[ JAVA 프로그래밍 ]

[ 제어문 사용하기 ]

담당 교수 : 지정희 교수님

학 번 : 202311264

이 름 : 김민홍

제출일 : 2023년 09월 18일

|  |  |
| --- | --- |
| 1 | **문제정의 및 분석** |

메인 메뉴 운영하기

이 프로그램은 4번 종료하기를 누를 때 까지 계속 수행되어야 하고 선택한 메뉴가 끝났을 때도 다시 메뉴를 선택할 수 있어야 한다. 또한 1~4이외의 숫자 값이 들어오더라도 잘못된 입력이라고 출력되고 다시 선택할 수 있어야한다.

|  |  |
| --- | --- |
| 2 | **주요 소스코드 설명** |

while(true){  
 int input = *selectMenu*();  
 switch (input){  
 case 1->*cuponAndChoco*();  
 case 2-> *puzzle*();  
 case 3->*gugudan*();  
 case 4-> *sysExit*();  
 default -> System.*out*.println("잘못된 입력");  
 }  
}

switch문을 통해 사용자가 입력한 번호에 따라 각 메뉴에 따른 함수를 실행하도록 하였고 끝나더라도 while문을 통해 다시 메뉴가 나오도록 해주었다.

public static int selectMenu(){  
 System.*out*.print("1)쿠폰으로 초콜랏 구매하기 ");  
 System.*out*.print("2)문자 산수 퍼즐 ");  
 System.*out*.print("3)구구단 출력하기 ");  
 System.*out*.print("4)종료\n");  
 System.*out*.print("메뉴를 선택하세요 :");  
 int menu = *scanner*.nextInt();  
 return menu;  
}

메뉴 함수를 하나 만들어 사용자가 번호를 입력했을 때 값을 메인함수로 리턴할 수 있도록 해주었다.

|  |  |
| --- | --- |
| 3 | **실행결과** |

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

selectMenu함수에서 메뉴 내용을 출력해주고 밑에서 그 번호를 입력하게 구현함.

|  |  |
| --- | --- |
| 4 | **느낀점 및 토의사항** |

메뉴등을 만들 때는 if보다는 switch가 훨씬 편하다는 것을 느꼈고, 복잡한 코드가 메인에 있는 것 보다는 메소드를 통해 입력한 값에 따라서 메뉴를 실행할 수 있도록 구현한 것이 훨씬 코드에 대한 가독성이 좋다고 느꼈다.

|  |  |
| --- | --- |
| 5 | **과제 수행 완료 체크 리스트** |

<체크리스트>

1. 메뉴가 모두 출력되었는가? **네**

2. 1~4 이외의 값이 입력된 경우 메인 메뉴가 다시 동작되는가? **네**

3. 선택한 메뉴를 수행 완료한 후 다시 메인 메뉴가 동작되는가? **네**

4. 종료 메뉴를 선택해서 프로그램을 종료하기 전까지 반복 수행되는가? **네**

5. 종료 메뉴를 선택하면 프로그램이 종료되는가? **네**

6. 보고서에 문제 1 과 관련된 문제정의 및 분석, 해결방안과 실행결과를 충분히 설명하였는가 **네**

|  |  |
| --- | --- |
| 1 | **문제정의 및 분석** |

메뉴 1)의 쿠폰으로 초콜릿 구매하기

사용자가 돈을 입력하면 1원당 초콜릿 하나를 얻을 수 있고 모든 초콜릿에는 쿠폰이 들어있다. 7개의 쿠폰을 통해 초콜릿 바를 하나 돌려 받을 수 있다. 이때 모든 돈과 쿠폰을 사용하여 얻을 수 있는 초콜릿 바와 남은 쿠폰개수를 구하는 문제이다. 남은 쿠폰이 7미만이 될 때까지 while문을 통해 처리하는 것이 관건일 것 같다.

|  |  |
| --- | --- |
| 2 | **주요 소스코드 설명** |

int cupon;  
int choco;  
System.*out*.println("문제1. 쿠폰으로 초콜릿 구매하기");  
System.*out*.print("돈을 넣어 주세요 :");  
  
int money = *scanner*.nextInt();  
choco = money;  
cupon = choco;  
while(cupon>=7){  
 int addTmp = 0;  
 addTmp = cupon/7;  
 cupon %= 7;  
 cupon += addTmp;  
 choco += addTmp;  
}  
  
System.*out*.println("-".repeat(20));  
System.*out*.println("초콜릿 "+choco+"개, 쿠폰 "+cupon+"개");  
System.*out*.println("-".repeat(20));

우선 사용자로부터 돈을 입력받고 그 받은 돈을 통해 처음 초콜릿의 개수와 쿠폰의 개수를 정해준다. while문을 통해 쿠폰의 개수가 7개 미만이 될 때까지 쿠폰을 쓰도록 하고 쿠폰을 통해 산 초콜릿의 개수를 addTmp이라는 변수에 담아 나중에 초콜릿과 쿠폰에 더해주도록 구현하였다.

|  |  |
| --- | --- |
| 3 | **실행결과** |

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 4 | **느낀점 및 토의사항** |

우선 초콜릿 바를 구매하면서 생기는 쿠폰으로 다시 초콜릿 바를 사고 이런 과정을 어떻게 구현해야 할지 생각해봤을 때 while문으로 처리하자는 생각이 떠올랐다. 하지만 쿠폰을 사용하고 다시 더해야 되는 과정이 있기 때문에 addTmp라는 임시 변수를 사용하여 더해야되는 값을 담고 나머지 연산을 통해 쿠폰을 사용할 수 있도록 구현하였다.

|  |  |
| --- | --- |
| 5 | **과제 수행 완료 체크 리스트** |

<체크리스트>

1. 코인을 넣으면 쿠폰을 포함해 구매할 수 있는 초콜릿 바의 개수가 정확하게 구해지는가? **예**

2. 초콜릿을 구매하고 남은 쿠폰 개수가 정확하게 구해지는가? **예**

3. 보고서에 문제 2 와 관련된 문제정의 및 분석, 해결방안과 실행결과를 충분히 설명하였는가? **예**

|  |  |
| --- | --- |
| 1 | **문제정의 및 분석** |

메뉴 2)의 문자 산수 퍼즐 문제

0~9범위의 숫자를 담고 있는 영어 알파벳 계산을 통해 문자 안에 있는 숫자를 찾는 문제이다. 여기서 조건은 2개의 문자가 같은 숫자를 가질 수 없다는 것이다. 여기서는

TOO + TOO + TOO + TOO = GOOD 라는 문자 산수의 해법을 찾아야 한다. TOO는 모두 같은 숫자이므로 TOO \* 4 = GOOD를 찾는 것이기도 하다.

|  |  |
| --- | --- |
| 2 | **주요 소스코드 설명** |

Random random = new Random();  
int[] num = new int[4];  
int GOOD;  
int TOO;  
  
do{  
 for(int i=0;i<num.length;i++){  
 num[i] = random.nextInt(10);  
 for(int j=0;j<i;j++){  
 if(num[i] == num[j]){  
 i--;  
 }  
 }  
 }  
  
 GOOD = num[2] \* 1000 + num[1]\*100 + num[1]\*10 + num[3];  
 TOO = num[0]\*100 + num[1]\*10 + num[1];  
}while(GOOD != TOO\*4);  
System.*out*.println("T = "+ num[0] + " O = "+ num[1]+" G = " + num[2]+ " D = "+ num[3]);  
System.*out*.println();

4종류의 문자를 나타내는 int배열 타입의 변수를 선언해주고 random을 통해 배열의 0번째부터 3번째까지 0~9사이의 랜덤 값을 할당해주었다. 여기서 두 개 이상의 문자는 같은 숫자를 가지지 않는다는 조건이 있어서 중복되는 숫자가 없도록 검사를 해주었다. 배열의 0번째는 T, 1번째는 O, 2번째는 G, 3번째는 D가 된다고 가정하고 각 숫자를 GOOD, TOO변수에 넣어 줬다. 이후 TOO\*4와 GOOD이 같을 때 까지 반복을 하여 같은 숫자가 나왔을 때 반복문을 나와 찾은 숫자를 출력할 수 있도록 구현했다.

|  |  |
| --- | --- |
| 3 | **실행결과** |

폰트, 텍스트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 4 | **느낀점 및 토의사항** |

우선 이 값을 어떻게 찾아야 하는지 막막하였다, 먼저 든 생각은 0~9까지의 범위의 숫자밖에 없으므로 랜덤값을 계속 돌려 각 문자를 나타내는 숫자를 찾자는 생각이 들었다. 하지만 중복처리를 하지 않고 그냥 하였더니 정답이 여러 개가 도출되었다. 그리하여 for문을 통해 중복검사와 처리를 하였더니 값 한 개가 도출될 수 있었다. 문자에 해당하는 변수를 각각 생성할 수도 있었지만 반복문의 편의성을 위해 배열을 통해 구현하였다.

|  |  |
| --- | --- |
| 5 | **과제 수행 완료 체크 리스트** |

<체크리스트>

1. 연산식을 만족하는 각 문자의 값을 찾았는가? **예**

2. GOOD 는 4 자리의 정수 값이 맞는가? **예**

3. 보고서에 문제 3 과 관련된 문제정의 및 분석, 해결방안과 실행결과를 충분히 설명하였는가? **예**

|  |  |
| --- | --- |
| 1 | **문제정의 및 분석** |

메뉴 3)의 구구단 출력하기

2단에서 9단까지 출력하되 가로로 출력되는 단수를 입력받아 해당 수 만큼 가로로 출력하는 문제이다. 출력시 자리 수를 지키고, 출력 단수는 1~8 사이의 값이 입력되어야 한다. 이외의 값이 입력된 경우 입력이 잘못 되었습니다 를 출력해야 한다. 각 단마다 세로로 출력되고 입력한 단수만큼 가로로 단들이 나열 되어 있어야한다.

|  |  |
| --- | --- |
| 2 | **주요 소스코드 설명** |

int printNum; //가로로 출력할 단수  
 System.*out*.println("=".repeat(5)+"구구단 출력하기"+"=".repeat(5));  
 while(true){  
 System.*out*.print("출력 단수 : ");  
 printNum = *scanner*.nextInt();  
 if(printNum >= 1 && printNum <=8){  
 break;  
 }  
 System.*out*.println("출력 단수 입력이 잘못 되었습니다.");  
 }  
 int cnt =2; //몇단까지 출력이 되었는지 나타내는 변수  
 int tmp=0;  
 while(cnt <=9){  
 for(int i=1;i<=9;i++) {  
 for (int j = cnt; (j < printNum + cnt) && (j<10); j++) {  
 System.*out*.printf("%2d \*%2d = %2d", j, i, i\*j);  
 System.*out*.print("\t\t");  
 tmp = j;  
 }  
 System.*out*.println();  
 }  
 cnt = ++tmp;  
 System.*out*.println("\n");  
 }  
}

우선 출력 단수를 printNum에 받을 때 while문과 break를 통해 1~8사이의 값을 받고 이외의 int값이 들어오면 while을 통해 다시 값을 입력받도록 하였다. 이후 2개의 변수를 만들어 주었다. cnt변수는 현재 몇단까지 출력되었는지 나타내주는 변수이고, tmp변수는 임시변수로서 cnt변수에 대한 값을 가지고 있는 변수이다. 이후 9단까지 출력되어야 하므로 while문을 하나 만들어주고 그 안에 단이 세로로 출력되게 해주는 for문 그안에 가로 역할을 담당하는 for문을 만들어 주었다. 현재 몇단인지 나타내는 cnt부터 printNum +cnt 미만이 될때까지 가로로 출력하면 입력한 단수 만큼 가로로 출력된다. 자리수를 맞춰야 하므로 %d를 활용해 자릿수를 맞춰주었다. tmp마지막 단수까지를 담아 1을 증가시킨후 다음에 다음 단수가 출력되도록 cnt에 담았다.

|  |  |
| --- | --- |
| 3 | **실행결과** |

텍스트, 스크린샷, 메뉴이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 4 | **느낀점 및 토의사항** |

cnt < cnt+printNum까지 하여 출력하였을 때 10단까지 출력되는 오류가 있어서 j<10까지의 and연산을 통해 최대 9단까지만 출력되도록 오류를 수정하였다.

|  |  |
| --- | --- |
| 5 | **과제 수행 완료 체크 리스트** |

<체크리스트>

1. 1~8 이외의 값이 입력되었을 경우 문제에서 요구한 대로 처리하였는가? **예**

2. 입력한 단수만큼씩 가로 출력이 잘 수행되는가? **예**

3. 출력시 숫자의 자리수는 맞춰서 출력되었는가? **예**

4. 보고서에 문제 4 와 관련된 문제정의 및 분석, 해결방안과 실행결과를 충분히 설명하였는가? **예**