# Scope: Some midterm questions + following questions

중간고사 문제:

<https://docs.google.com/document/d/1TkHWtERD3rP_WxW87wIs49_0iDe0GCc5NYTn2fSJgVw/edit>

## 1. Discuss about inheritance in terms of the 'is-a' relationship with an example.

* 특정 객체가 다른 객체에게 자신의 능력을 물려주는 상속관계를 의미한다. 비슷한 속성 및 동작을 가진 객체들 사이에서 공통점을 추출하여 상위 개념의 객체로 만들 수 있으며, 하위 객체들의 상위 개념의 객체를 상속 받아서 사용함으로써 중복되는 기능 및 속성을 제거하고 재사용성을 향상시킬 수 있다. 상위 개념을 포함하여 하위 객체에 상속해 주는 것의 상위 클래스(super class)이고, 상위 객체를 상속받아 사용하는 것이 하위클래스(sub class)이다. 대부분의 경우 하위클래스는 상위클래스의 기능에 더하여 조금 더 세부적인 기능들을 가질 수 있도록 설계되어진다. 이 때, 하위클래스의 객체는 상위클래스에서 자료형인 것처럼 사용되어질 수 있으나, 그 반대는 성립하지 않는다.
* ex) Dog is an Animal
  + Super class = Animal
  + Sub class = Dog
* Dog is an Animal 이라고 하는 것은 성립되지만, 모든 동물이 개는 아니기 때문에 Animal is an Dog는 성립하지 않는다.
* 이렇게 Dog is a Animal 이라고 말할 수 있는 관계를 IS-A 관계, 즉 상속 이라고 한다.

## 부모(상위)클래스와 자식(하위)클래스가 있으며, 자식클래스는 부모클래스의 멤버를 상속받아 쓸 수 있다. Eg. Student is a Person. Student = 자식클래스. Person = 부모클래스

## 2. Discuss about the 'Object' class.

- 자바에서 모든 클래스는 사실 Object를 암시적(자동적)으로 상속받고 있는 것이다. 그런 점에서 Object는 모든 클래스의 조상이라고 할 수 있다. Object 클래스를 자동적으로 상속 받는 이유는 모든 클래스가 공통으로 포함하고 있어야 하는 기능을 제공하기 위해서다.

Object class에는 toString, equals 등의 메소드가 있다.

- Object 클래스는 자바 API의 모든 클래스와 사용자가 정의한 모든 클래스의 최상위 클래스로, Java에서 만드는 모든 클래스는 Object라는 클래스로부터 상속 받는다. 하지만 모든 클래스들이 Object 클래스를 자동으로 상속받게끔 되어 있어 굳이 상속하도록 코딩하지 않아도 Object 클래스의 모든 method와 변수들을 사용할 수 있다.

## 3. Provide an example scenario where we need interfaces in a real world software project.

Ex 1)

- 가정: 계산기 기능이 필요한 프로젝트를 진행하는데, 계산기 클래스를 만드는 개발자 A와 그 클래스를 사용하여 로직을 만드는 개발자 B가 있다. 이 둘은 작업시간을 단축하기 위해 동시에 작업을 진행한다. 이 경우 가짜 클래스(실질적인 논리들은 없이 값을 고정해서 사용하도록 정의만 해 놓은 클래스)를 만들어서 사용하게 되는데, 상호간에 구체적인 약속을 하지 않으면 아래와 같은 상황이 발생할 가능성이 생기게 된다.

* 개발자 A는 두개의 변수를 받아 사용하도록 계산기 클래스를 만들었다.
* 개발자 B는 계산기 클래스가 세개의 변수를 받아 사용하도록 작성되었을 것을 기대하며 하위 로직을 만들었다.

- 이 경우 서로의 코드가 맞지 않아 작업을 처음부터 다시 해야 하는 상황이 발생한다. 이때 계산기를 위한 인터페이스를 사용하여 서로가 동일한 메소드를 만들도록 약속을 체결하여 공유하면, 상대의 일정이나 구현하는 방식에 상대적으로 적은 영향을 받으면서 작업을 진행할 수 있다.

Ex 2)

navigation과 gps를 합치는 하청 회사가 있다고 하자.

자동차 소프트웨어 회사는 gps의 heading만을 필요로 한다.

그래서 하청회사에서 넘겨주는 자료에서 사용자가 필요로 하는 정보만 사용

→ 융통성을 가지고 서로 다른 회사 개발자들이 재사용 교류나 의사소통 가능

→ 네비게이션 내의 소프트웨어를 개발하는 회사 내에서 하청 회사에게 Interface를 받으면 서로 다른 개발자들끼리 함께 개발할 때 Interface를 보며 서로 교류하고 의사소통하며 만들기 쉽다.

ex) Comparable 을 implements 하고 compareTo() 라는 함수만 만들어주면

Arrays.sort(); 를 사용하여 쉽게 정렬이 가능하다

## 4. What is polymorphism and give an example.

다형성: 인스턴스가 자바 프로그램 안에서 다양한 형태로 존재할 수 있는 것. //

Undergraduate ug = new Undergraduate("Nam, JC", 1111, 4);

Student st = ug;

Person ps = ug;

Object obj = ug;

이 예에서 Undergraduate class는 Student를 상속 받았고, Student는 Person을 상속받았고, 모든 클래스는 Object를 상속 받았기 때문에, ug라는 인스턴스가 다양한 형태의 인스턴스로 할당되어 존재할 수 있다. 다양한 interface들을 implements한 클래스도 해당 interface로 선언된 인스턴스로 존재할 수 있다.

interface로 다형성을 설명한 예시

Public interface TV { … }

Public interface UHDTV{...}

Public class LedTV implements TV, UHD { … }

Public class Main {

Public static void main(String[] args) {

//이 부분이 위의 밑줄 친 부분. LedTV 인스턴스가 TV와 UHDTV등 (다)양한 (형)태로 표현됨.

TV tv = new LedTV();

UHDTV uhdTV = tv;

}

}

## 5. Discuss about the reserved word, *instanceof*. (What is it for and/or When do we need this?)

Person me = new Undergraduate();

Person you = new Student();

ArrayList<Person> personList = new ArrayList<Person>();

personList.add(me);

personList.add(you);

for(Person p: personList){

if(p instanceof Student)

p.studentUniqueAction(); //Student class( Not Undergraduate) //에만 존재하는 메소드

}

//만약 if(p instanceof Student) 부분이 없다면 Undergraduate 로 인스턴스화 된 me 에서 Student 메소드를 실행시키기 때문에 오류가 난다.

Reserved word가 뭐죠?

→ 변수로 선언 불가능한 단어, 말 그대로 이미 예약되어 있어서 변수나 Class 이름, 메소드 이름 등으로 사용자가 임의로 선언하는 것이 불가능하다. e.g., if, for 처럼.

**Reserved word**는 자바 언어 자체에서 의미가 약속되어 등록되어 있는 단어를 뜻한다. 예약어로 등록된 단어는 속성의 변수명, 메소드명, 클래스명 등으로 사용자가 임의로 선언하는 것이 불가능하다. 프로그래밍 과정에서 예약어를 식별자로 사용하게 되면 에러나 버그가 발생하거나, 자바가 가진 본래의 기능을 사용할 수 없게 된다.

* 예약어는 프로그래밍 과정 중에 발생하는 시행 착오를 줄이기 위해 사용된다.

**instanceof** 연산자는 프로그램 실행 시 참조 데이터형을 검사하기 위해 사용되는 연산자이다. 왼쪽에 있는 객체가 오른쪽에 있는 클래스의 객체이거나 하위클래스의 객체일 경우 true를 반환하고 그렇지 않을 경우 false를 반환한다. 왼쪽에 있는 객체와 오른쪽에 있는 클래스가 아무런 관계가 없거나 객체가 아닌 다른 데이터형을 비교하려고 할 경우 error가 발생한다.

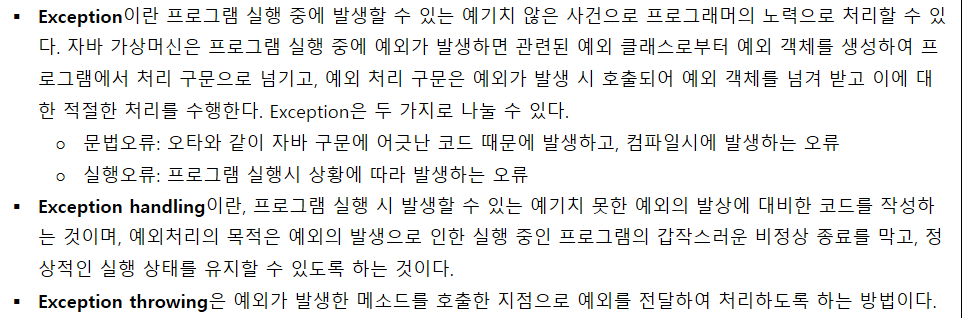
* instanceof 연산자는 참조변수가 참조하고 있는 인스턴스의 실제 타입을 알아보기 위해 사용된다.

## 6. Discuss about dynamic binding (late binding)

슈퍼클래스나 인터페이스로 선언된 변수의 실제 인스턴스 타입이 런타임시 해당 변수와 binding(결정) 되는 것. 실제 인스턴스 타입에 따라 부를 수 있는 method가 달라진다.

예를들어, 우리가 슈퍼클래스인 Person의 인스턴스를 불러들었지만 프로그램이 실행 될 때 Person type으로 선언된 변수에 할당된 실제 서브클래스가 Student의 인스턴스라면 Student로 캐스팅 후 Student 인스턴스만 가지고 있는 method를 사용할 수 있게 된다. 하지만, 런타임시 서브클래스가 Undergraduate 인스턴스라면 Student 인스턴스만 가지고 있는 메소드를 부를 수가 없게 된다.

## 7. What is an exception, throwing an exception, and handling an exception?



exception이란? 내가 짠 프로그램에서 의도하지 않은 예외적인 상황이 벌어지는 경우 내가 작성한 코드로 exception은 우아하게 처리해 줄 수 있음 => 문제가 무엇인지 알고, 또 처리 할 수 있음! 여기에서 예외상황 발생시 exception을 발생시켜서 추후에 처리할 수 있도록 준비하는 것을 'throwing', 해결해 주는 것을 'handling' 이라고 함. 예외상황이 발생할 수 있는 코드 영역을 Try block로 감싸주고 실제로 exception이 try block에서 throwing 됐을 때, Catch block에서 exception을 처리(handling)해주는 것임.

## 8. List at least three predefined exception classes and briefly explain them.

EOFException: 파일의 끝에 도달했을 때,

FileNotFoundException: 파일이 발견되지 않았을 때,

NullPointerException: null인 오브젝트를 접근했을 때,

StackOverflowError: 스택이 초과되었을 때.

IOException: 입출력시에 지정한 파일 이 시스템에 존재하지 않을 때,

IndexOutOfBoundsException: 접근가능한 배열의 index를 벗어났을 때.

## 9. Discuss how a text file and a binary file are different with an example.

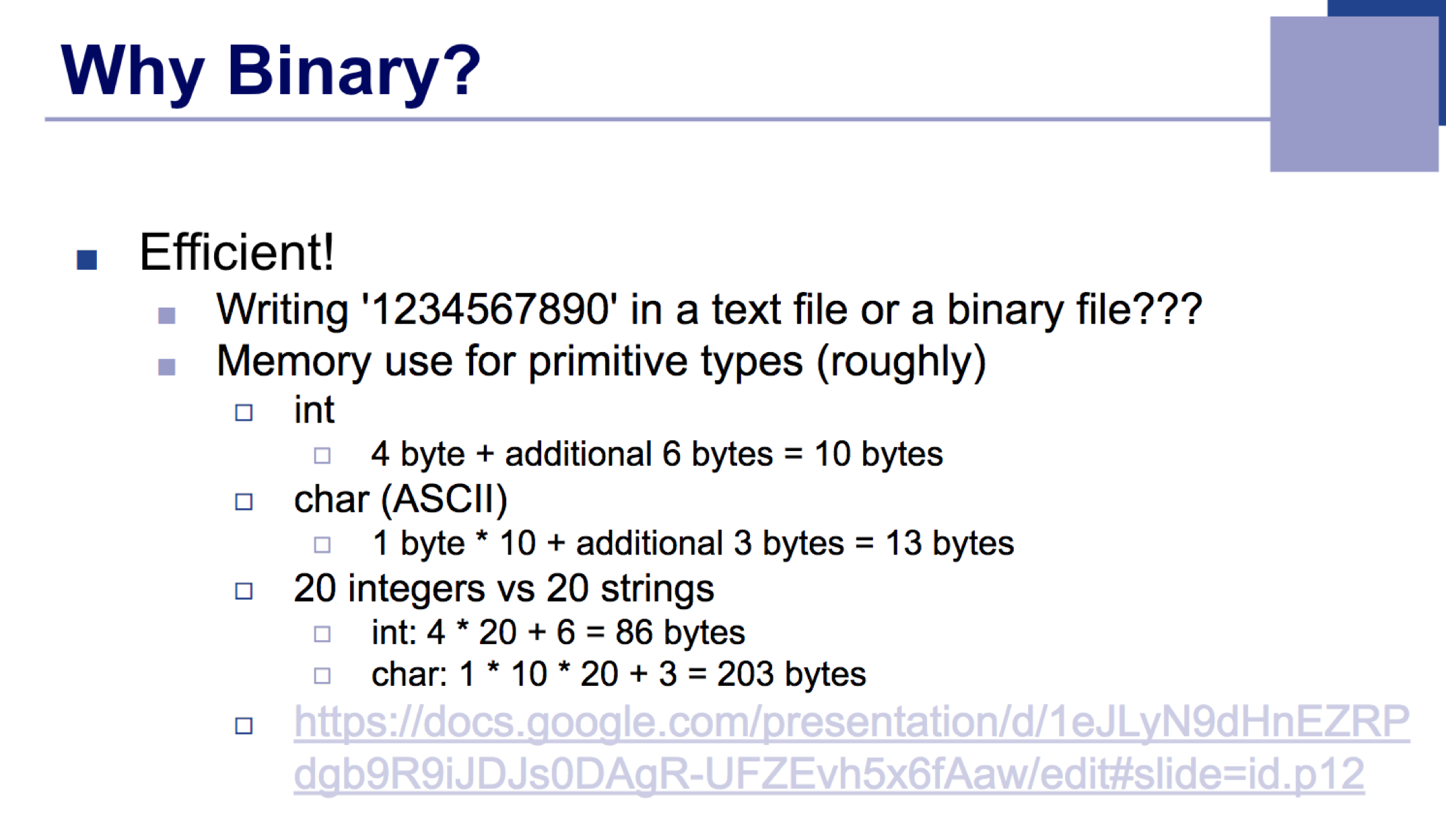
<https://docs.google.com/presentation/d/19_1kQalDTG8MEERIvj27VjJehhDjIZkTbYJfPWkiius/edit#slide=id.p8>

Text file(treated as sequence of characters)

: Java program source code

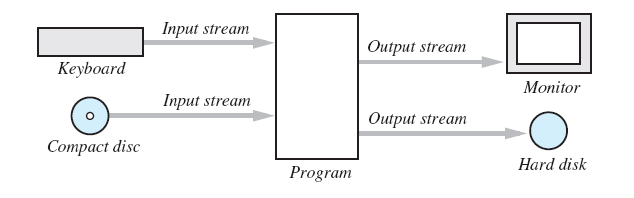
Binary file (All other files)

: Movie, music files



바이너리로 저장할때는 정수로 보고 4바이트로 저장하고 텍스트 파일일 때는 char형으로 각자를 1바이트로 봐서 10바이트 저장한다 => 메모리의 효율

## 10. What is a stream and explain streams with a figure we learned in the class.



* A stream is a flow of input or output data

=> 데이터의 인풋/아웃풋의 흐름(또한 스트림의 실행을 위한 특정 클래스가 있다), 다른 표현으로는 자료의 입출력을 도와주는 중간 매개체. 혹은 중간 이동통로 정도로 비유 가능

## 11. Write single line comments the following source code. (Reading/writing a file/an object/an array will be given.)

## 12. Write a statement that defines a Hashmap instance named '*myMap*' and its key is a *String* type and its value type is an *ArrayList* that can have an *Integer* instance as an element. (Underlined ones can be changed in the real exam question.)

HashMap<String,ArrayList<Integer>> myMap = new HashMap<String,ArrayList<Integer>>();

*//HashMap 형식: HashMap<’key\_type’,’value\_type’>*

## 13. Write single lines comments about the following source code. (Concurrency related code will be given)n

Concurrency 슬라이드: <https://docs.google.com/presentation/d/1NFmRs2OXPHb_z6Ql-p0WauvHFXzCSv757ZxzXdeYZWw/edit#slide=id.g3ba561cbcf_0_58>

## 14. Design a java program by using class diagrams for the given problem.

예) 다음과 같은 csv 파일 여러개를 받아 총액이 가장 높은 주문 내역 순서대로 csv 파일로 나타내시오

Order1.csv

Name,price($),count

Apple ,2,2

Banana,4,6

…

…

⇒

| OrderSorter |
| --- |
| - pathToCSV:String |
| + main(String argv[]): void |

| Order |
| --- |
| - items: Item[]  - totalPrice : double |
| - calcTotalPrice(): double  +getTotalPrice(): double  + compareTo(): int |

…

| Item |
| --- |
| - price: double  - count: int |
| +getPrice(): double  +getCount(): double |

| FileReader |
| --- |
| - filePath: String |
| + readFiles() : Order |

| FileExporter |
| --- |
| - filePath: String |
| + exportTo(filePath path): void |