

# Data Structure

**Fall 2019**

**M 16:00-18:00 W 11:00-13:00**

**<http://smart.hallym.ac.kr>**

**Instructor: Jin Kim**

**010-6267-8189(033-248-2318)**

**[jinkim@hallym.ac.kr](mailto:jinkim@hallym.ac.kr)**

**Office Hours:**

# Lab(Stack1)

**Fall 2019**

**<http://smart.hallym.ac.kr>**

**Instructor: Jin Kim**  
**010-6267-8189(033-248-2318)**  
**[jinkim@hallym.ac.kr](mailto:jinkim@hallym.ac.kr)**

**Office Hours:**

- ◆ ArrayStack.java
- ◆ LinkStack.java
- ◆ StackMain.java
- ◆ 를 사용하여 다음을 수행하라.

# Array-based Stack method implementation

1. 주어진 배열로 이루어진 스택 프로그램에 내용을 채워넣어라.

Peek() //top이 가르키는 원소의 내용을 return한다.

delete()// top이 가르키는 원소를 제거만 한다. 값을 return할 필요없다.

2. 배열을 사용하여 스택을 구현하면 Stack overflow를 고려해야한다. Stack overflow가 발생하면 새로운 보다 더 큰 배열을 만들고, 그 배열에 현재 스택배열의 내용을 복사한 후, 그 배열을 스택으로 사용하도록 해보자. 프로그램을 확인하라. Pop(), delete() 메소드에서 현재 원소의 개수가 스택크기의 50%보다 작으면 스택크기를 반드로 줄이는 코드를 추가하라.

3. push(10)-push(20)-push(30)-pop(push(40))-pop()-pop()-push(pop())-pop()등으로 테스트해보라.

# Linked list based Stack method implementation

1. 주어진 연결리스트로 이루어진 스택 프로그램에 내용을 채워넣어라.

Peek() //top이 가르키는 원소의 내용을 return한다.

delete()// top이 가르키는 원소를 제거만 한다. 값을 return 할 필요없다.

Empty() // 스택에 원소가 있는지의 여부를 계산한다.

3. push(10)-push(20)-push(30)-pop(push(40))-pop()-pop()-push(pop())-pop()등으로 테스트해보라.

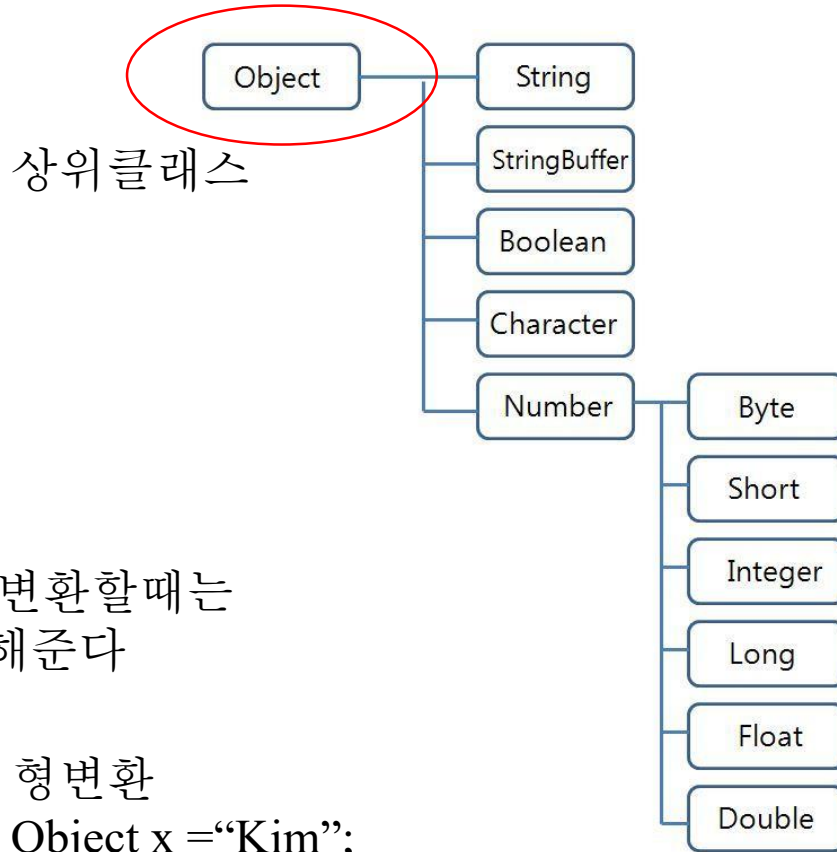
\* 스마트 캠퍼스에 업로드하라.

# Generic을 이해하자

- ◆ StackGenericMain.java
- ◆ ArrayStackGeneric.java

# Object 클래스를 이해하자

# Java.lang 패키지 클래스 구조



상위에서 하위로 형변환할 때는  
명시적으로 형변환해준다

형 변환

```
Object x =5;  
Integer y;  
y=(Integer) x;
```

형 변환

```
Object x ="Kim";  
String z;  
z=(String) x;
```

제네릭 형 변환

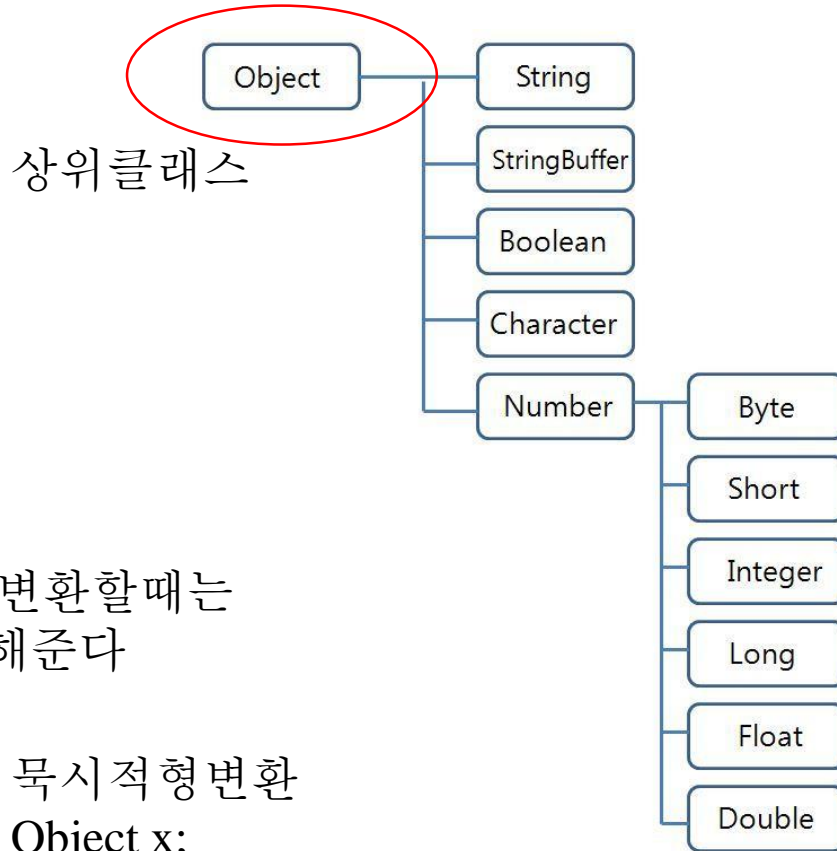
```
AAA<String> m = new AAA<>(); 혹은  
AAA<Integer> m = new AAA<>();
```

AAA 빵틀에서 제네릭 형 변환

```
Object x ="Kim";  
E z;  
z=(E) x;
```



# Java.lang 패키지 클래스 구조



하위에서 상위로 형변환할 때는  
묵시적으로 형변환해준다

묵시적 형 변환  
Object x ;  
Integer y=5;  
X=y;

묵시적 형 변환  
Object x;  
String z="Kim";  
x= z;

◆ 정수를 저장하는 스택을 만들어라.

# Postfix(후위표기식계산)

- ◆ 주어진 제너릭 스택관련 프로그램과 연결하여 후위표기식 계산프로그램을 완성하라.
  - ◆ PostExpMain.java
  - ◆ ListEvalPostfix.java
  - ◆ 위의 ListEvalPostfix.java를 참고하여 EvalPostfix.java를 만들어라.  
다른 부분은 고칠 곳이 없고, 스택생성부분 한줄과 메소드이름정도 고치면 될 것이다.

- ◆ 프로그램을 zip한 후, 업로드하라.