문제풀이

[6-1 & 6-2]

```
class SutdaCard
{
  int num;
  boolean isKwang;

  SutdaCard()
  {
    this.num = 1;
    this.isKwang = True;
  }
  SutdaCard(int num, boolean isKwang)
  {
    this.num = num;
    this.isKwang = isKwang;
  }
  String info()
  {
    String result
    result += num;
    if (isKwang)
      result += "K";
    return result;
  }
}
```

[6-3 & 6-4]

```
class Student
{
  String name;
  int ban;
  int no;
  int kor;
  int eng;
  int math;

  int getTotal()
  {
    int result = this.kor + this.eng + this.math;
    return result;
  }

  float getAverage()
  {
    float result = (this.kor + this.eng + this.math)/(float)3;
```

```
    return result;

  }
}
```

> **236 / 3 → 78**
> **236 / 3f → 78.666664**
> **236 / 3f * 10 → 786.66664**
> **236/3f * 10 + 0.5 (반올림을 위함) → 787.16664**
> **(int)(236/3f * 10 + 0.5) → 787**
> **787 / 10f → 78.7**

[6-5]

```
class Student
{
  Student(String name, int ban, int no, int kor, int eng, int math)
  {
    this.name = name;
    this.ban = ban;
    this.no = no;
    this.kor = kor;
    this.eng = eng;
    this.math = math;
  }
  String info()
  {
    String result = name + "," + no + "," + kor
      + "," + eng + "," + math + "," + this.getTotal() + "," + this.getAverage();
    return result;
  }
}
```

[6-6]

```
static double getDistance(int x, int y, int x1, int y1)
{
  double result = (x1 - x)*(x1 - x) + (y1 - y)(y1 - y);
  result = Math.sqrt(result);
  return result;
}
```

[6-7]

```
class MyPoint {
  int x;
  int y;

  MyPoint(int x, int y)
  {
    this.x = x;
    this.y = y;
  }

  double getDistance(int x1, int y1)
  {
    double result = (x1 - this.x)*(x1-this.x) + (y1 - this.y)*(y1-this.y);
    result = Math.sqrt(result);
    return result;
  }
}
```

[6-8]

- **클래스변수(static 변수)** : width, height
- **인스턴스변수** : kind, num;
- **지역변수** : k , n, card

[6-9]

- static 을 붙여야 하는 것 → weapon , armor : 모든 병사들의 공격력과 방어력을 공유해야 하므로.

[6-10] 다음 중 생성자에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

1. 모든 생성자의 이름은 클래스의 이름과 동일해야한다.
2. **생성자는 객체를 생성하기 위한 것이다.**
   - 생성자가 객체를 생성할 때 사용되기는 하지만, 객체를 초기화할 목적으로 사용되는 것이다. 객체를 생성하는 것은 new 연산자이다.
3. 클래스에는 생성자가 반드시 하나 이상 있어야 한다.
   - 없으면 컴파일러에 의해 기본 생성자가 추가된다.
4. 생성자가 없는 클래스는 컴파일러가 기본 생성자를 추가한다.
5. **생성자는 오버로딩 할 수 없다.**

[6-11] 다음 중 this에 대한 설명으로 맞지 않은 것은? (모두 고르시오)

1. 객체 자신을 가리키는 참조변수이다.
2. **클래스 내에서라면 어디서든 사용할 수 있다.**
3. 지역변수와 인스턴스변수를 구별할 때 사용한다.
4. 클래스 메서드 내에서는 사용할 수 없다.

this는 인스턴스 자신의 주소를 저장하고 있으며, 모든 인스턴스 메서드에 숨겨진 채로 존재하는 지역변수이다. 그래서 인스턴스메서드 내에서만 사용할 수 있다.

[6-12] 다음 중 오버로딩이 성립하기 위한 조건이 아닌 것은? (모두 고르시오)

1. 메서드의 이름이 같아야 한다.
2. 매개변수의 개수나 타입이 달라야 한다.
3. **리턴 타입이 달라야 한다.**
4. **매개변수의 이름이 달라야 한다.**

[6-13] 다음 중 아래의 add 메서드를 올바르게 오버로딩 한 것은? (모두 고르시오)

> **long add(int a, int b) {return a + b;}**

1. long add(int x, int y) {return x +y;}
2. **long add(long a, long b) {return a + b;}**
3. **int add(byte a, byte b) {return a + b;}**
4. **int add(long a, int b) { return (int)(a+b);}**

[6-14] 다음 중 초기화에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

1. 멤버변수는 자동 초기화되므로 초기화하지 않고도 값을 참조할 수 있다.
    - 기본값으로 초기화 되므로.
2. 지역변수는 사용하기 전에 반드시 초기화해야 한다.
3. **초기화 블럭보다 생성자가 먼저 수행된다.**
4. 명시적 초기화를 제일 우선적으로 고려해야 한다.
5. **클래스변수보다 인스턴스 변수가 먼저 초기화된다.**

[6-15] 다음 중 인스턴스 변수의 초기화 순서가 올바른 것은?

1. **기본값-명시적초기화-초기화블럭-생성자**

2. 기본값-명시적초기화-생성자-초기화블럭
3. 기본값-초기화블럭-명시적초기화-생성자
4. 기본값-초기화블럭-생성자-명시적초기화

[6-16] 다음 중 지역변수에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

1. **자동 초기화되므로 별도의 초기화가 필요없다.**
2. 지역변수가 선언된 메서드가 종료되면 지역변수도 함께 소멸된다.
3. 메서드의 매개변수로 선언된 변수도 지역변수이다.
4. 클래스변수나 인스턴스 변수보다 메모리 부담이 적다.
5. **힙(heap)영역에 생성되며 가비지 컬렉터에 의해 소멸된다.**
    - vs 스택(stack)
    - 힙(heap) 영역에는 인스턴스(인스턴스 변수)가 생성되는 영역이며, 지역변수는 호출스택(call stack)에 생성된다.

[6-17] 호출스택이 다음과 같은 상황일 때 옳지 않은 설명은? (모두 고르시오)

main → method2 → method1 → println

1. 제일 먼저 호출스택에 저장된 것은 main 메서드이다.
2. **println 메서드를 제외한 나머지 메서드들은 모두 종료된 상태이다.**
3. method2 메서드를 호출한 것은 main 메서드이다.
4. **println 메서드가 종료되면 method1 메서드가 수행을 제개한다.**
5. main-method2-method1-println 의 순서로 호출되었다.
6. 현재 실행중인 메서드는 println 뿐이다.

[6-18]

1. **라인 A - static 변수의 초기화에 인스턴스 변수를 사용할 수 없다. 꼭 사용해야 한다면, 객체를 생성해야 한다.**
2. **라인 B** → static method 에서는 인스턴스 변수를 참조할 수 없다.
3. **라인 D** → 마찬가지.

[6-19]

“ABC123”

“After change : ABC123”

- 문자열은 내용을 변경할 수 없기 때문에 덧셈연산을 하면 새로운 문자열이 생성되고 새로운 문자열의 주소가 변수 str에 저장된다. 이때 변수 str 은 지역변수.

[6-20]

```
int[] shuffle(int[] arr)
{
  for (int i = 0; i < arr.length; i++)
  {
    int random = (int)(Math.random() * arr.length);
    int temp = arr[i];
    int shuffle = arr[random];

    arr[i] = arr[random];
    arr[random] = temp;
  }
}
```

[6-21]

```
void turnOnOff() {
  isPowerOn == true ? (isPowerOn = false) : (isPowerOn = true);
}
void volumeUp() {
  if (this.volume < MAX_VOLUME)
    this.volume++;
}
void volumeDown() {
  if (this.volume > MIN_VOLUME)
    this.volume--;
}
void channelUp() {
  if (this.channel >= MAX_CHANEEL)
    this.channel = MIN_CHANNEL;
  else
    this.channel++;
}
void channelDown() {
  if (this.channel <= MIN_CHANNEL)
    this.channel = MAX_CHANNEL;
  else
    this.channel--;
}
```

[6-22]

```
boolean isNumber(String str)
{
  for (int i = 0; i < str.length(); i++)
  {
    char target = str.charAt(i);

    if (0 <= target - '0' <= 9)
      continue;
    else
      return false;

    return true;
  }
}
```

[6-23]

```
int max(int[] arr) {
  if (arr == NULL || arr.length = 0)
    return -999999;
  else
    int max = arr[0];
    for (int i = 0; i < arr.length; i++)
    {
      if (max < arr[i]
        max = arr[i];
    }
    return max;
}
```