Java

Java

- First appeared in 1995.
- Simple, robust.
- Architecture independent, portable.
- High performance.
- Interpreted, threaded, dynamic.

Day 4
Java Web Programming (cont.)

Lecturer: Msc. Minh Tan Le

Step-by-step

I. Standard Tag Library
II. JavaBeans
III. Dependency inversion
IV. Spring introduction

Standard Tag Library (JSTL)

Standard Tag Library (JSTL)

<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = “...” %>

prefix = “c”

No.	Tag	Equivalence
1	<c:out value>	<%= ... >
2	<c:set var value>	<%! ... %>
3	<c:catch var>	try... catch...
4	<c:if test=“${...}”>	if ...
5	<c:choose> <c:when test=“${...}”> <c:otherwise>	switch(...) { case ...: } (nearly equivalent)
6	<c:import url var>	*Get the content fron URL and print or save to var.*
7	<c:forEach var begin end>	For loop.

prefix = “fmt”

No.	Tag	Equivalence
1	<fmt:formatNumber value>	Number format, such as **String.format**
2	<fmt:parseNumber>	Process string to NUMBER, CURRENCY, or PERCENT.
3	<fmt:formatDate value pattern>	Date, time format, such as **DateFormat**.
4	<fmt:parseDate>	Process string to DATE, TIME, or BOTH.

Learn more: https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm

Demo

Try using JSTL in our previous demo.

We only need models now

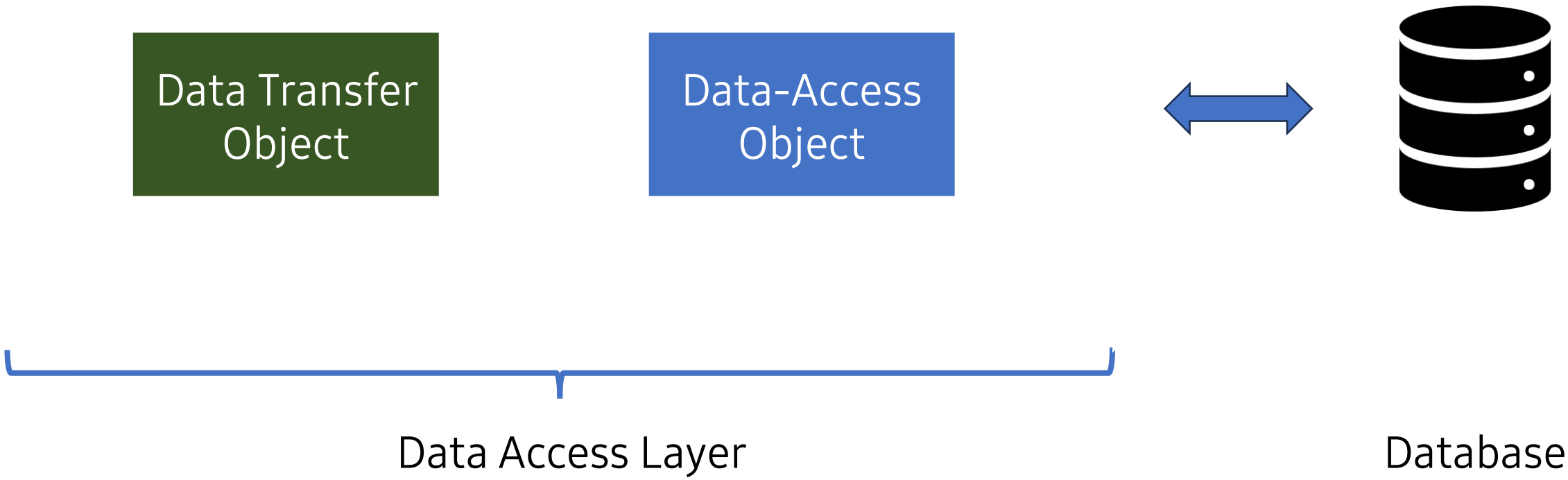
- Model layer contain business logic.
- Model classes have some properties of DTO.
- Here are some models:
  - Student
  - Employee
  - ...

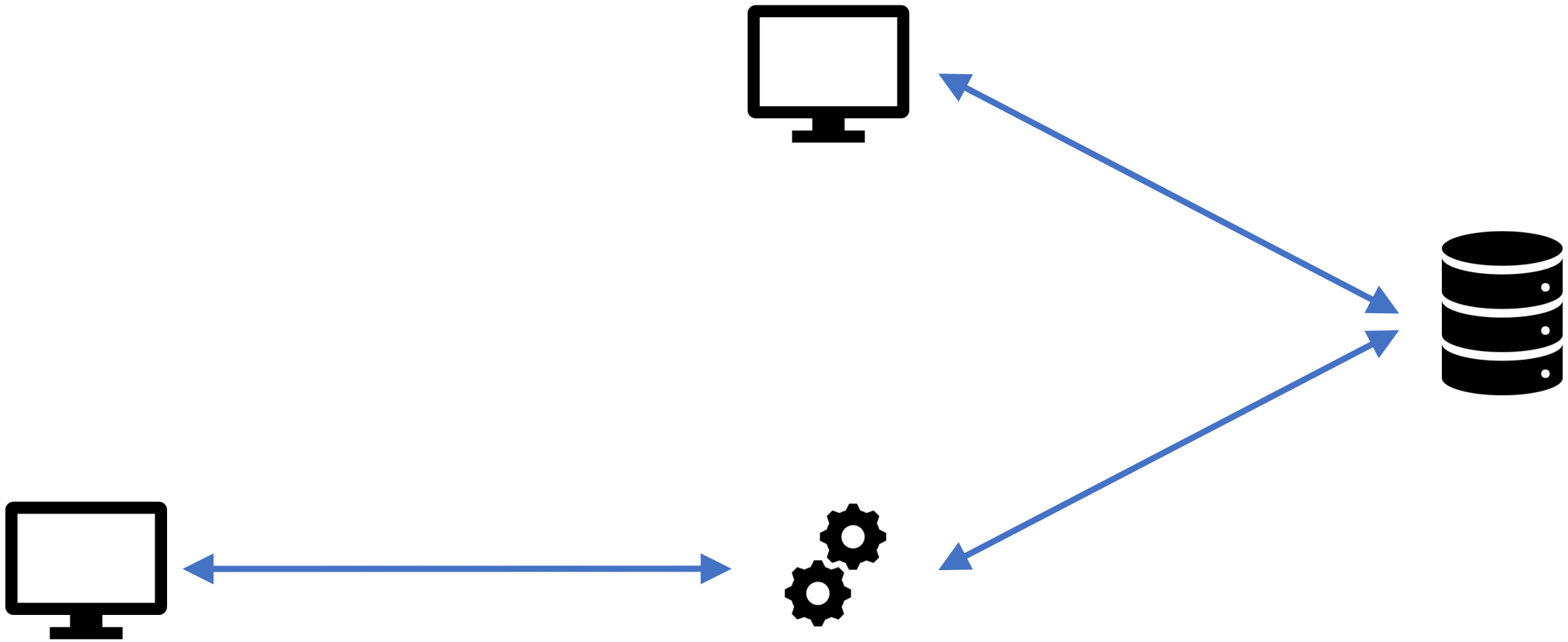
Notes for model classes

- Names must be singular.
- First letter must be uppercase.
- Must be a noun.
- Must follow view’s purpose.
- Constructor must accept DTO and DTO can be casted to it.

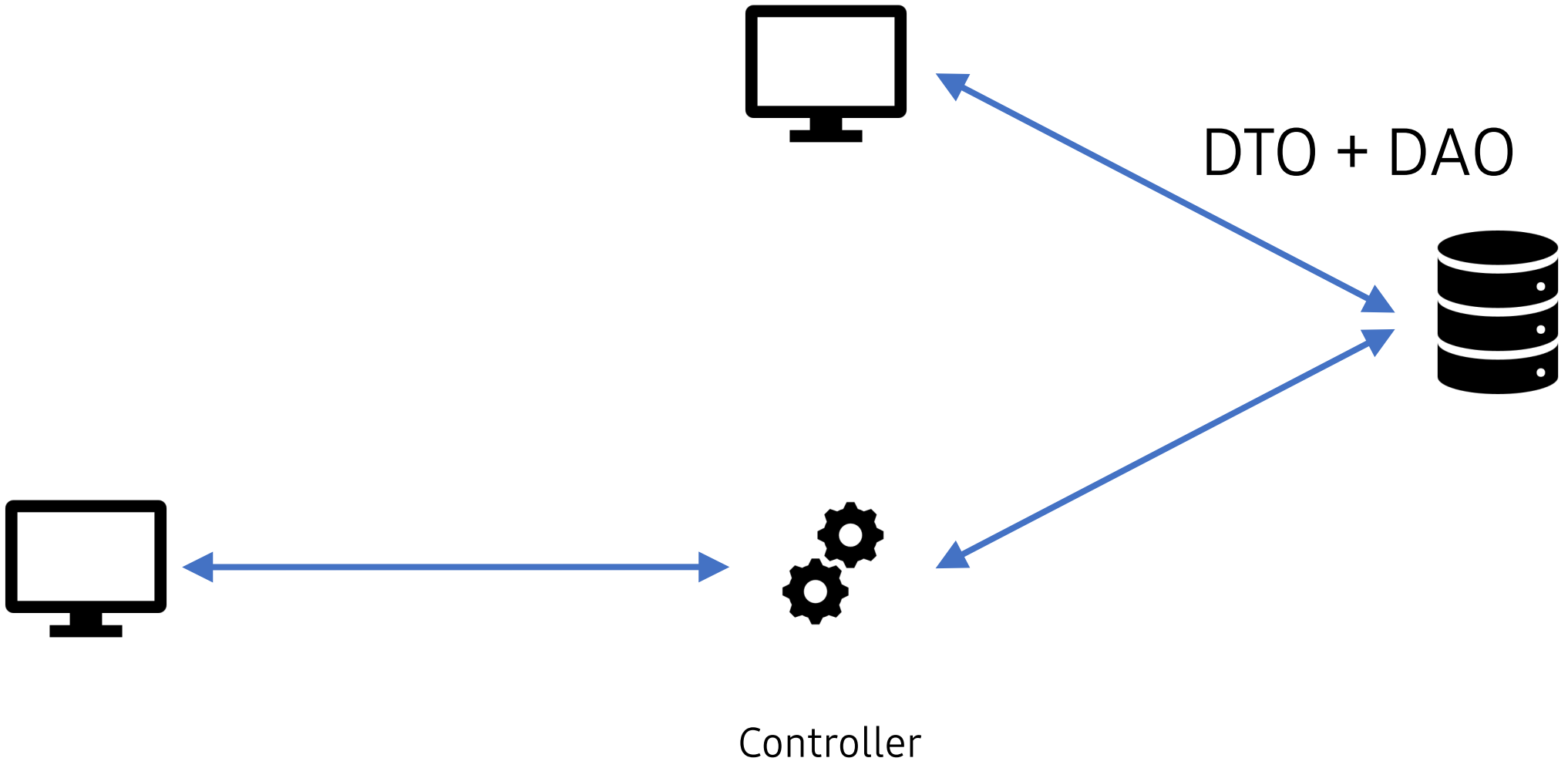
Students may not be able to distinguish model & DB object.

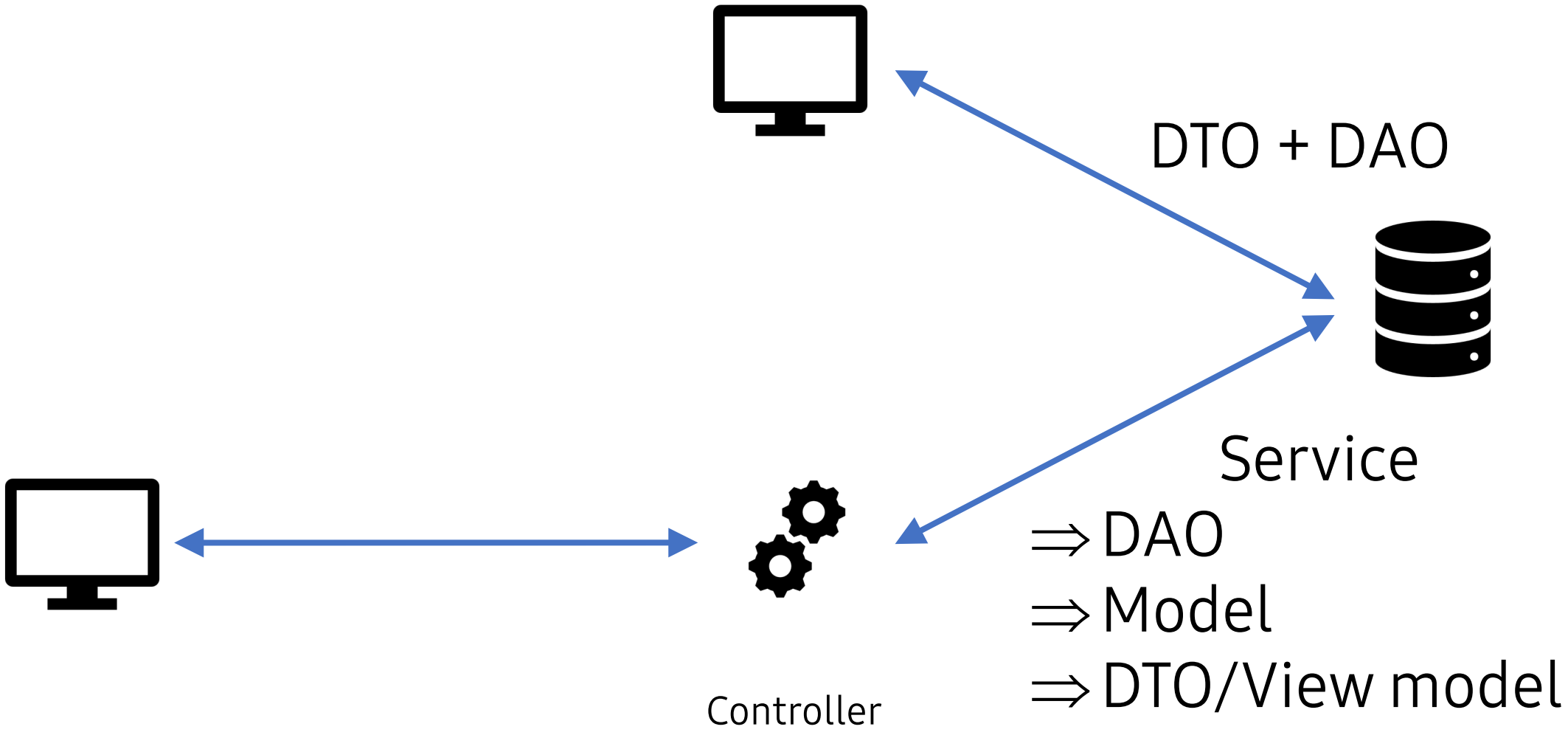
Programming on Windows review



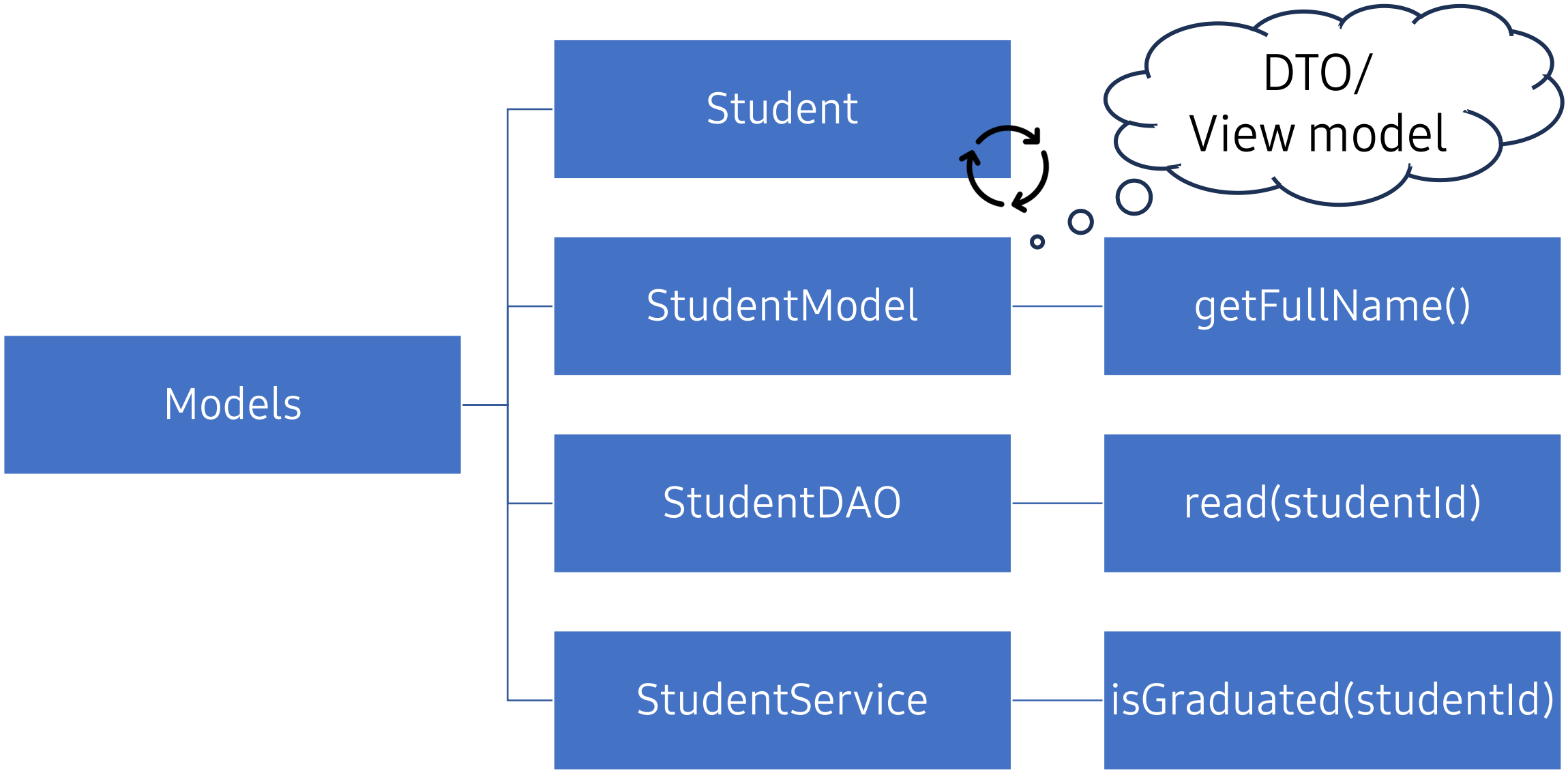


Controller


DTO + DAO
Controller


DTO + DAO
Service
⇒ DAO
⇒ Model
⇒ DTO/View model
Controller

ORM components



Bean is a standard to define a class

- Properties: get...(), set...(...)
- No-argument constructor, can have methods
- Serializable: Every field should be fully presentable as string.
  - 50 (integer) is serializable.
  - Pointer/cursor is not serializable.
- Mostly used for dependency inversion.

Case study

https://github.com/ikismail/ShoppingCart

What is “dependency” In programming?

What is “dependency” In programming?

```
public class Student {
    public String id;
    public boolean graduated;
    public Student(String id) {
        this.id = id
    }
}
```

Let’s convert it to a bean!

```
public class Student {
    public String id;
    public boolean graduated;
    public Student(String id) {
        this.id = id
    }
}
```

```
public class Student implements Serializable {
    private String id;
    private boolean graduated;

    public Student() {}

    public String getId() {
        return this.id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public boolean isGraduated() {
        return this.graduated;
    }

    public void setGraduated(boolean graduated) {
        this.graduated = graduated;
    }
}
```

Dependency inversion

- Sometimes, we want to get an object with default properties.
- For instance:
  - Services: You shouldn’t create multiple service object.
  - Configuration: One object for storing is enough.

Object

Consumer

Injected object

Injector (container)

Consumer

Definitions

- D.Inversion is transferring the control of objects to a helper or framework.
- D.Injection is the pattern to implement D.Inversion.
  - Setter-based.
  - Constructor-based.
  - Method-based.

Demo: The beginning

1. Create a Java console project.
2. Create a Controller class.
3. Create a Student model.
4. Create an interface for Services.
5. Create a Service as JavaBean class.

spring®

vmware®

by Broadcom

2-day homework

Create a simple website that collect student data through a HTML form:

- Assuming you don’t need any login or database.
- After finishing the form, user will be provided a result id.
- Navigating to /groupn/id will show form submission data.