Inheritance in Java

By the end of this video you will be able to…

- Use the keyword **extends** with confidence
- Explain the relationship between a superclass and a subclass
- Use UML Diagrams to display class hierarchies

What did we want?

1. Keep common behavior in one class
2. Split different behavior into separate classes
3. Keep all of the objects in a single data structure

```
public class Person
{
 private String name;
 …
}
```

```
public class Student
{
 private String name;
 …
}
```

```
public class Faculty
{
 private String name;
 …
}
```

```
public class Person
{
 private String name;
 …
}
```

```
public class Student extends Person
{
 private String name;
 …
}
```

"extends" means "inherit from"

```
public class Person
{
 private String name;
 …
}
```

base/ super class

```
public class Student extends Person
{
 private String name;
 …
}
```

derived/ sub class

```
public class Person
{
 private String name;
 …
}
```

```
public class Student extends Person
{
 private String name;
 …
}
```

What is inherited?

- Public instance variables

```
public class Person
{
 private String name;
 …
}
```

```
public class Student extends Person
{
 private String name;
 …
}
```

What is inherited?

- Public instance variables
- Public methods

```
public class Person
{
 private String name;
 …
}
```

```
public class Student extends Person
{
 private String name;
 …
}
```

What is inherited?

- Public instance variables
- Public methods
- Private instance variables

```
public class Person
{
 private String name;
 …
}
```

```
public class Student extends Person
{
 private String name;
 …
}
```

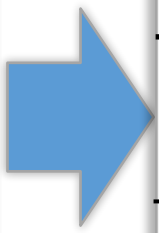
Private vars can be accessed only through public methods!

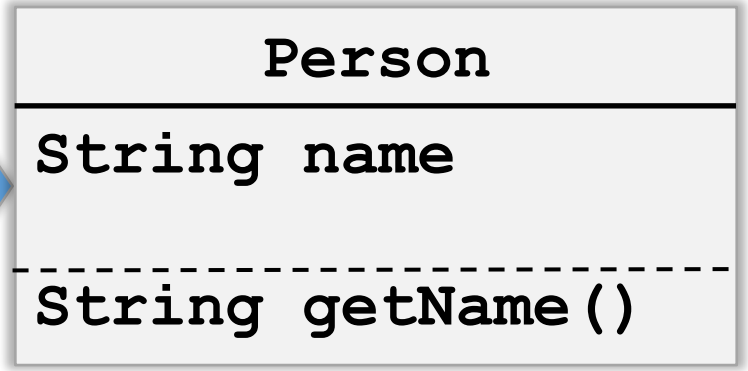
What is inherited?

- Public instance variables
- Public methods
- Private instance variables

```
public class Person
{
 private String name;
 public getName() { return name;}
}
```

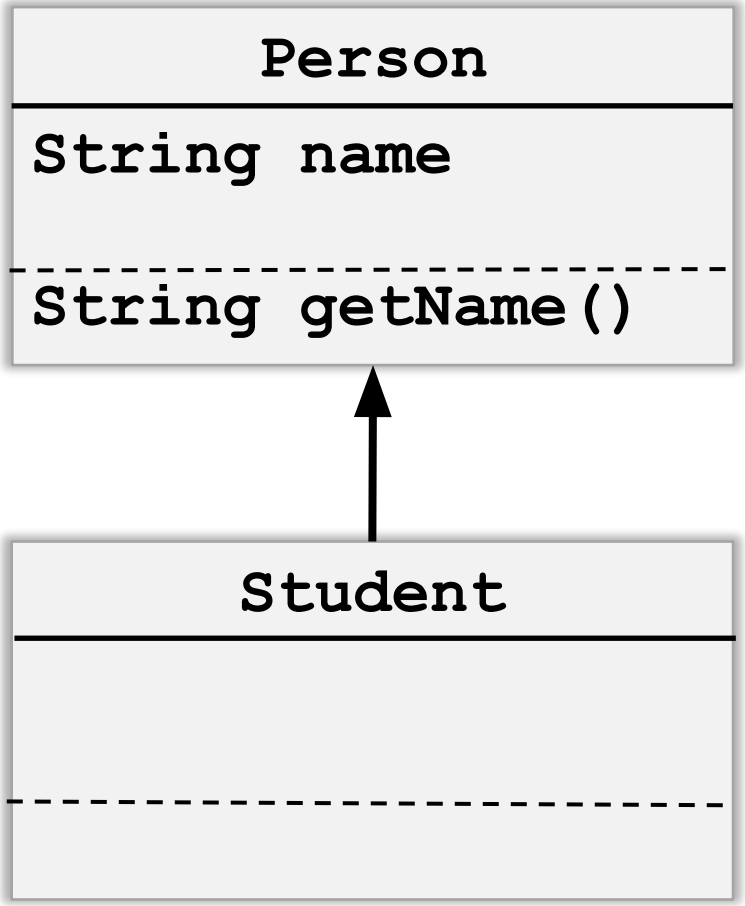
Too much detail!

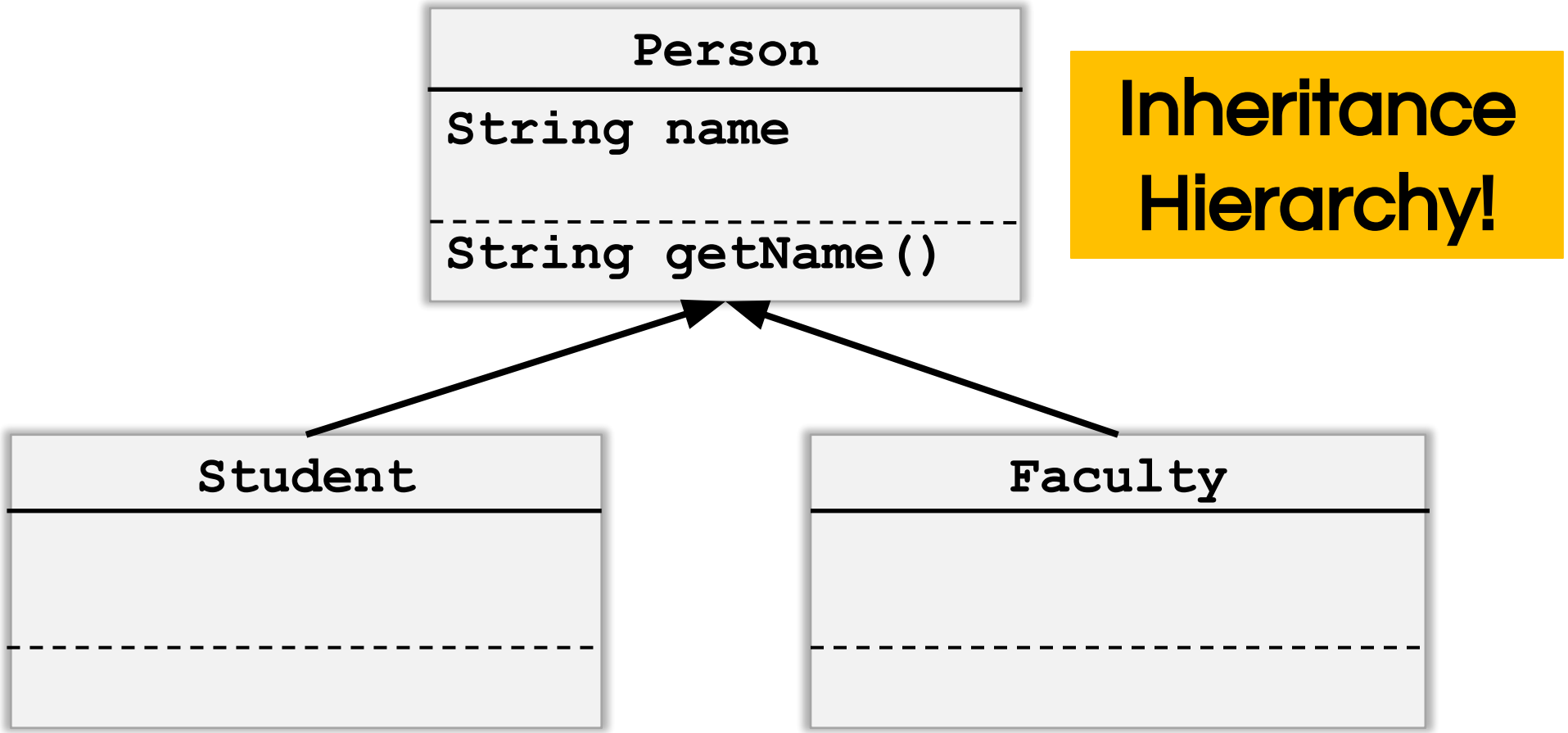




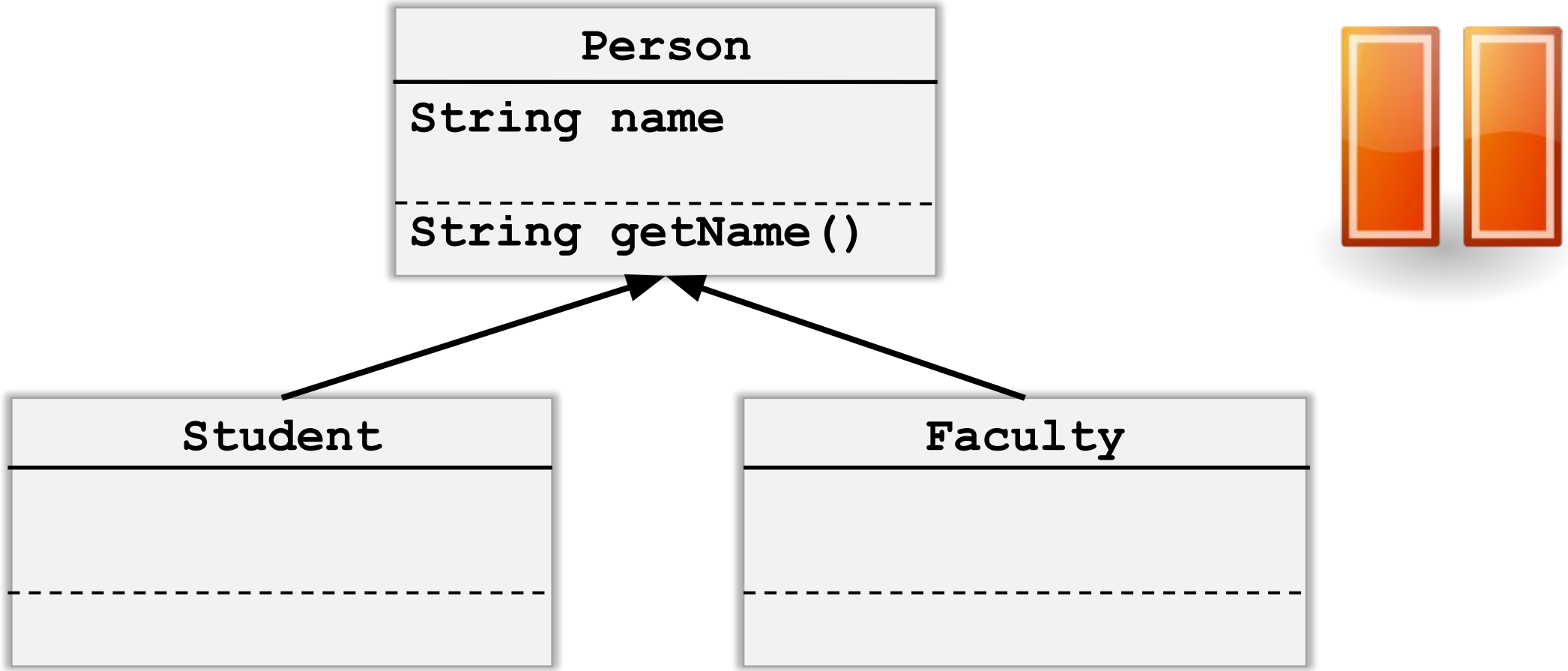
```
public class Person
{
 private String name;
 public getName() { return name;}
}
```

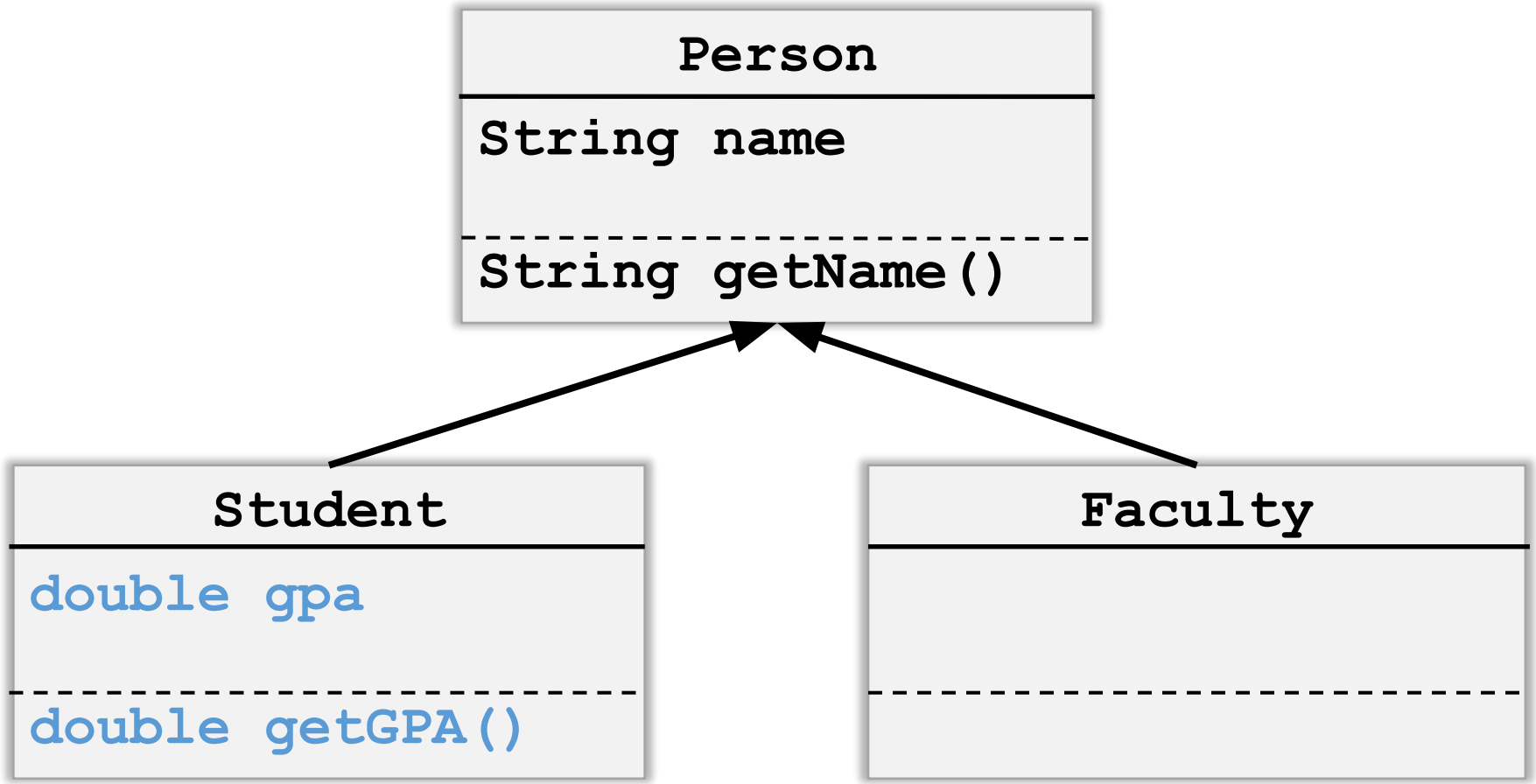
```
public class Student extends Person
{

}
```

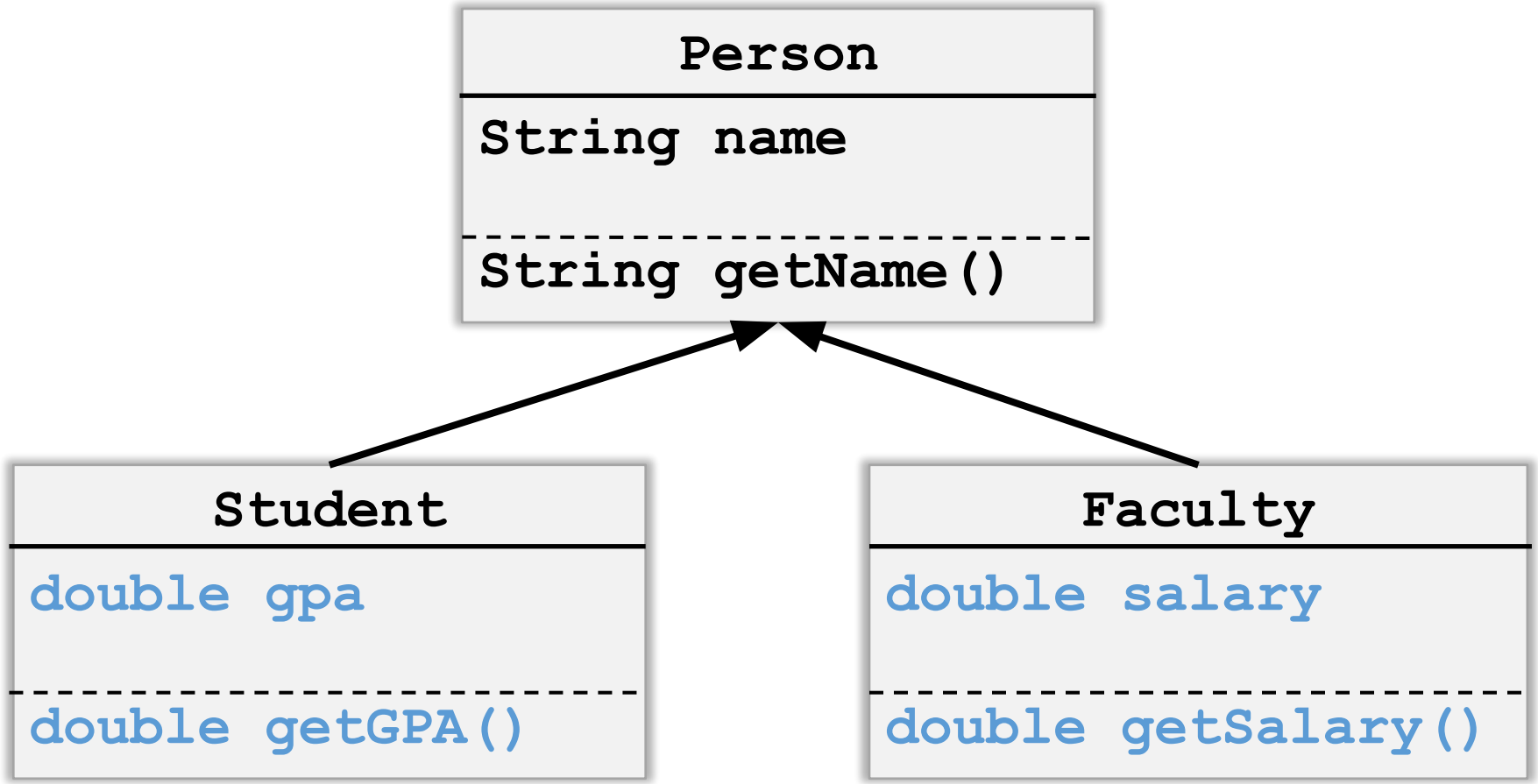




Inheritance Hierarchy!


Person
String name
String getName()
Student
Faculty


Person
String name
String getName()
Student
double gpa
double getGPA()
Faculty


Person
String name
String getName()
Student
double gpa
double getGPA()
Faculty
double salary
double getSalary()

What did we want?

1. Keep common behavior in one class
2. Split different behavior into separate classes
3. Keep all of the objects in a single data structure

What did we want?

1. Keep common behavior in one class
2. Split different behavior into separate classes
3. Keep all of the objects in a single data structure

Coming up!