 출력합니다.

4번을 입력받으면 종료되었다는 문자열을 출력하고 break를 통해 while을 붕괴시킵니다. else로는 1-4사이 정수를 입력하지 않았을 때 출력됩니다. 오류 메시지를 출력합니다.

2) lotto

Random한 수를 생성하기 위해 자바 유틸리티에서 Random을 임포트합니다.

로또 번호를 저장할 배열 lotto는 총 6개의 수를 저장해야 하므로 new int[6]로 초기화했습니다. 이후 랜덤 객체를 만들고 for 문 내에서 i가 0~5까지 1씩 후위 증가되며 총 6번 반복할 때 randmInt(45) + 1 을 함으로 인하여 무작위로 생성되는 수는 1부터 45까지의 범위를 가집니다. 그리고 isDuplicate를 통해 중복 체크를 하기 위하여 우선 false로 설정했습니다. 이후 내부 for문에는 해당 loop 차례의 i의 범위까지 lotto내의 함수를 검사하고 만약 이미 있는 번호라면 isDuplicate를 true로 변경하고, break를 통해 루프를 붕괴시킵니다. 중복된 수가 없다면 ( not isDuplicate 인 경우), i번째 인덱스의 lotto 리스트에, 생성된 수가 저장됩니다.

로또 번호를 출력하기 위하여 for문 내에 j가 0에서 5까지 증가하며 lotto 내의 인덱스 번호에 접근하고 이를 출력합니다.

3) calculator

우선 스캐너를 정의하고, 연산자를 입력받습니다. Char op 에 next.charAt(0)를 통해 스트링에 있는 첫 번째 문자를 리턴받아 저장합니다. 다음으로는 피연산자를 두 개 입력받는데 이때는 nextDouble()을 두 번 사용하여 double형의 num1과 num2를 저장합니다. If else문에서 op가 더하기일 때, 빼기일 때, 곱하기일 때 모두 문자열 합치기를 이용하여 결과를 출력하고, 계산 결과 부분에서는 피연산자와 연산자를 사용하여 계산식을 적어줍니다. 주의해야 하는 부분으로 나누기 오퍼레이션을 할 때 나누는 수, 즉 num2가 0이 되면 안 되므로 num2의 값을 비교하여 만약 이가 0이라면 오류 구문을 출력하고, 오류가 발생하지 않았다면 다른 사칙연산과 동일하게 문자열 합치기를 통하여 피연산자와 연산자 그리고 계산결과를 출력합니다.

5) drawing

우선 스캐너를 임포트하고 무한반복문을 실행하여 메뉴를 출력합니다. 사용자가 원하는 메뉴를 입력하면 scanner의 nextInt()가 입력된 정수를 userInput에 저장합니다.

userInput 처리:

0이 입력되면 break를 통해 while문을 붕괴시키고 종료합니다.

1이 입력되면 첫 번째와 두 번째 점의 좌표를 입력받고 이는 각각 (x1, y1), (x2, y2)입니다. 이는 모두 nextInt()를 사용하며 입력될 때 공백으로 나뉘어 정수가 각각 저장됩니다.

drawLine 함수에 첫 번째 점과 두 번째 점을 인수로 전달합니다.

2가 입력되면 1이 입력될 때와 마찬가지로 두 점을 입력받고 drawSquare 함수에 인수를 각각 전달합니다.

1 2 혹은 0이 아닌 입력을 받으면 else문이 실행되고 오류 메시지를 출력합니다.

drawLine 직선 그리기 함수:

메뉴에서 1을 입력하고 점을 입력받은 이후 인수들이 이 함수에 int형으로 전달됩니다.

5 by 5 필드가 필요하므로 field라는 이름을 가진 5\*5 2차원 배열을 초기화합니다.

아래는 수업 시간에 교수님께서 적어주신대로 선형 보간법을 사용하였습니다.

Step은 x, y축 중에 더 멀리 있는 첫번쨰 두번째 점이 축 길이라는 점을 응용하여

Math의 absolute value, 즉 Math.abs에서 dx에는 x좌표의 차를 절댓값으로 나타내어 저장하고, dy에는 y좌표의 차를 절댓값으로 나타내어 저장했습니다.

step별로 값을 채울 좌표를 찾기 위하여 보간법 공식을 사용하였습니다.

내부 if문에서는 위의 x와 y좌표를 필드를 벗어날 만큼 크거나 작게 저장되어 오류가 포함된 채 그리지 않도록 x가 0 이상 5 미만이면서 y 또한 0 이상 5미만인 경우에 field(행(y),열(x))에 1로 저장하였습니다. 이 field는 다시 printField에 인수로 제공됩니다.

drawSquare 정사각형 그리기 함수

메뉴에서 2를 입력하고 점을 입력받은 후에는 drawLine과 마찬가지로 인수들이 이 함수에 int형으로 전달되고 이후 field를 5\*5 2차원 배열로 초기화합니다.

외곽 for문에서는 i가 x1에서 x2까지 증가하고, 내부 for문에서는 j가 y1에서 y2까지 증가합니다. 위에서와 마찬가지로 필드 범위를 벗어나지 않는 경우에 field ( 행(j), 열(i) ) 에 차례로 1이 저장되고, printField에 인수로 제공됩니다.

printField 필드 출력 함수

2차원 배열 형식의 field가 인수로 제공된 이후 외곽 for문에서는 0에서 4까지 i가 1씩 증가하며 루프를 돌고 내부 for문은 0에서 4까지 j가 1씩 증가하며 루프를 생성합니다. 루프를 돌 때 특정 (i, j) 위치의 field에 1이 저장되어 있을 때 그 자리에 별표를 출력합니다. 1이 저장되어 있지 않은 경우에는 밑줄을 출력합니다.

한 줄이 출력되었다면 줄을 바꿔주고 루프가 끝날 때까지 반복합니다.