

(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용SW개발자 양성과정

2021년 5월 17일
[7일차 복습]

- 수강생 : 김 민 규
- 강의장 : 강 남 C
- 수강 기간 : 2021. 05. 07 ~ 2021. 12. 08
- 수강 시간 : 15:30 ~ 22:00
- 이상훈 강사님 | 이은정 취업담임



▶ 내용 : Quiz10 _ 검사

```
1 ▶ public class b_Quiz10 {
2 ▶     public static void main(String[] args) {
3         int bcnt = 0, scnt = 0;
4
5         for (int i = 1; i <= 1000; i++) {
6             // 아래와 같은 코드는 조건이 몇 번 검사되었는지를 보는 것이 아니라
7             // if 문이 몇 번 통과되었는지를 체크하는 cnt(카운트) 값에 해당한다.
8             // 공배수 제외하면 515개가 출력됨
9             if ((i % 2 == 0) || (i % 33 == 0)) {
10                 //500 || 30
11                 bcnt++;
12
13             }
14
15             // 이하 등문
16             if ((i % 33 == 0) || (i % 2 == 0)) {
17                 scnt++;
18
19             }
20
21         }
22
23         System.out.printf("많은 수/작은 수가 있던 이 경우는 모두 %d, %d", bcnt, scnt);
24     }
25 }
26
27
```

주요 내용

--

왼쪽의 if문은
몇번 통과되는지만 보유중
공배수 제외하면 515 출력

--

▶ 내용 : new문을 이용한 배열동적할당

```
1  import java.util.Scanner;
2
3  ▶ public class c_new_배열동적할당 {
4  ▶   public static void main(String[] args) {
5      Scanner scan = new Scanner(System.in);
6      System.out.print("학급에 학생이 몇 명 있습니까 ? ");
7
8      int studentNum = scan.nextInt();
9      int studentArr[] = new int[studentNum];
10
11     for(int i = 0; i < studentNum; i++) {
12         studentArr[i] = (int)(Math.random()*21)+80;
13         System.out.printf("studentArr[%d] = %d\n", i, studentArr[i]);
14     }
15 }
16
17
```

주요 내용

--

동적할당되는 데이터를 관리하는 메모리 Heap

--

New로 만든 데이터는 전부 Heap에서 관리

--

기존 stack에서 배열을 만들 때에는 할당된 값에 의하여 개수가 고정되었다는 한계가 있다.

--

Heap에서는 공간을 필요한만큼 할당할 수 있어 유연하게 접근이 가능함.

--

다만 stack보다 느리다.

--

New를 통해 공간을 만드는 법

1. new 를 적는다.
2. 데이터타입을 적는다.
3. 만약 데이터타입이 배열이라면 대괄호를 열고 몇 개를 만들지 적는다.
만약 데이터타입이 클래스라면 소괄호를 열고 필요하다면 인자를 설정한다.
만약 데이터타입이 클래스라면 소괄호를 열고 닫고 필요하다면 인자를 설정한다.
(아직 클래스는 배우지 않았으므로 패스)

▶ 내용 : 배열길이 구하기

```
1 ▶ public class d_배열길이구하기 {  
2 ▶     public static void main(String[] args) {  
3         int arr[] = { 2, 3, 4, 5, 6, 7};  
4         System.out.println("arr의 길이" + arr.length);  
5  
6         int dynamicArr[] = new int [30];  
7  
8         System.out.println("dynamicArr의 길이 = " + dynamicArr.length);  
9     }  
10 }  
11
```

주요 내용

--

Arr의 경우 2,3,4,5,6,7 총 6개가 length

dynamicArr의 경우 30개가 length

--

▶ 내용 : 배열_등비수열구하기

```
1  import java.util.Scanner;
2
3  ▶ public class e_등비수열1 {
4  ▶  public static void main(String[] args) {
5      System.out.println("1, 2, 4, 8, ... 1024, ...");
6
7      Scanner scan = new Scanner(System.in);
8
9      System.out.print("몇 번째 항을 구할까요 ? ");
10     int num = scan.nextInt();
11
12     if (num <= 0) {
13         System.out.println("잘못된 값을 입력하였습니다.");
14     } else if (num < 2) {
15         System.out.println("당신이 찾는 값은 1입니다.");
16     } else {
17         int numArr[] = new int[num];
18
19         numArr[0] = 1;
20
21         for (int i = 1; i < numArr.length; i++) {
22             numArr[i] = numArr[i - 1] * 2;
23         }
24
25         System.out.printf("%d 번째 항은 = %d\n", num, numArr[num - 1]);
26     }
27 }
28 }
29 }
30 }
```

주요 내용

--

numArr[i-1]은 0부터 배열이 시작하기
때문에

*2는 2의배수

--

▶ 내용 : 배열_등비수열구하기_pow

```
1  import java.util.Scanner;
2
3  ▶ public class e_등비수열2 {
4  ▶   public static void main(String[] args) {
5      System.out.println("1, 2, 4, 8, ... 1024, ...");
6
7      Scanner scan = new Scanner(System.in);
8
9      System.out.print("몇 번째 항을 구할까요 ? ");
10     int num = scan.nextInt();
11
12     if (num <= 0) {
13         System.out.println("잘못된 값을 입력하였습니다.");
14     } else if (num < 2) {
15         System.out.println("당신이 찾는 값은 1입니다.");
16     } else {
17         int numArr[] = new int[num];
18
19         numArr[0] = 1;
20
21         for (int i = 1; i < numArr.length; i++) {
22             // Math.pow(A, B)는 A^B(A의 B승)을 계산한다.
23             // Math.pow는 double을 결과로 내놓기 때문에 강제로 int 타입으로 변형하였음
24             numArr[i] = (int)Math.pow(2, i);
25         }
26
27         System.out.printf("%d 번째 항은 = %d\n", num, numArr[num - 1]);
28     }
29 }
30 }
31 }
```

주요 내용

--

numArr[0] = 1;
1부터 등비수열 시작하기 때문에

(int) Math.pow
Pow는 결과를 double로 내놓음
따라서 int로 변형시켜야함

Pow는 >>와 유사하다 볼 수 있음

--

▶ 내용 : class_커스텀 데이터타입

```
1 class Person {
2     int age;
3     String name;
4 }
5 public class f_class_커스텀데이터타입 {
6     public static void main(String[] args) {
7         Person person = new Person();
8
9         person.age = 21;
10        person.name = "태식이";
11
12        System.out.println("몇살?" + person.age);
13        System.out.println("이름은?" + person.name);
14    }
15 }
16
17
```

주요 내용

--

여기서 ‘. ‘은 공간 내부에 접근하겠다는 의미를 가짐

즉 변수가 가진 공간 중 age에 21저장
Name에 태식이 저장

--

▶ 내용 : class_커스텀 데이터타입_%s는 문자열 출력하는데 사용.

```
1  import java.util.Scanner;
2
3  class Student {
4      String major;
5      String name;
6      int age;
7  }
8
9
10 public class e_class예제_printf_s {
11     public static void main(String[] args) {
12         System.out.println("학생 본인의 신상을 입력해주세요.");
13
14         Student stu = new Student();
15
16         Scanner scan = new Scanner(System.in);
17         System.out.print("전공을 입력하세요: ");
18         stu.major = scan.nextLine();
19
20         System.out.print("이름을 입력하세요: ");
21         stu.name = scan.nextLine();
22
23         System.out.print("나이를 입력하세요: ");
24         stu.age = scan.nextInt();
25
26         // %s의 경우에는 printf에서 문자열을 출력하는데 사용한다.
27         System.out.printf("이름: %s, 전공: %s, 나이: %d\n", stu.name, stu.major, stu.age);
28     }
29 }
```

주요 내용

--

Q.

%d의 경우 문자열작성시 오류가 나지만,
%s의 경우 숫자 문자 상관없이 출력됨.

%s에 숫자를 입력하면 인식이나 출력에
%d와 상이한 부분이 있는지
궁금합니다.

--

▶ 내용 : Getter, Setter

선 제일 간단한 Getter, Setter 부터 갑니다.

// 마우스 커서를 아래 갖다 놓고 Alt + Insert를 누른다
// Getter & Setter -> 전부 선택하고 OK 하면 자동으로 Setter와 Getter가 만들어진다.
// 만약 키보드 설정이 안맞는 경우 Code -> Generate로 같은 작업을 할 수 있다.

// 매서드를 만드는 방법
// 1. 먼저 리턴(return) 타입을 작성한다.
// 2. 매서드의 이름을 작성한다(용도에 맞게 작성한다)
// 보통 Getter의 경우 값을 얻고자 할 때(즉 return 용도로 사용)
// Setter의 경우 값을 설정하고자 할 때 사용한다.
// 그 외에도 커스텀 매서드의 경우엔
// 자동 완성 기능으로 만들 수 없기 때문에 매서드 작성법에 대해 알 필요가 있다.
// 3. 소괄호 내부에 인자로 입력 받을 매개변수를 설정한다.
// 4. 중괄호 내부에 해당 매서드(기능)이 수행할 업무를 작성한다.

// 누군가가 age가 궁금해서 물어본다.
// 이에 대한 답을 해준다면 Getter에 해당한다.
// 클래스 작성할 때는 이니셜마다 대문자를 붙였다.
// 매서드는 시작은 소문자 그 이후부터의 이니셜은 대문자
// 소괄호 내부는 인자가 배치되는데 텅 비어 있는 것은 인자(입력)이 없다는 뜻이다.
// 결론: 값을 얻는 목적으로 사용하는 것이 Getter

회원이입시 집 주소 적으라고 나오는데 여기서 입력하는 값들을 처리하는게 Setter에 해당한다.
특정한 값을 설정하는 목적으로 Setter가 사용된다.
결론: 값을 설정하는 목적으로 사용하는 것이 Setter

this.age는 클래스 내부에 있는 age를 의미한다.
age는 입력으로 들어온 age에 해당한다.

주요 내용

--

--

▶ 내용 : Getter, Setter

```
1  class Teacher {
2      int age;
3      String name;
4      String major;
5
6      int getAge() {
7          return age;
8      }
9
10     void setAge(int age) {
11
12         this.age = age;
13     }
14
15     String getName() {
16         return name;
17     }
18     void setName(String name) {
19         this.name = name;
20     }
21
22     String getMajor() {
23         return major;
24     }
25     void setMajor(String major) {
26         this.major = major;
27     }
28 }
29 public class g_Getter_Setter {
30     public static void main(String[] args) {
31         Teacher t = new Teacher();
32
33         t.setAge(40);
34         t.setName("태식");
35         t.setMajor("경영");
36
37         System.out.printf("%s 는 %s를 전공하였고 %d 세다.\n", t.getName(), t.getMajor(), t.getAge());
38     }
```

주요 내용

--

Getter는 값을 얻고자 할 때 작성
Setter는 값을 설정하고자 할때 작성

--

▶ 내용 : Getter, Setter_Quiz32

주요 내용

```
1  class 멍멍이 {
2
3      int age;
4      String name;
5      String species;
6
7      int getAge() {
8          return age;
9      }
10     void setAge(int age) {
11         this.age = age;
12     }
13     String getName() { return name; }
14     void setName(String name) {
15         this.name = name;
16     }
17     String getSpecies() { return species; }
18     void setSpecies(String species) { this.species = species; }
19
20 }
21
22 public class h_Quiz32 {
23     // 멍멍이 클래스
24
25     public static void main(String[] args) {
26         멍멍이 dog = new 멍멍이();
27
28         dog.setAge(3);
29         dog.setName("가루");
30         dog.setSpecies("치와와");
31
32         System.out.printf("나이 : %d / 이름 : %s / 견종 : %s\n", dog.getAge(), dog.getName(), dog.getSpecies());
33     }
34 }
```

▶ 내용 : Getter, Setter_Quiz33

주요 내용

```
1 class 고양이 {
2     int a;
3     int p;
4     String f;
5
6     int getA() {
7         return a;
8     }
9     void setA(int a) {
10        this.a = a;
11    }
12    int getP() {
13        return p;
14    }
15    void setP(int p) {
16        this.p = p;
17    }
18    String getF() {
19        return f;
20    }
21    void setF(String f) {
22        this.f = f;
23    }
24 }
25 public class i_Quiz33 {
26     //고양이
27     public static void main(String[] args) {
28         고양이 cm = new 고양이();
29         cm.setA(13);
30         cm.setF("영희");
31         cm.setP(3);
32         System.out.printf("길고양이가 몇마리있나요? %d / 몇분이서 관리하시나요? %d / 누가 밥을주시나요? %s\n", cm.getA(), cm.getP(), cm.getF());
33     }
34 }
```

--

--