Development Process

Step-By-Step

© luv2code LLC

Development Process

Step-By-Step

1. In superclass, specify

Development Process

Step-By-Step

1. In superclass, specify
    1. Inheritance strategy

Development Process

Step-By-Step

1. In superclass, specify
    1. Inheritance strategy
    2. Discriminator column name

© luv2code LLC

Development Process

Step-By-Step

1. In superclass, specify
   1. Inheritance strategy
   2. Discriminator column name

2. In subclass, specify discriminator value

© luv2code LLC

Development Process

Step-By-Step

1. In superclass, specify
   1. Inheritance strategy
   2. Discriminator column name
2. In subclass, specify discriminator value
3. Develop main application

© luv2code LLC

Annotation for Inheritance

Annotation for Inheritance

Annotation	Description

© luv2code LLC

Annotation for Inheritance

Annotation	Description
`@Inheritance`	**Specify the inheritance strategy. Possible values: SINGLE_TABLE, TABLE_PER_CLASS, JOINED** **Defaults to SINGLE_TABLE.**

Annotation for Inheritance

Annotation	Description
`@Inheritance`	**Specify the inheritance strategy. Possible values: SINGLE_TABLE, TABLE_PER_CLASS, JOINED** **Defaults to SINGLE_TABLE.**
`@DiscriminatorColumn`	**Name of the column that holds the discriminator values. Defaults to DTYPE.**

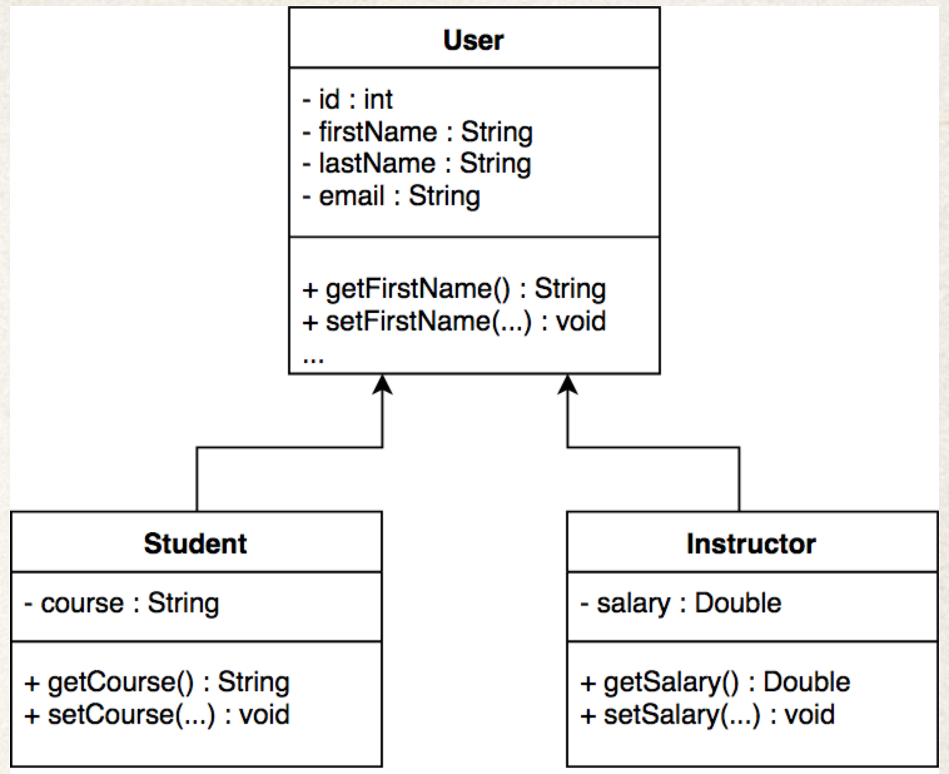
© luv2code LLC

Annotation for Inheritance

Annotation	Description
`@Inheritance`	**Specify the inheritance strategy. Possible values: SINGLE_TABLE, TABLE_PER_CLASS, JOINED** **Defaults to SINGLE_TABLE.**
`@DiscriminatorColumn`	**Name of the column that holds the discriminator values. Defaults to DTYPE.**
`@DiscriminatorValue`	**A unique value that describes a given subclass. Defaults to class name.**

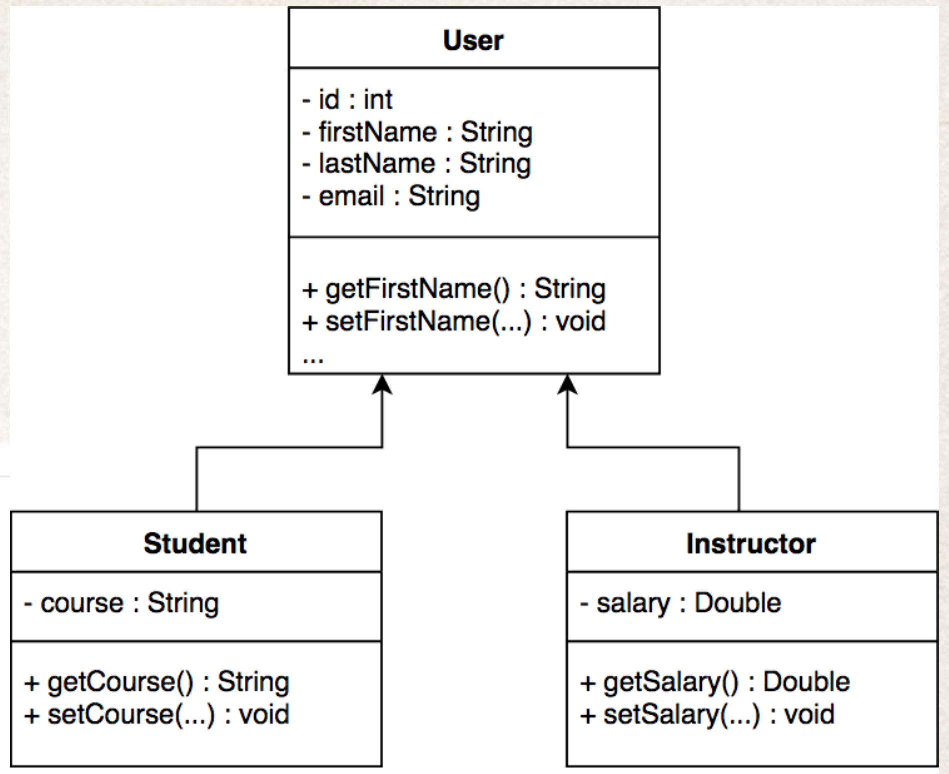
© luv2code LLC

Step 1: Superclass - Inheritance strategy



© luv2code LLC

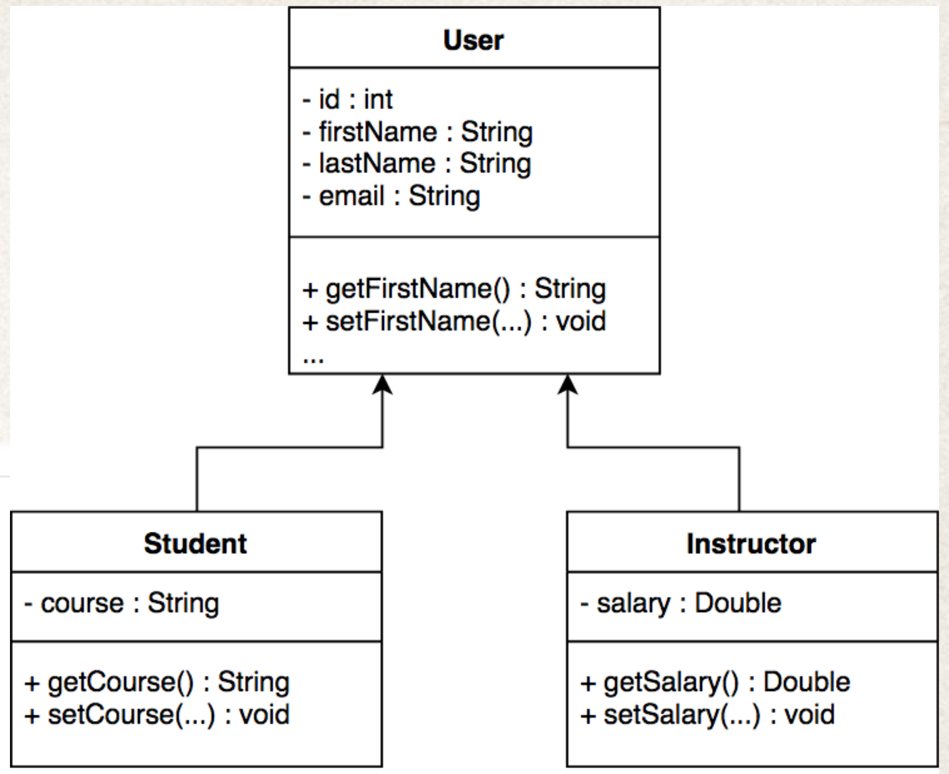
Step 1: Superclass - Inheritance strategy



```
@Entity
@Table(name="user")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
public class User {


…


}
```

© luv2code LLC

Step 1: Superclass - Inheritance strategy



User

- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student

- course : String

+ getCourse() : String
+ setCourse(...) : void

Instructor

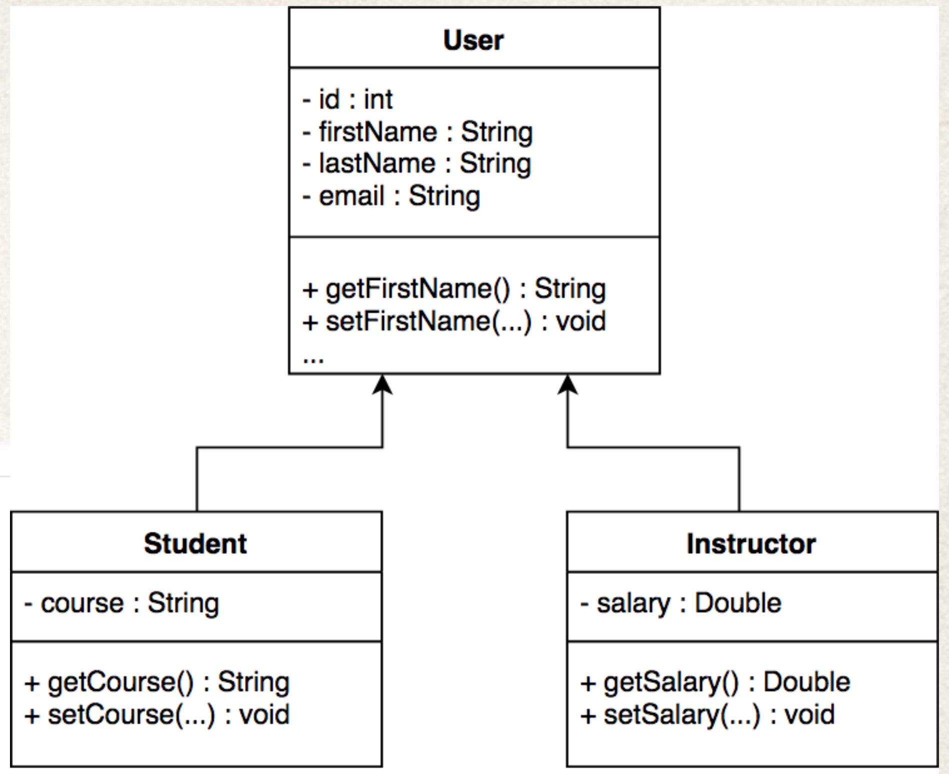
- salary : Double

+ getSalary() : Double
+ setSalary(...) : void

```
@Entity
@Table(name="user")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
public class User {

…

}
```

Map all fields in inheritance tree to a single table

Step 1: Superclass - Inheritance strategy



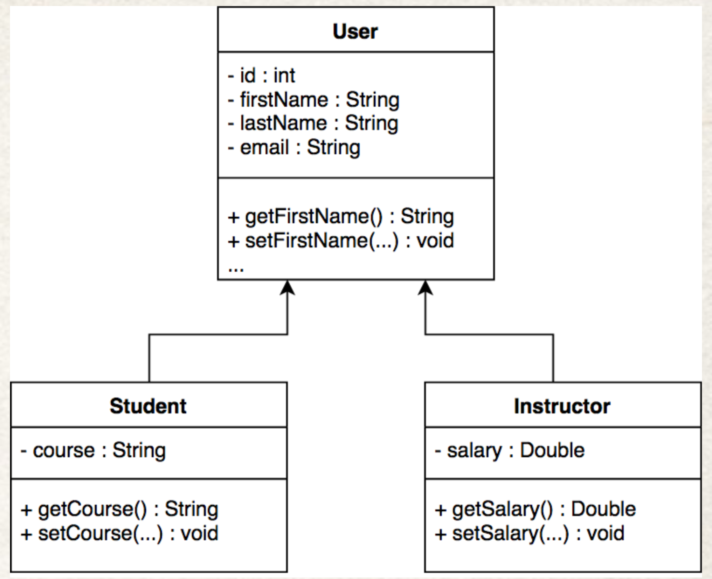
```
@Entity
@Table(name="user")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
public class User {


...


}
```

Map all fields in inheritance tree to a single table

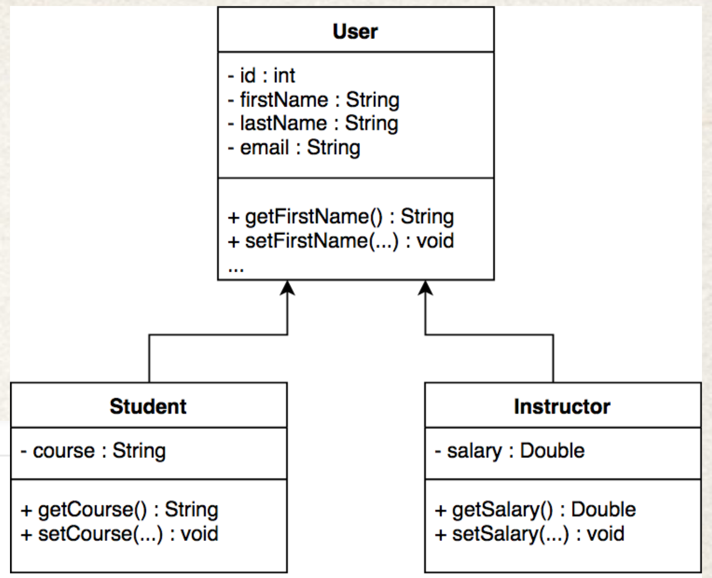
@Inheritance is optional
Defaults to SINGLE_TABLE

© luv2code LLC

Step 1: Superclass - Discriminator column



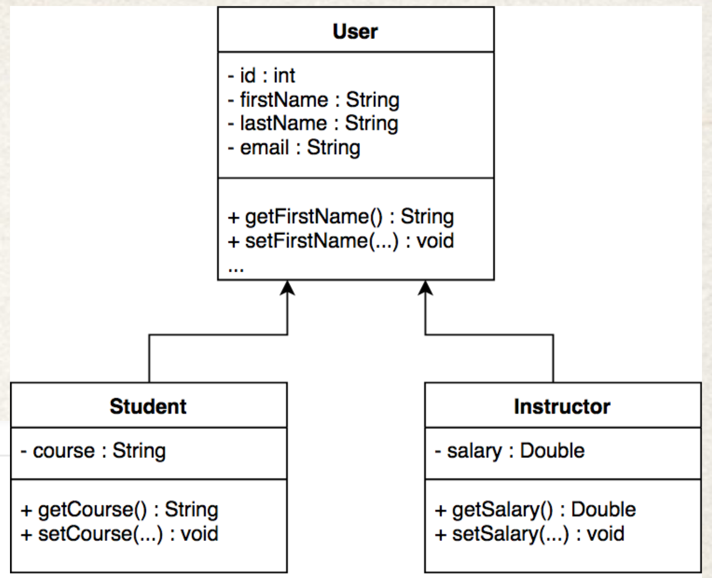
Step 1: Superclass - Discriminator column



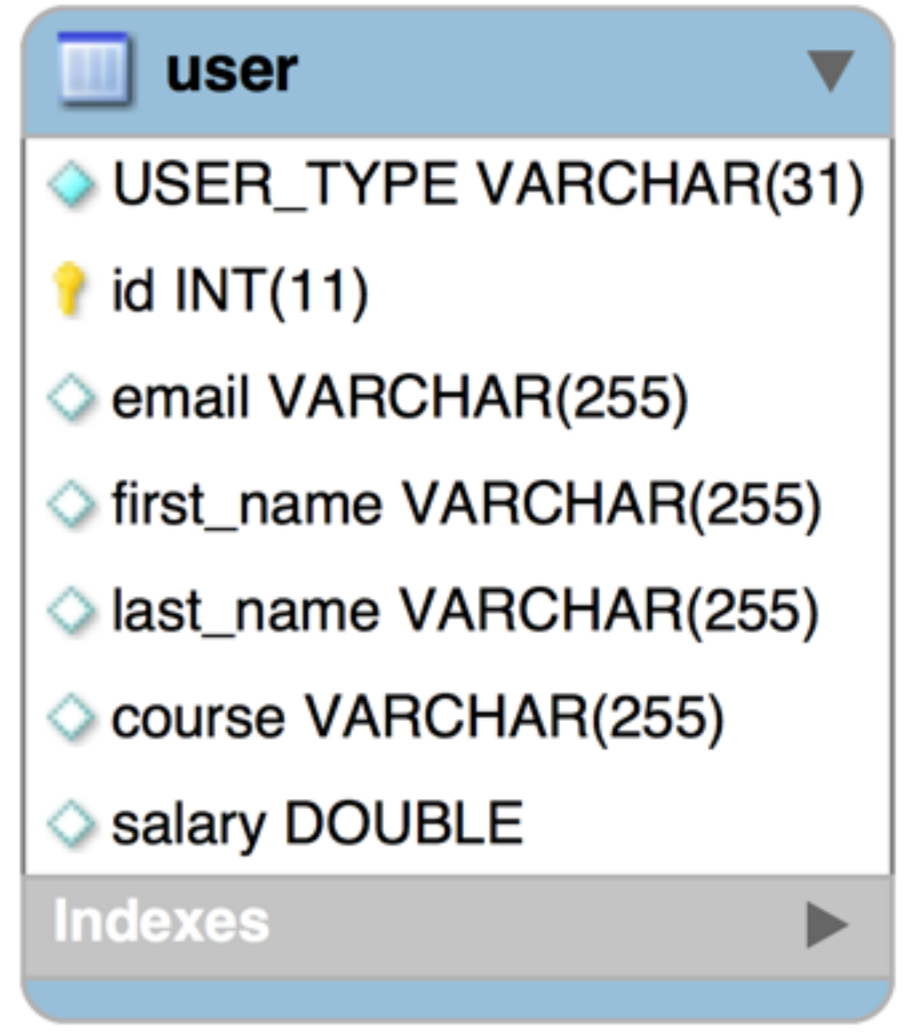
```
@Entity
@Table(name="user")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name="USER_TYPE", discriminatorType=DiscriminatorType.STRING)
public class User {

…

}
```

© luv2code LLC

Step 1: Superclass - Discriminator column



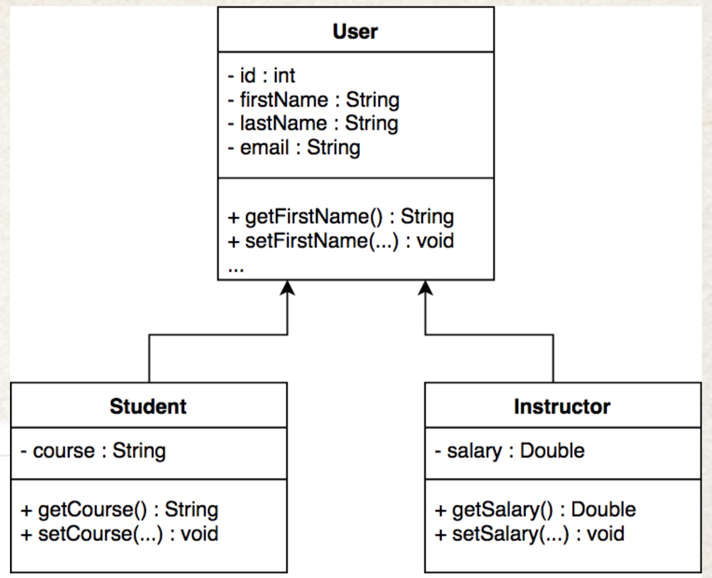
```
@Entity
@Table(name="user")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name="USER_TYPE", discriminatorType=DiscriminatorType.STRING)
public class User {

...

}
```


user

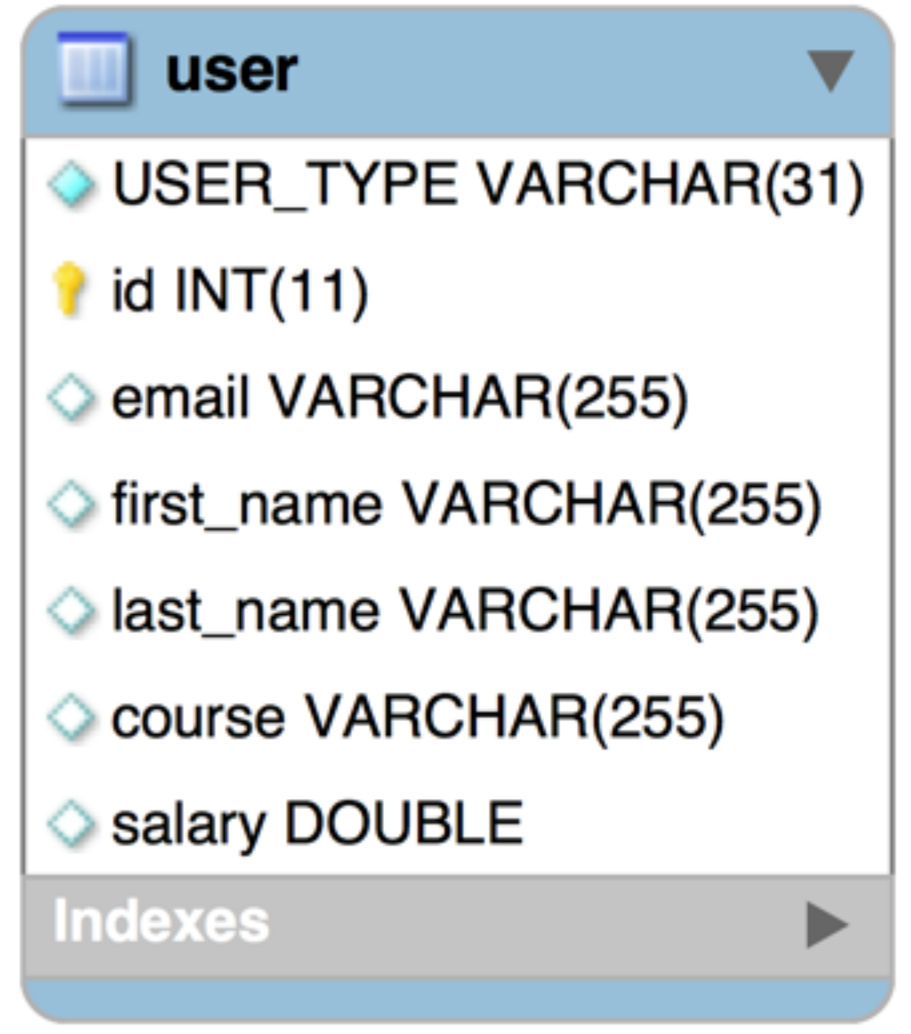
USER_TYPE VARCHAR(31)
id INT(11)
email VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
course VARCHAR(255)
salary DOUBLE

© luv2code LLC

Step 1: Superclass - Discriminator column



```
@Entity
@Table(name="user")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name="USER_TYPE", discriminatorType=DiscriminatorType.STRING)
public class User {

…

}
```



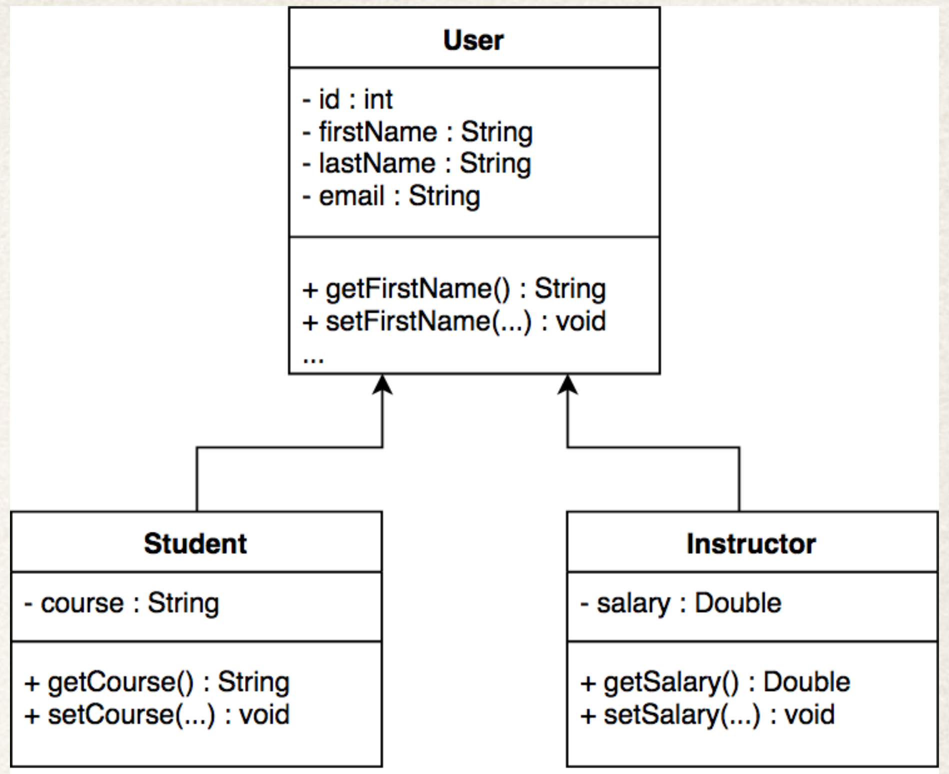
user

- USER_TYPE VARCHAR(31)
- id INT(11)
- email VARCHAR(255)
- first_name VARCHAR(255)
- last_name VARCHAR(255)
- course VARCHAR(255)
- salary DOUBLE

@DiscriminatorColumn is optional
Defaults to DTYPE

© luv2code LLC

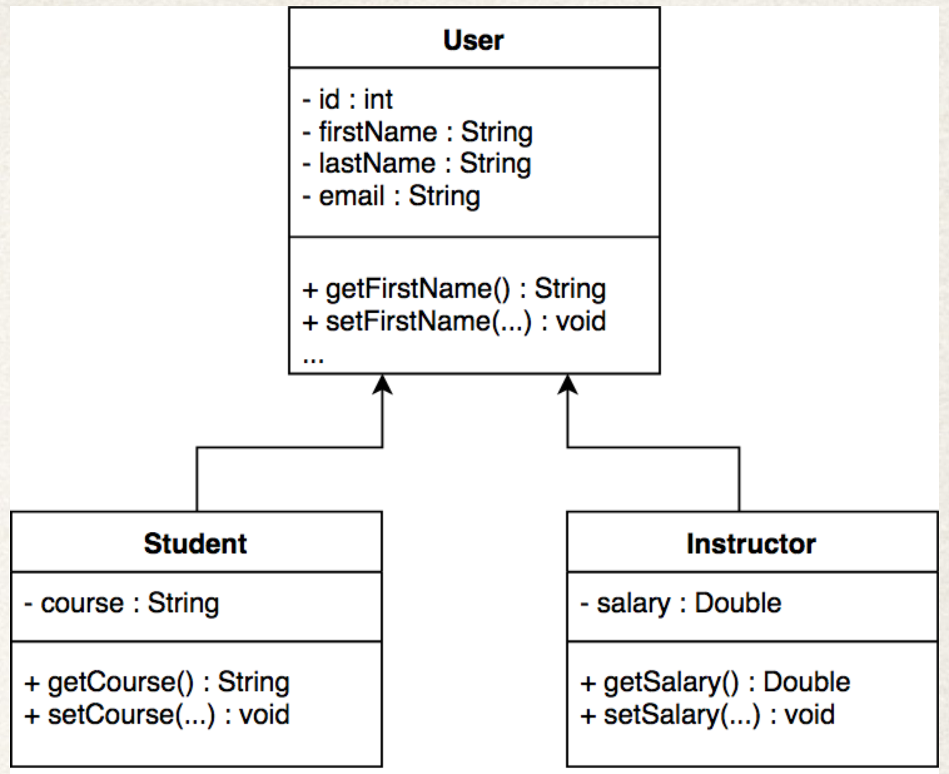
Step 2: Subclass - Discriminator value



© luv2code LLC

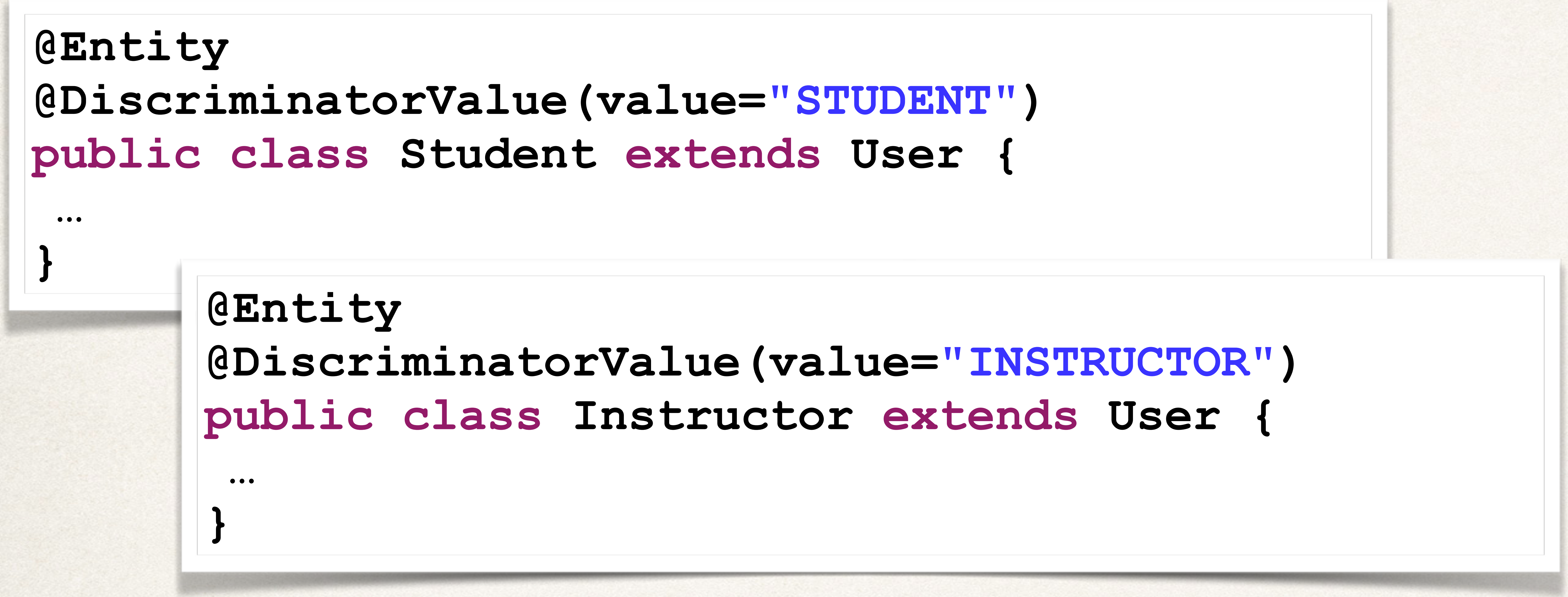
Step 2: Subclass - Discriminator value

```
@Entity
@DiscriminatorValue(value="STUDENT")
public class Student extends User {
 ...
}
```



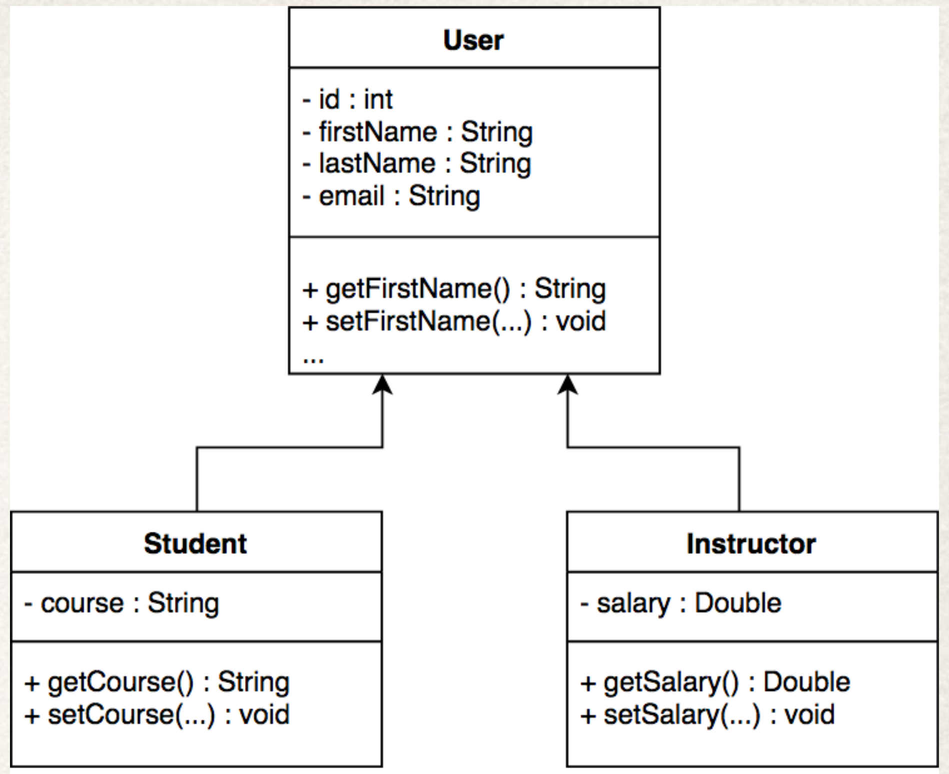
© luv2code LLC

Step 2: Subclass - Discriminator value



```
@Entity
@DiscriminatorValue(value="STUDENT")
public class Student extends User {
 …
}
```

```
@Entity
@DiscriminatorValue(value="INSTRUCTOR")
public class Instructor extends User {
 …
}
```



User
- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student
- course : String

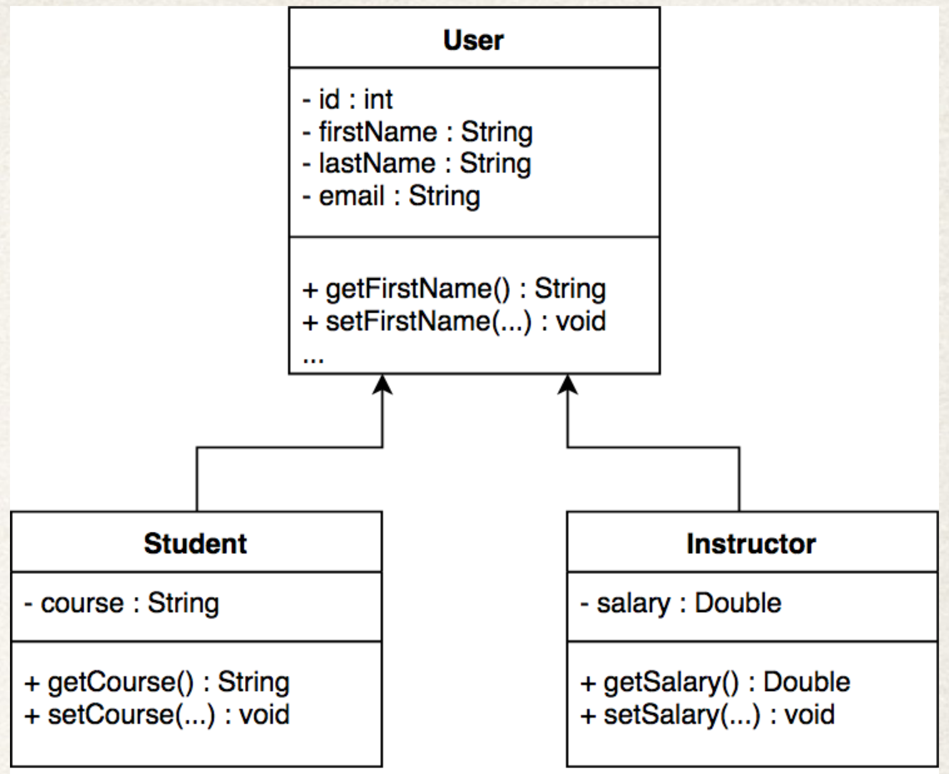
+ getCourse() : String
+ setCourse(...) : void

Instructor
- salary : Double

+ getSalary() : Double
+ setSalary(...) : void

© luv2code LLC

Step 2: Subclass - Discriminator value



```
@Entity
@DiscriminatorValue(value="STUDENT")
public class Student extends User {
 …
}
```

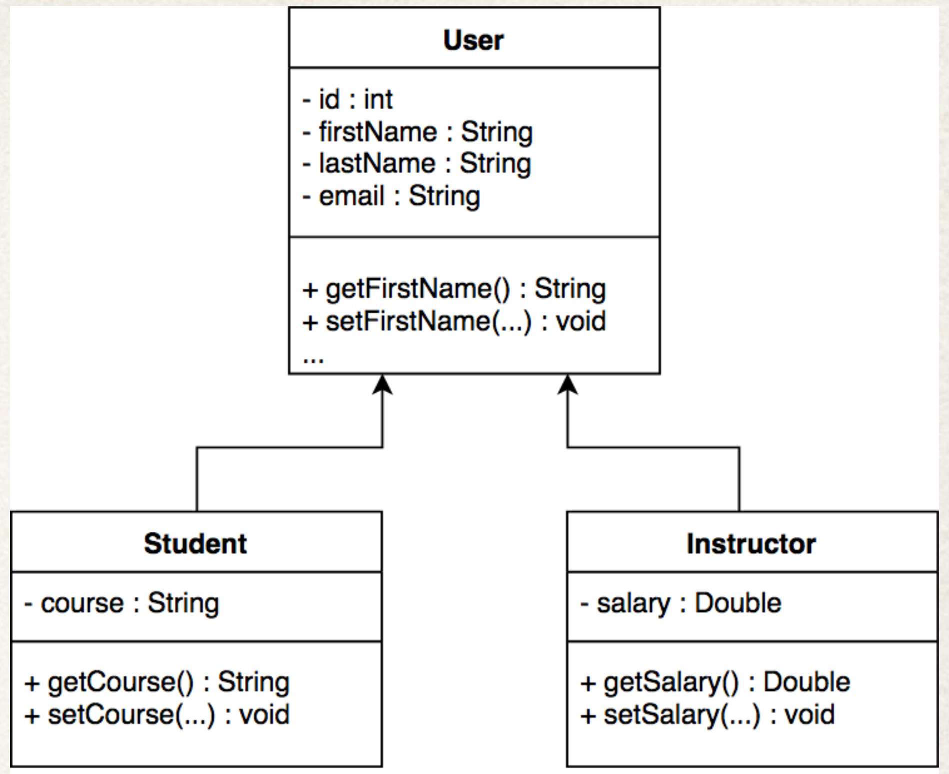
```
@Entity
@DiscriminatorValue(value="INSTRUCTOR")
public class Instructor extends User {
 …
}
```

Discriminator value specifies the class/type of data.

The value is stored in the discriminator column.

© luv2code LLC

Step 2: Subclass - Discriminator value



```
@Entity
@DiscriminatorValue(value="STUDENT")
public class Student extends User {
 …
}
```

```
@Entity
@DiscriminatorValue(value="INSTRUCTOR")
public class Instructor extends User {
 …
}
```

Discriminator value specifies the class/type of data.

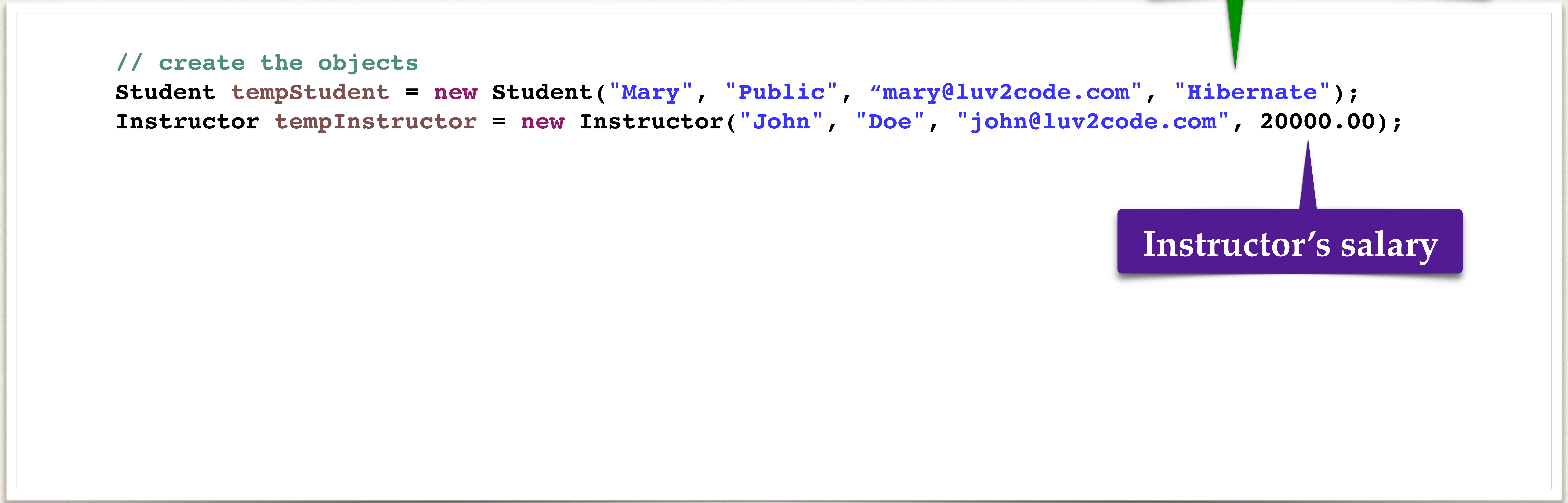
The value is stored in the discriminator column.

@DiscriminatorValue is optional
Defaults to class name

© luv2code LLC

Step 3: Develop the main application

Step 3: Develop the main application

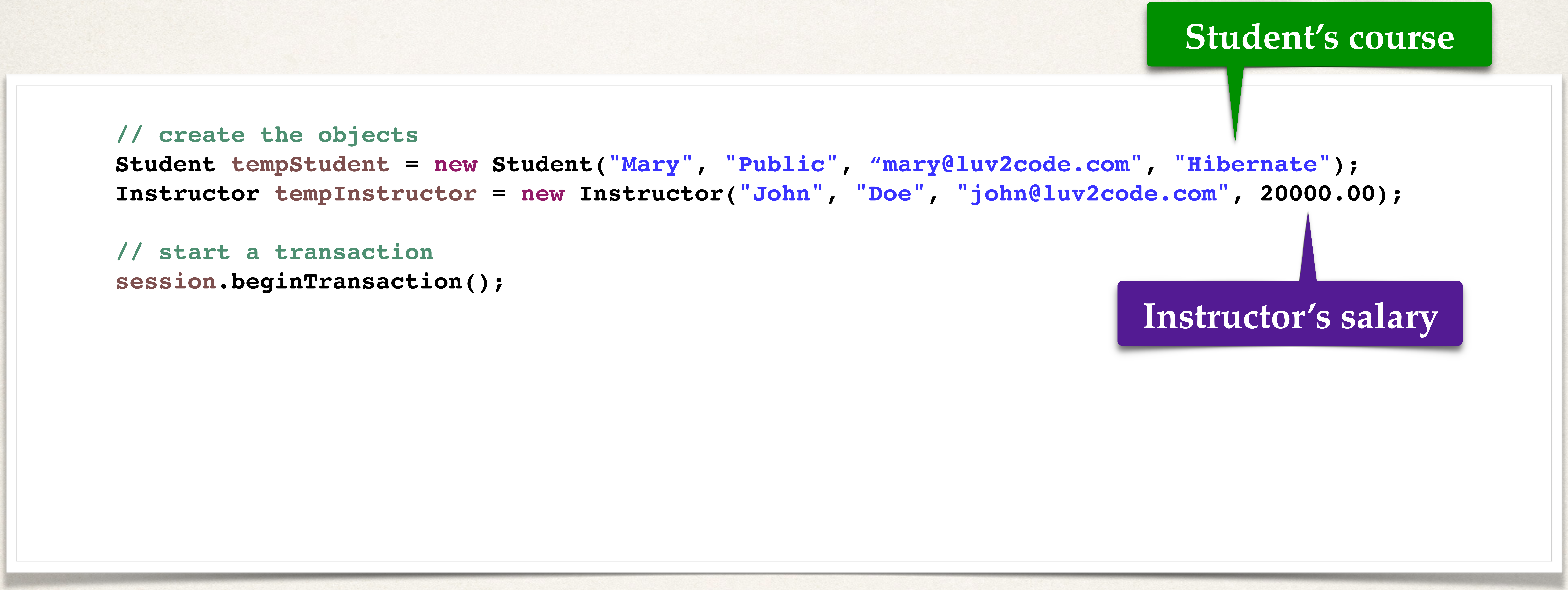


```
// create the objects
Student tempStudent = new Student("Mary", "Public", “mary@luv2code.com", "Hibernate");
Instructor tempInstructor = new Instructor("John", "Doe", "john@luv2code.com", 20000.00);
```

Student’s course

Instructor’s salary

Step 3: Develop the main application

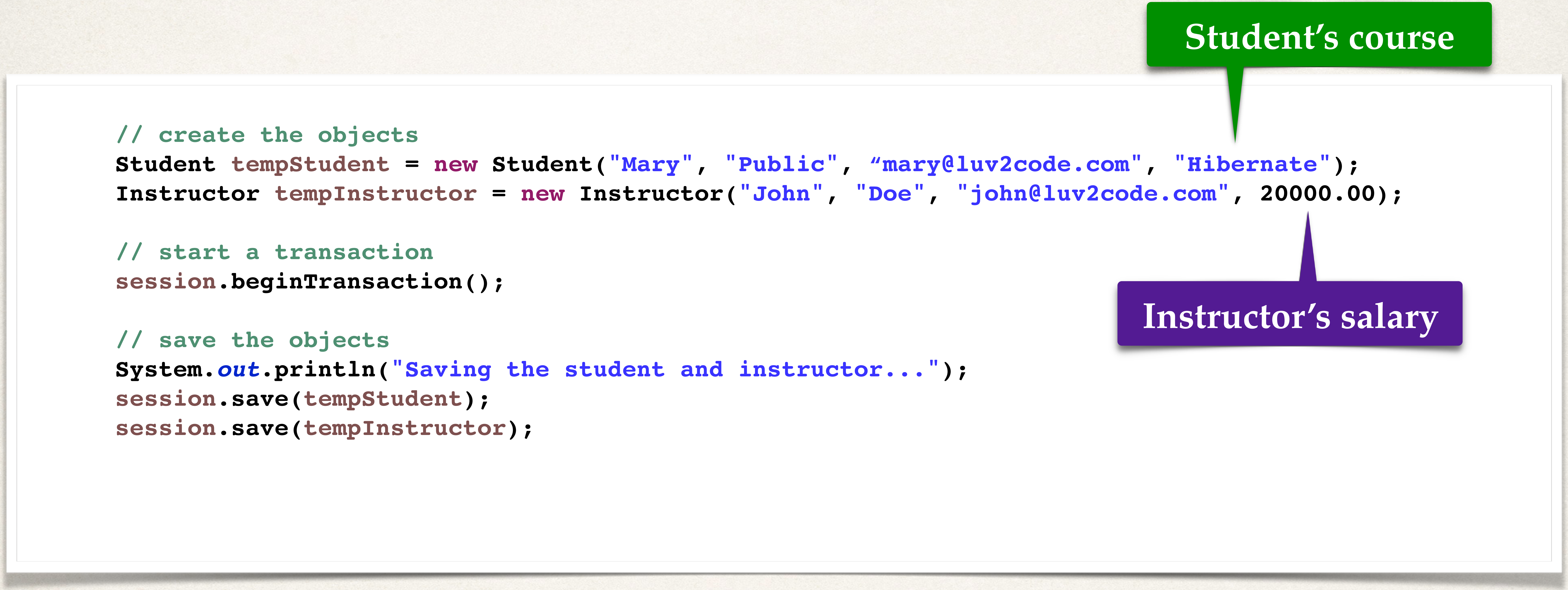


```
// create the objects
Student tempStudent = new Student("Mary", "Public", “mary@luv2code.com", "Hibernate");
Instructor tempInstructor = new Instructor("John", "Doe", "john@luv2code.com", 20000.00);

// start a transaction
session.beginTransaction();
```

Student’s course

Instructor’s salary

Step 3: Develop the main application

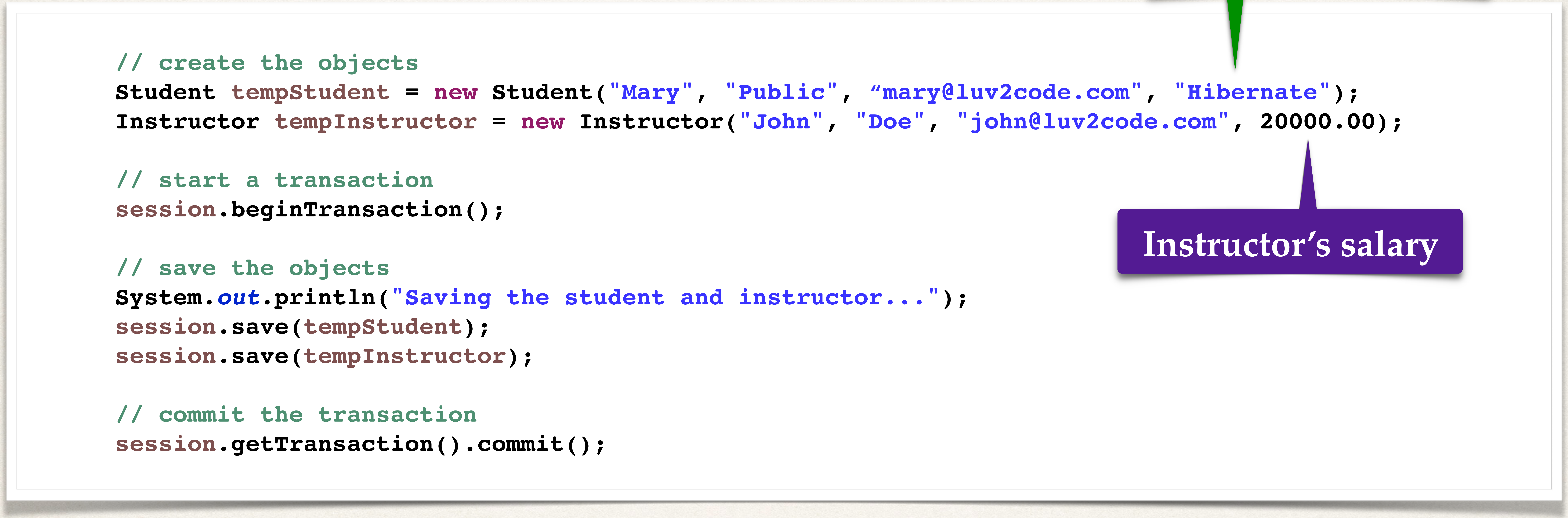


```
// create the objects
Student tempStudent = new Student("Mary", "Public", “mary@luv2code.com", "Hibernate");
Instructor tempInstructor = new Instructor("John", "Doe", "john@luv2code.com", 20000.00);

// start a transaction
session.beginTransaction();

// save the objects
System.out.println("Saving the student and instructor...");
session.save(tempStudent);
session.save(tempInstructor);
```

Student’s course

Instructor’s salary

Step 3: Develop the main application



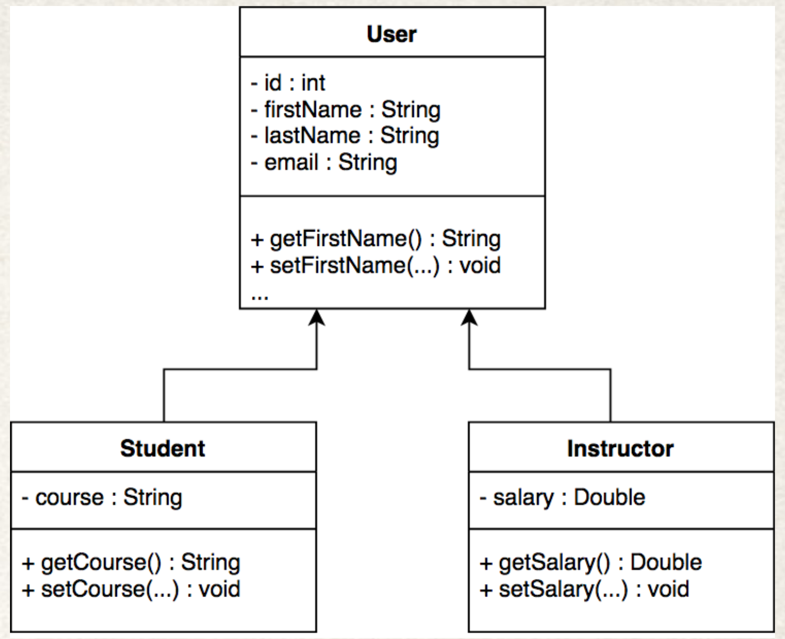
```
// create the objects
Student tempStudent = new Student("Mary", "Public", “mary@luv2code.com", "Hibernate");
Instructor tempInstructor = new Instructor("John", "Doe", "john@luv2code.com", 20000.00);

// start a transaction
session.beginTransaction();

// save the objects
System.out.println("Saving the student and instructor...");
session.save(tempStudent);
session.save(tempInstructor);

// commit the transaction
session.getTransaction().commit();
```

Student’s course

Instructor’s salary

© luv2code LLC

Run the App



User

- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student

- course : String

+ getCourse() : String
+ setCourse(...) : void

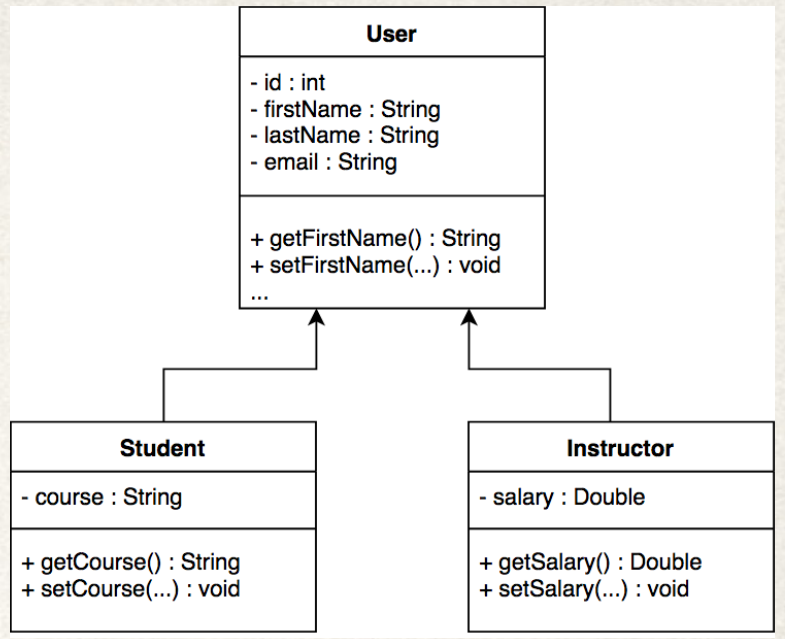
Instructor

- salary : Double

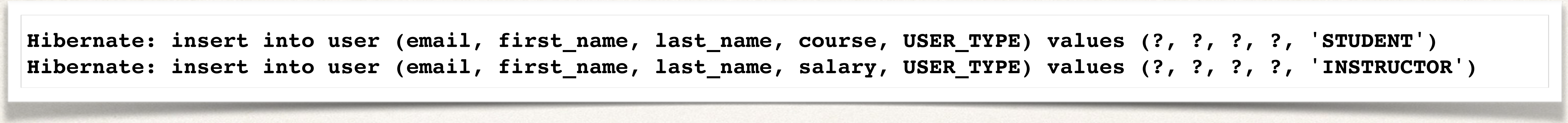
+ getSalary() : Double
+ setSalary(...) : void

© luv2code LLC

Run the App



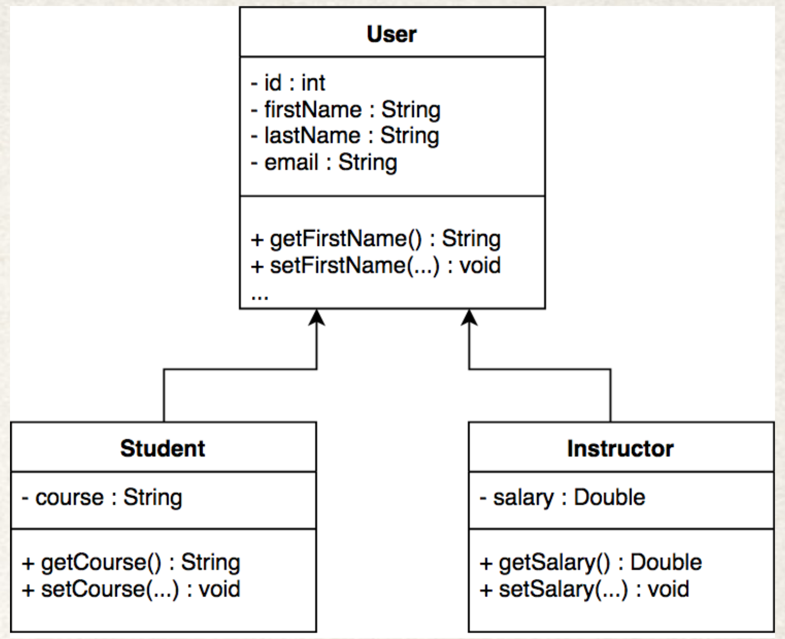
Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

© luv2code LLC

Run the App



User

- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student

- course : String

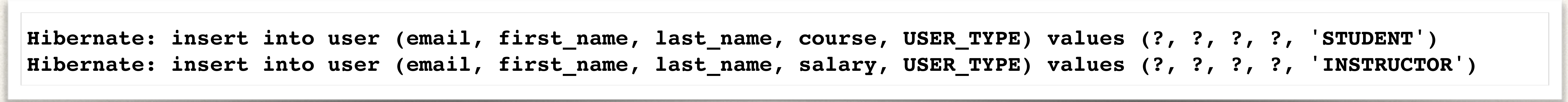
+ getCourse() : String
+ setCourse(...) : void

Instructor

- salary : Double

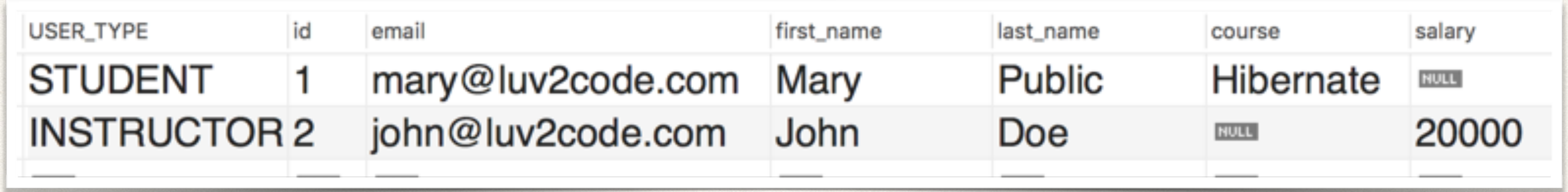
+ getSalary() : Double
+ setSalary(...) : void

Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

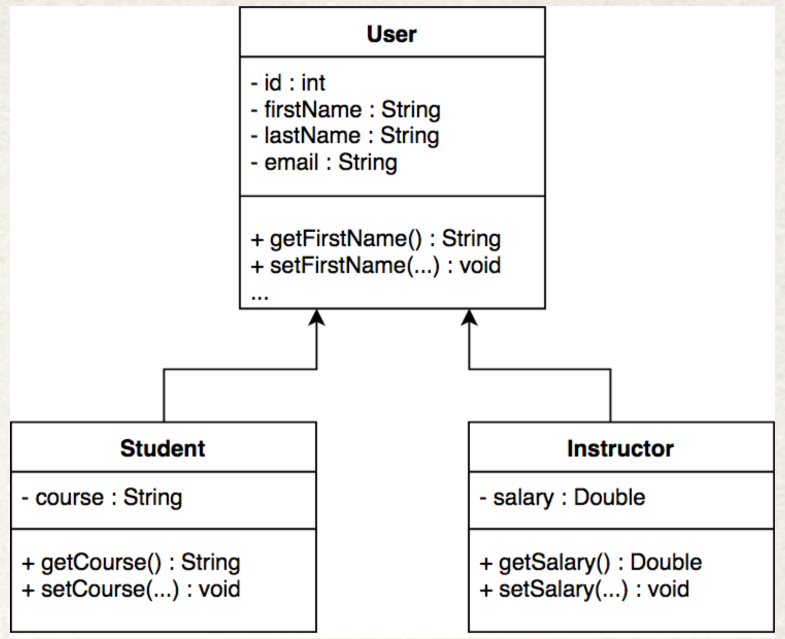
Table: user



USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

© luv2code LLC

Run the App



User

- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student

- course : String

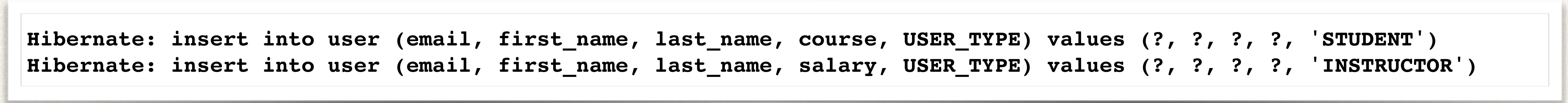
+ getCourse() : String
+ setCourse(...) : void

Instructor

- salary : Double

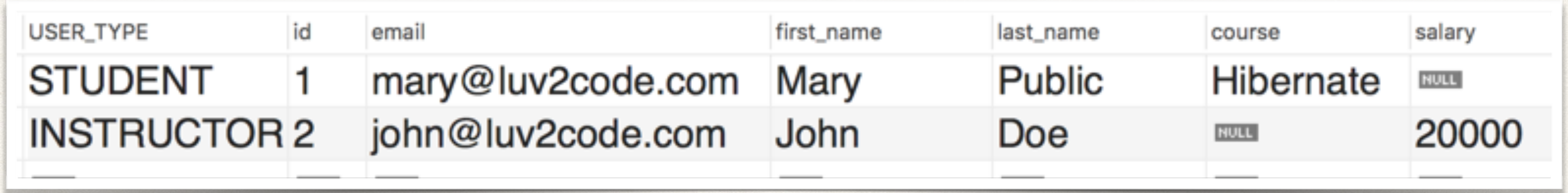
+ getSalary() : Double
+ setSalary(...) : void

Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

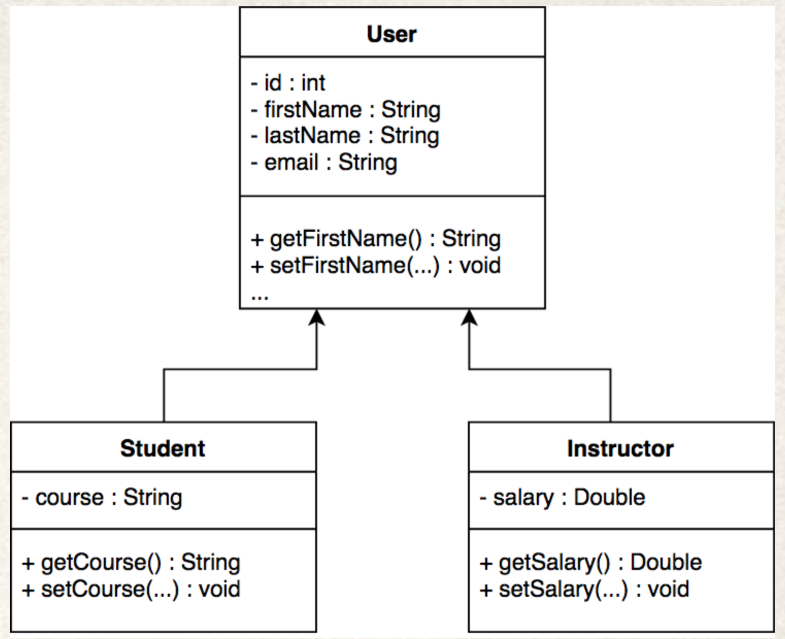
Table: user



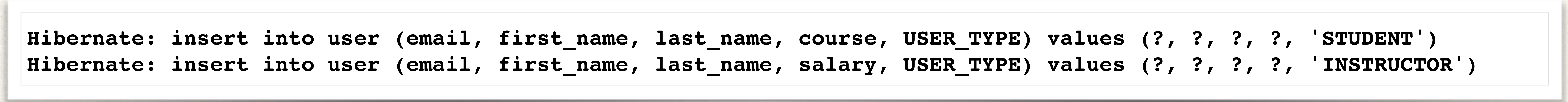
USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

Table has columns for all fields in inheritance tree

Run the App

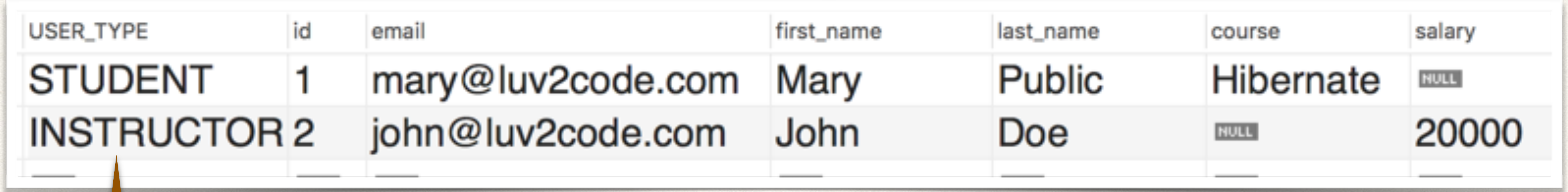


Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

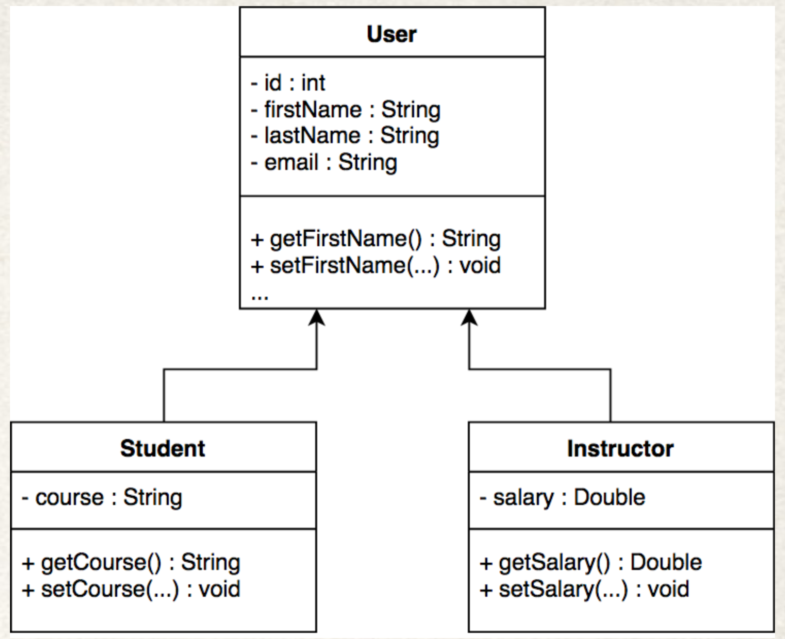
Table: user

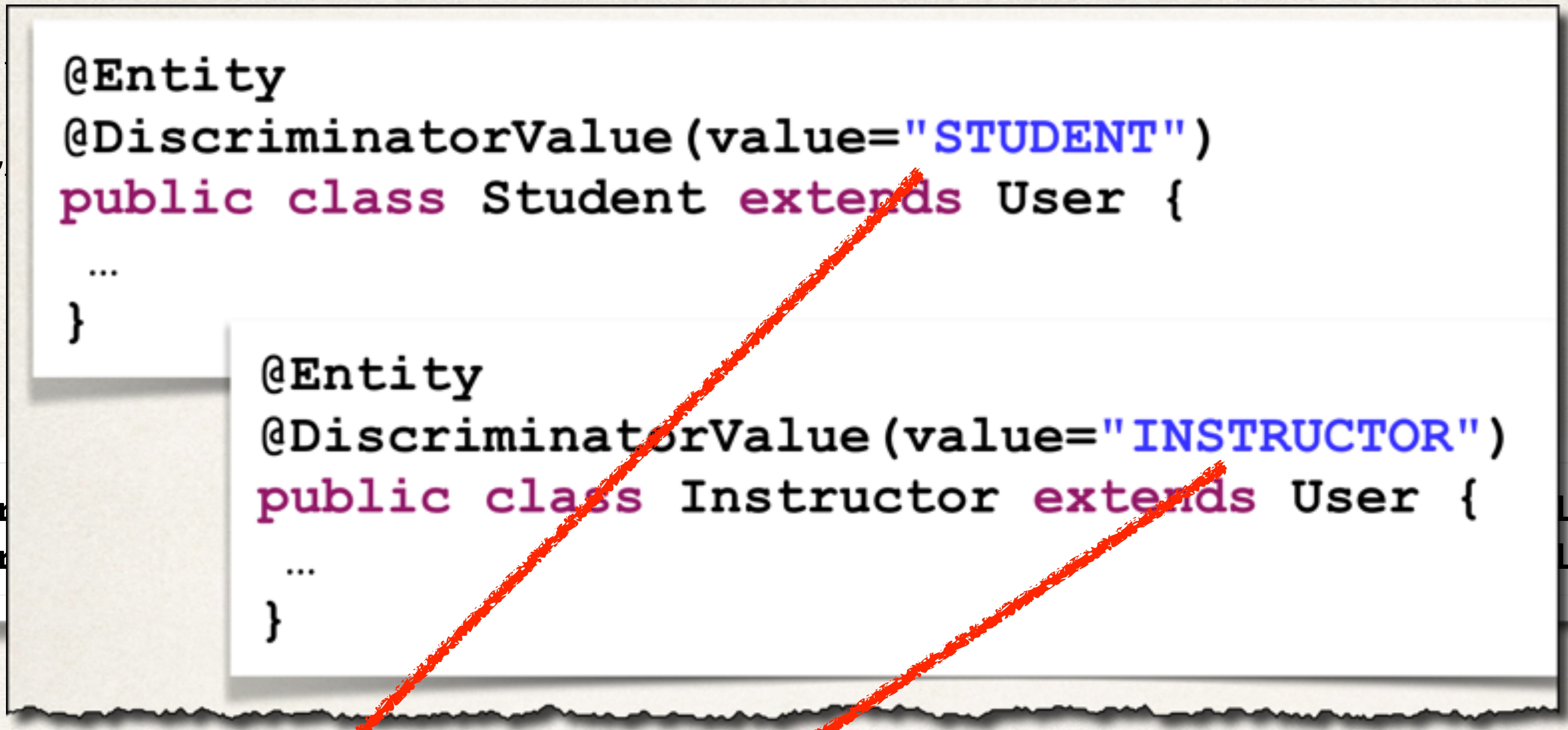


USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

Discriminator column
The class/type of data

Run t





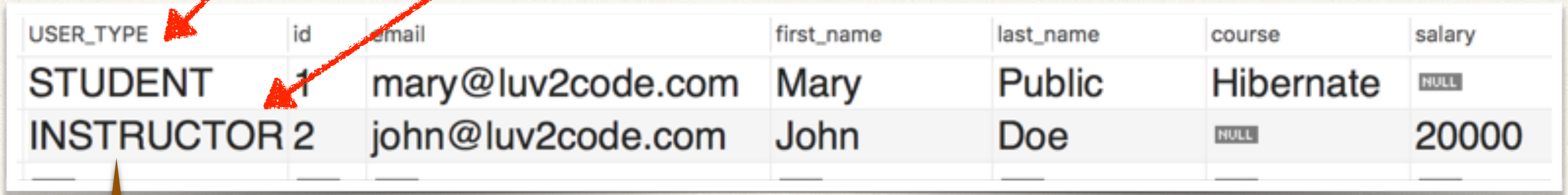
```
@Entity
@DiscriminatorValue(value="STUDENT")
public class Student extends User {
 …
}
```

```
@Entity
@DiscriminatorValue(value="INSTRUCTOR")
public class Instructor extends User {
 …
}
```

Console

```
Hibernate: inser                                              ues (?, ?, ?, ?, 'STUDENT')
Hibernate: inser                                              ues (?, ?, ?, ?, 'INSTRUCTOR')
```

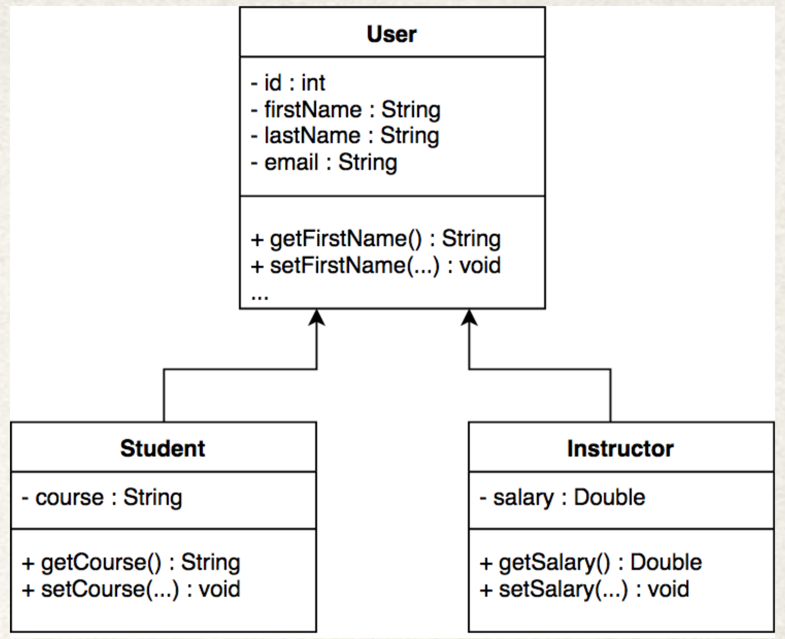
Table: user



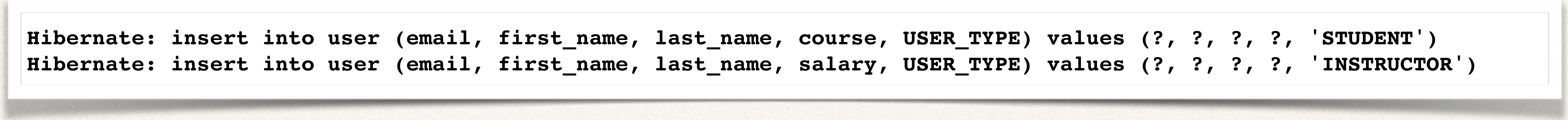
USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

Discriminator column
The class/type of data

Run the App

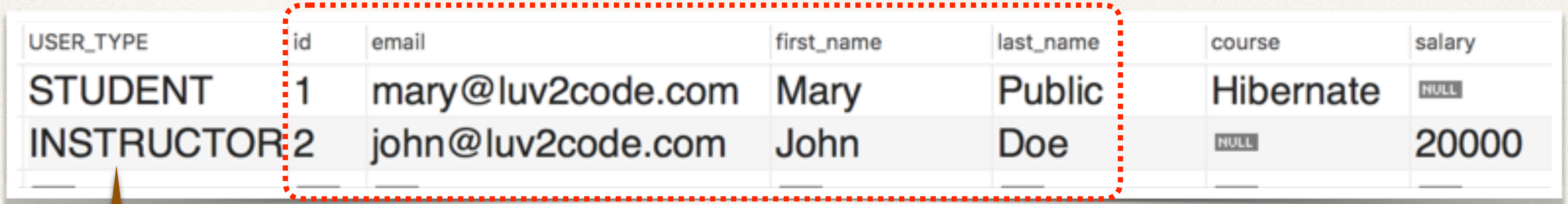


Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

Table: user

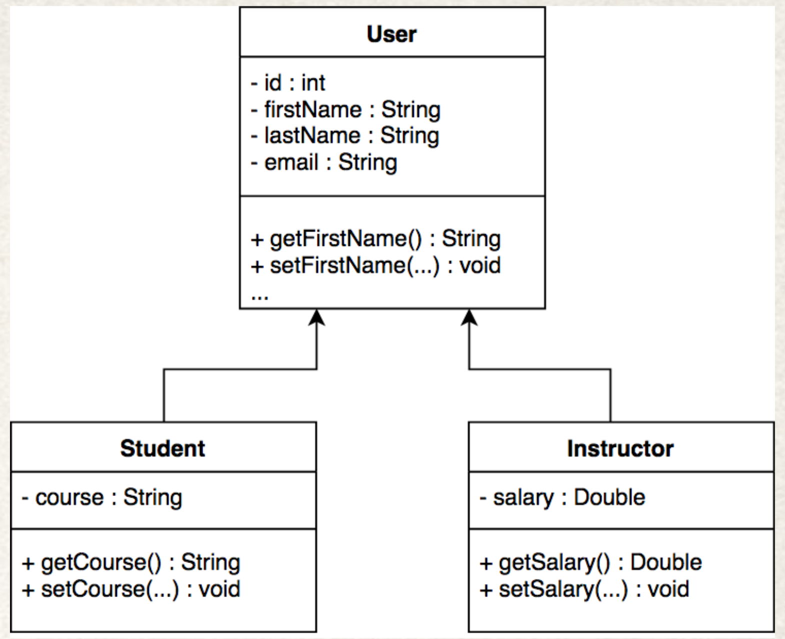


USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

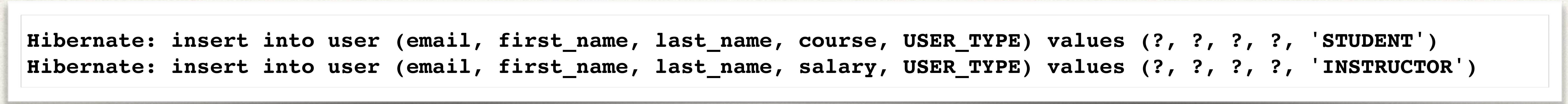
Discriminator column
The class/type of data

User

Run the App

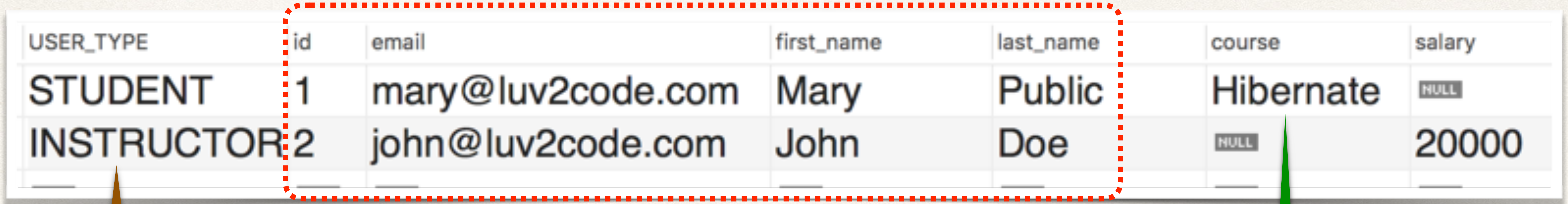


Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

Table: user



USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

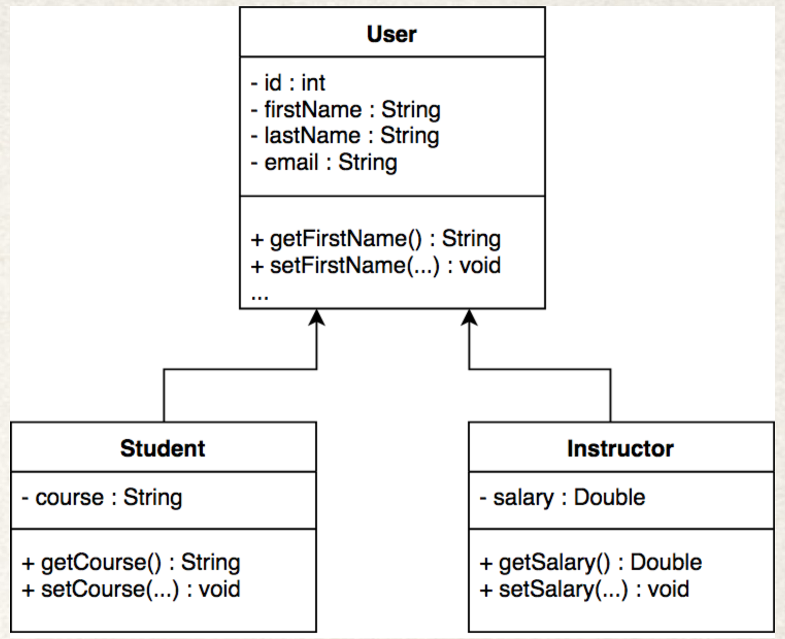
Discriminator column
The class/type of data

User

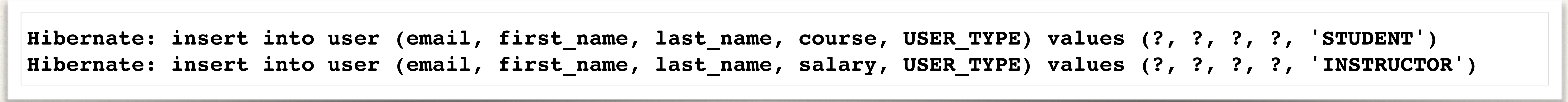
Student

© luv2code LLC

Run the App

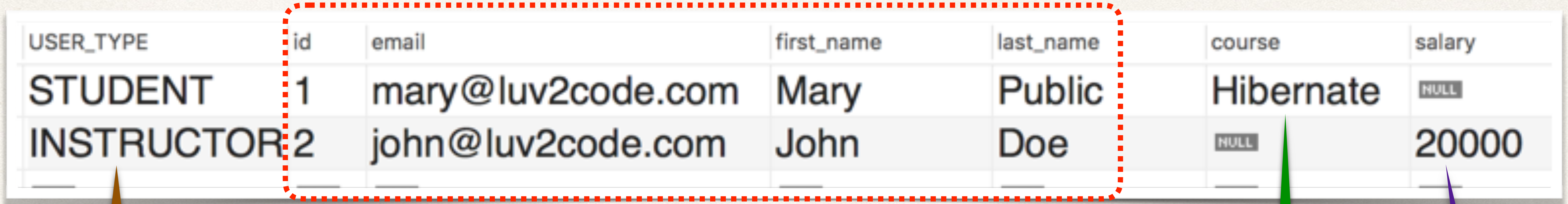


Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

Table: user



USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

Discriminator column
The class/type of data

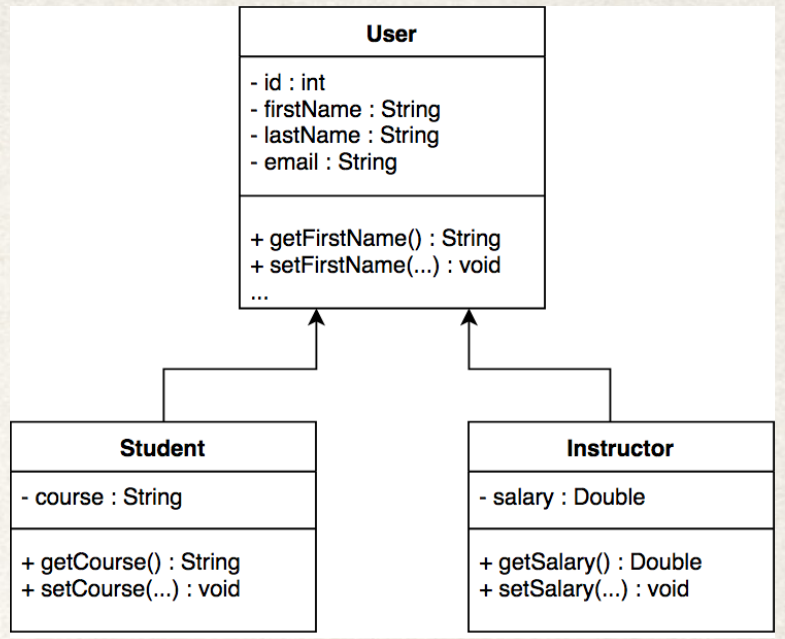
User

Student

Instructor

© luv2code LLC

Run the App



User

- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student

- course : String

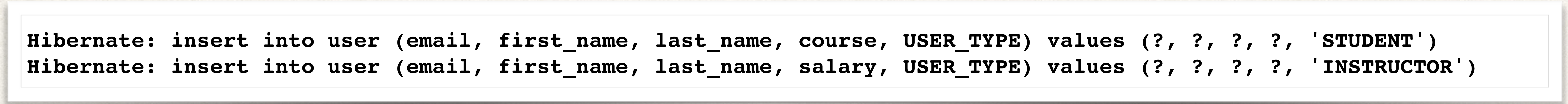
+ getCourse() : String
+ setCourse(...) : void

Instructor

- salary : Double

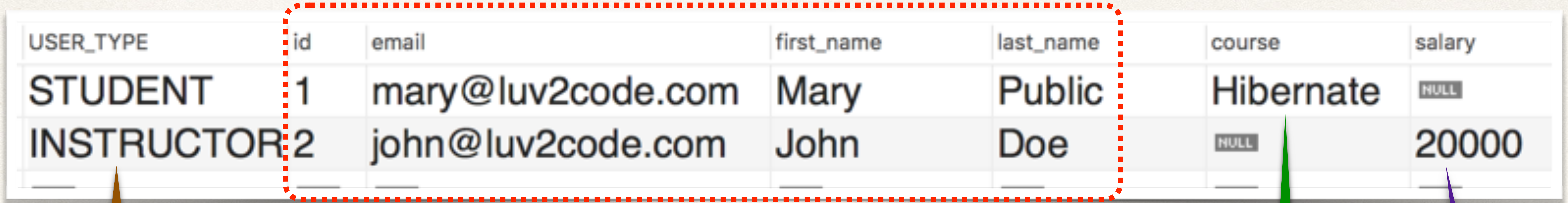
+ getSalary() : Double
+ setSalary(...) : void

Console



```
Hibernate: insert into user (email, first_name, last_name, course, USER_TYPE) values (?, ?, ?, ?, 'STUDENT')
Hibernate: insert into user (email, first_name, last_name, salary, USER_TYPE) values (?, ?, ?, ?, 'INSTRUCTOR')
```

Table: user



USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

Discriminator column
The class/type of data

User

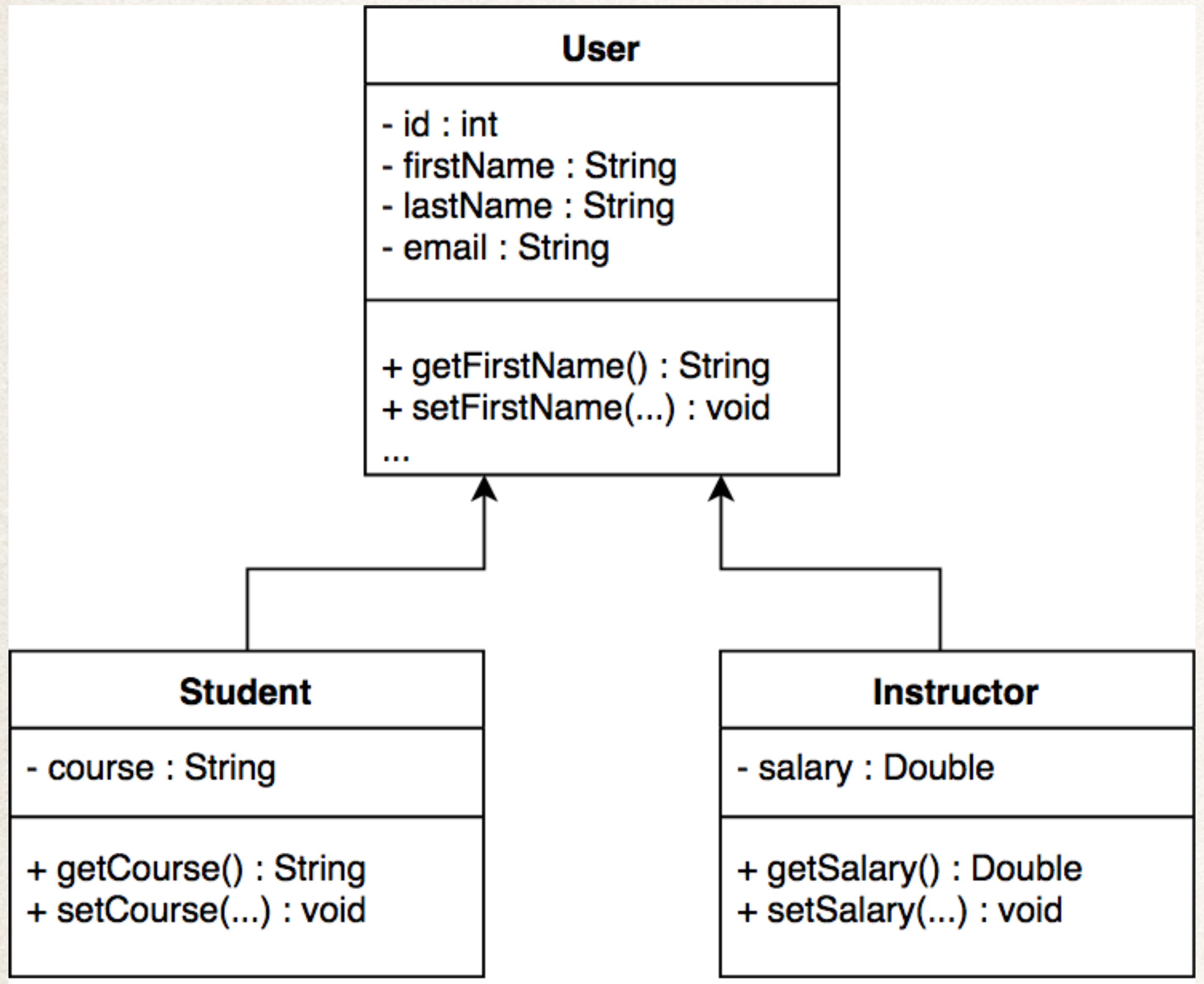
Student

Instructor

Each row uses subset of fields
Unused fields are null

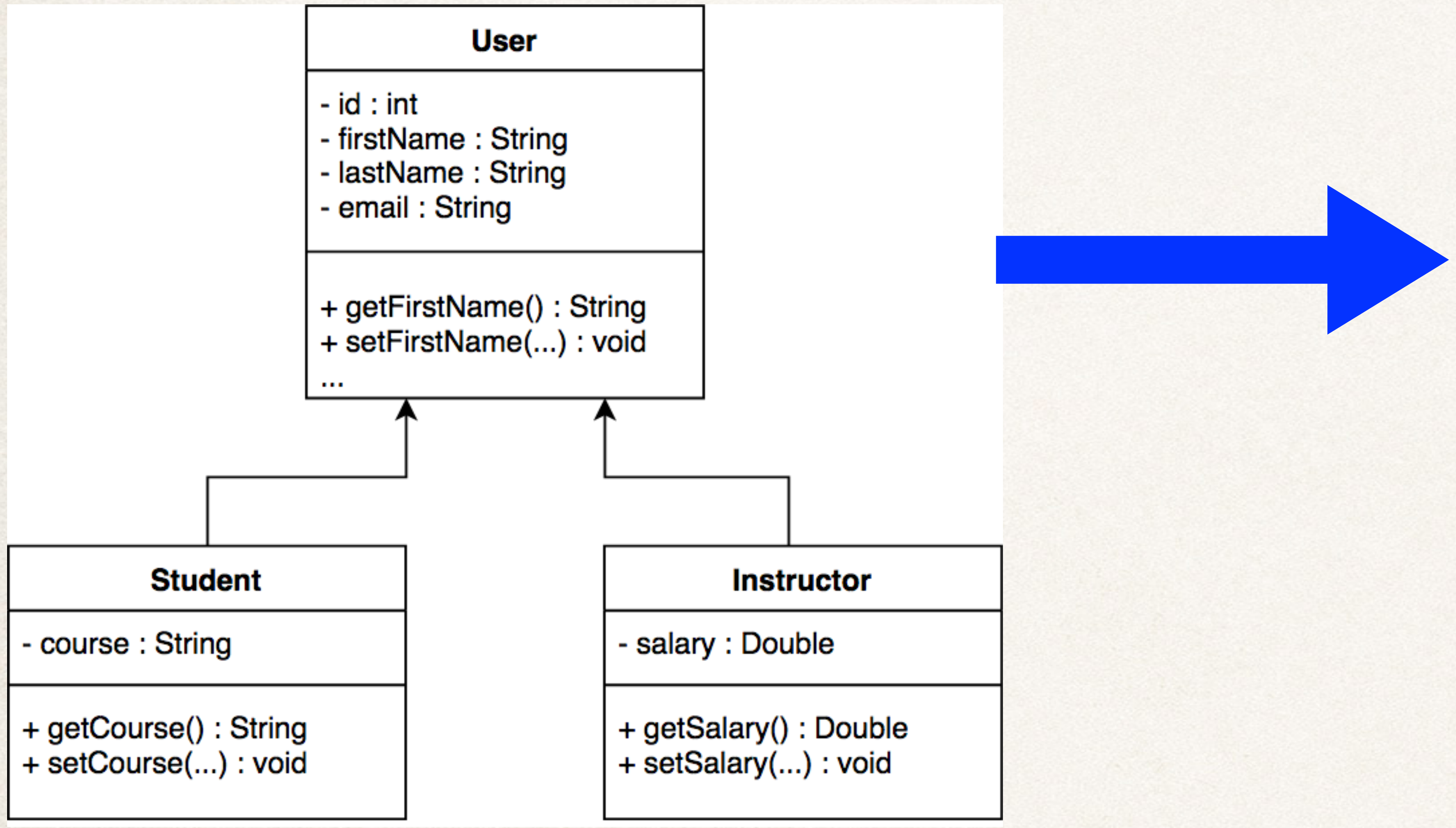
© luv2code LLC

Inheritance Strategy - Single Table



© luv2code LLC

Inheritance Strategy - Single Table



User

- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

Student

- course : String

+ getCourse() : String
+ setCourse(...) : void

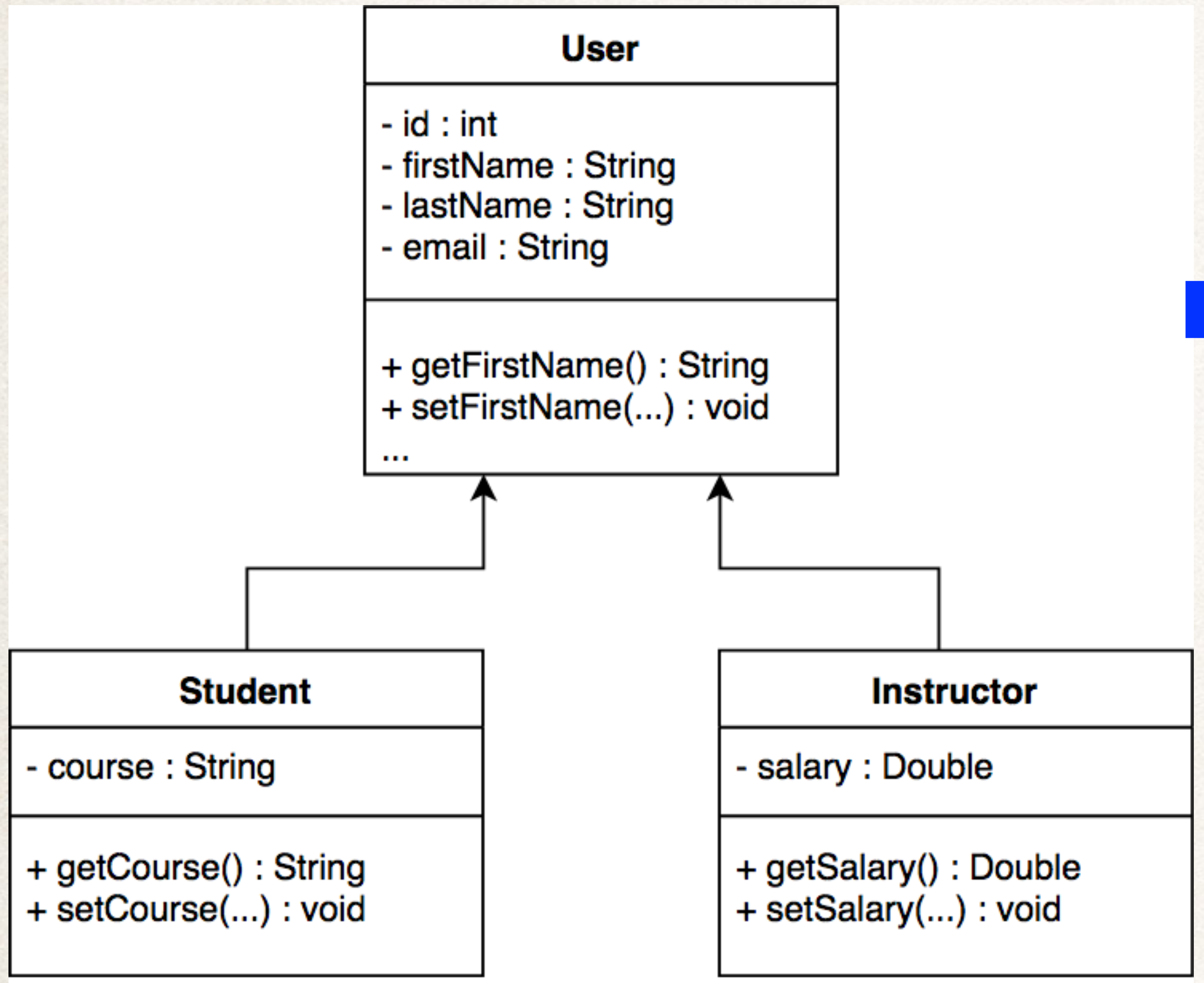
Instructor

- salary : Double

+ getSalary() : Double
+ setSalary(...) : void

© luv2code LLC

Inheritance Strategy - Single Table



User
- id : int
- firstName : String
- lastName : String
- email : String

+ getFirstName() : String
+ setFirstName(...) : void
...

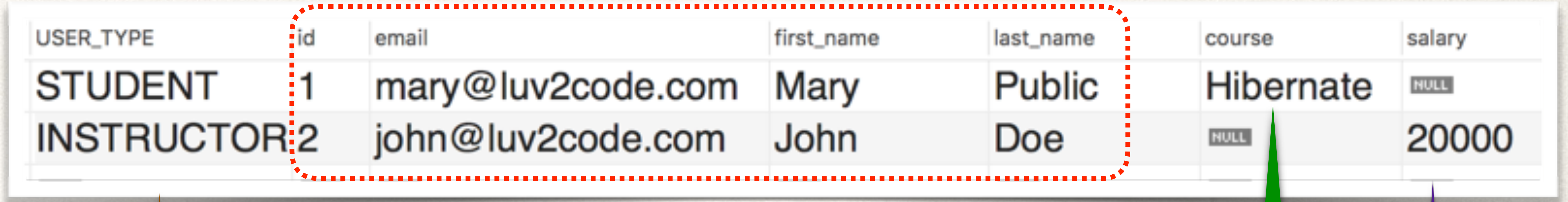
Student
- course : String

+ getCourse() : String
+ setCourse(...) : void

Instructor
- salary : Double

+ getSalary() : Double
+ setSalary(...) : void

Table: user



USER_TYPE	id	email	first_name	last_name	course	salary
STUDENT	1	mary@luv2code.com	Mary	Public	Hibernate	NULL
INSTRUCTOR	2	john@luv2code.com	John	Doe	NULL	20000

Discriminator column
The class/type of data

User

Student

Instructor

© luv2code LLC

Single Table

Single Table

- Advantages

© luv2code LLC

Single Table

- Advantages
  - Simple and straight-forward implementation

© luv2code LLC

Single Table

- Advantages
  - Simple and straight-forward implementation
  - Since data is in a single table, results in the best query performance

© luv2code LLC

Single Table

- Advantages
  - Simple and straight-forward implementation
  - Since data is in a single table, results in the best query performance
- Disadvantages

© luv2code LLC

Single Table

- Advantages
  - Simple and straight-forward implementation
  - Since data is in a single table, results in the best query performance
- Disadvantages
  - Each row only uses a subset of fields and sets others to null

© luv2code LLC

Single Table

- Advantages
  - Simple and straight-forward implementation
  - Since data is in a single table, results in the best query performance
- Disadvantages
  - Each row only uses a subset of fields and sets others to null
  - Since fields are nullable, this may present issues with data integrity

© luv2code LLC