**Amadou Sarjo Jallow**

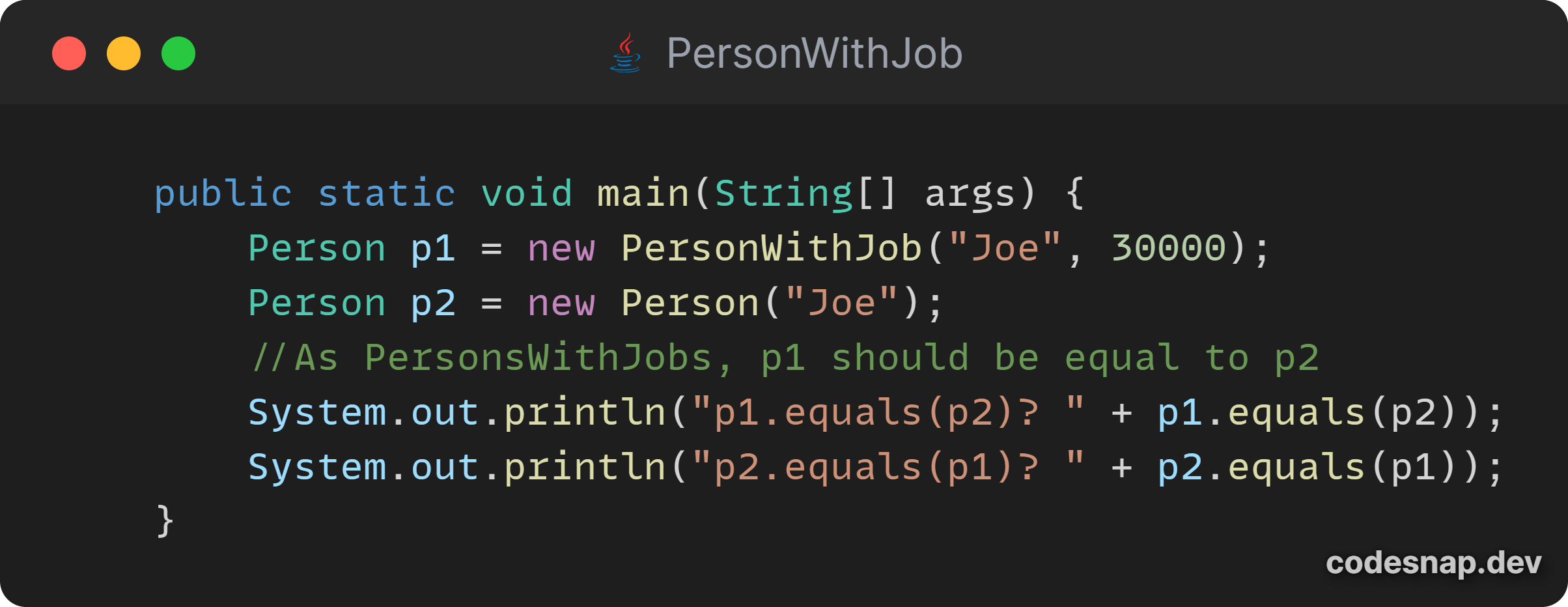
**619016**

**Lab Solutions**

1**.**

**A. Explanation of the comparison of the two person instances**:

|  |  |
| --- | --- |
|  |  |



**p1** is a **PersonWithJob** object and **p2** is a **Person** object.

* In **p1.equals(p2)**: The **equals** method of **PersonWithJob** is called. This checks if **p2** is an instance of **PersonWithJob**. Since **p2** is only an instance of **Person**, it returns **false**.
* In **p2.equals(p1)**: The **equals** method of **Person** is called. This checks if **p1** is an instance of **Person**, which it is true as a **PersonWithJob** is also a **Person**. Since the instance test is passed, this only checks the name field, which matches, so it returns **true**.

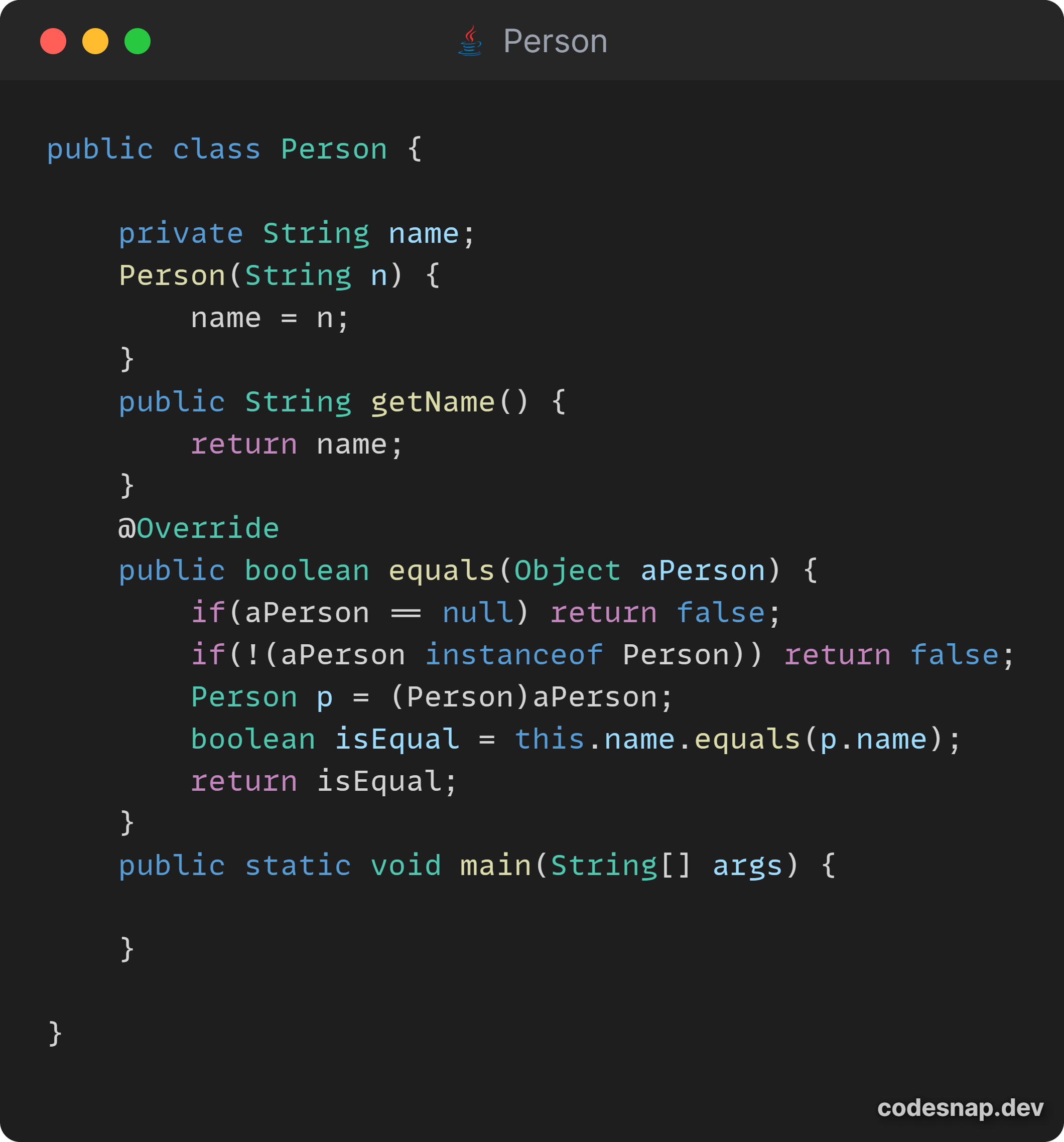
**B. Solution by replacing inheritance with composition**

Source:

Lab3 - code/code for prob1/lesson3/labs/prob1/composition/Person.java

Lab3 - code/code for prob1/lesson3/labs/prob1/composition/PersonWithJob.java





2.

Source:

Lab3 - code/code for prob2/Apartment.java

Lab3 - code/code for prob2/Building.java

Lab3 - code/code for prob2/LandlordInfo.java

Lab3 - code/code for prob2/Main.java

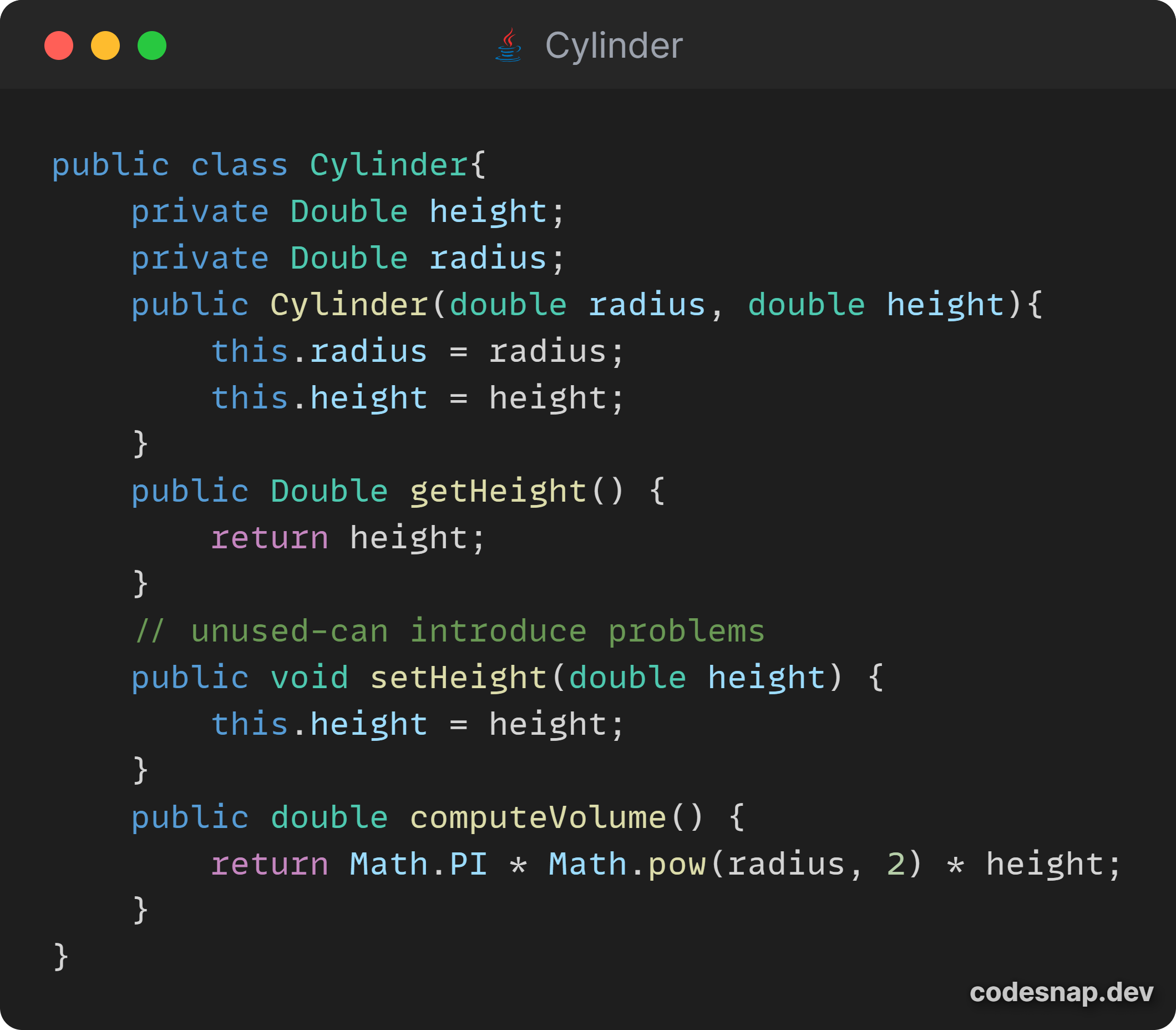
|  |  |
| --- | --- |
|  |  |
|  |  |

3.

**A. Does it make sense to use inheritance here? Explain.**

Source:

Lab3 - code/code\_for\_prob3/inheritance



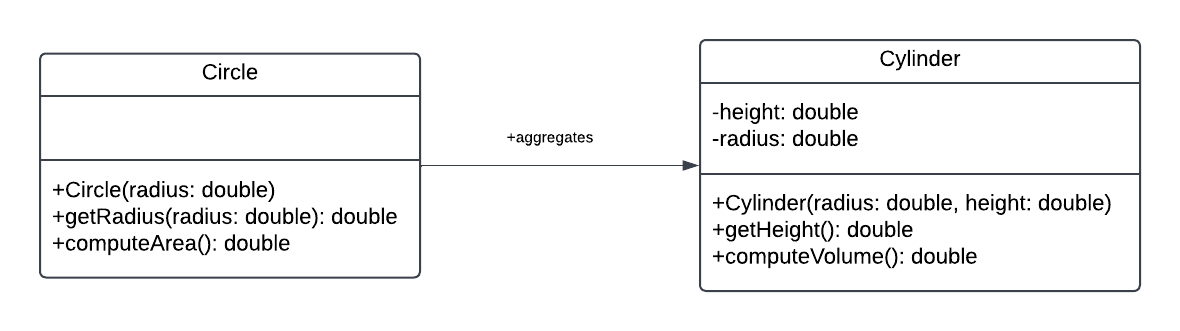
|  |  |
| --- | --- |
|  |  |

In the code implementation, it is clear to see that inheritance is not the best implementation for this relationship. It also violates both the guiding principles for using inheritance as a circle is not a cylinder and cannot be substituted for every case a cylinder object is needed. From the Main method, it is clear that you can create a circle instance type and for polymorphism, create a circle of type Cylinder and everything works fine. But the problem arises when you add setters to the superclass, setHeight() in this scenario. Although a circle has nothing to do with the height property but when you update the height property in this case, the side value and area of the circle will change even though it should not because a circle has a side property and not a height property. To address this issue, you can use composition.

**B. Design using composition and Code.**

Source:

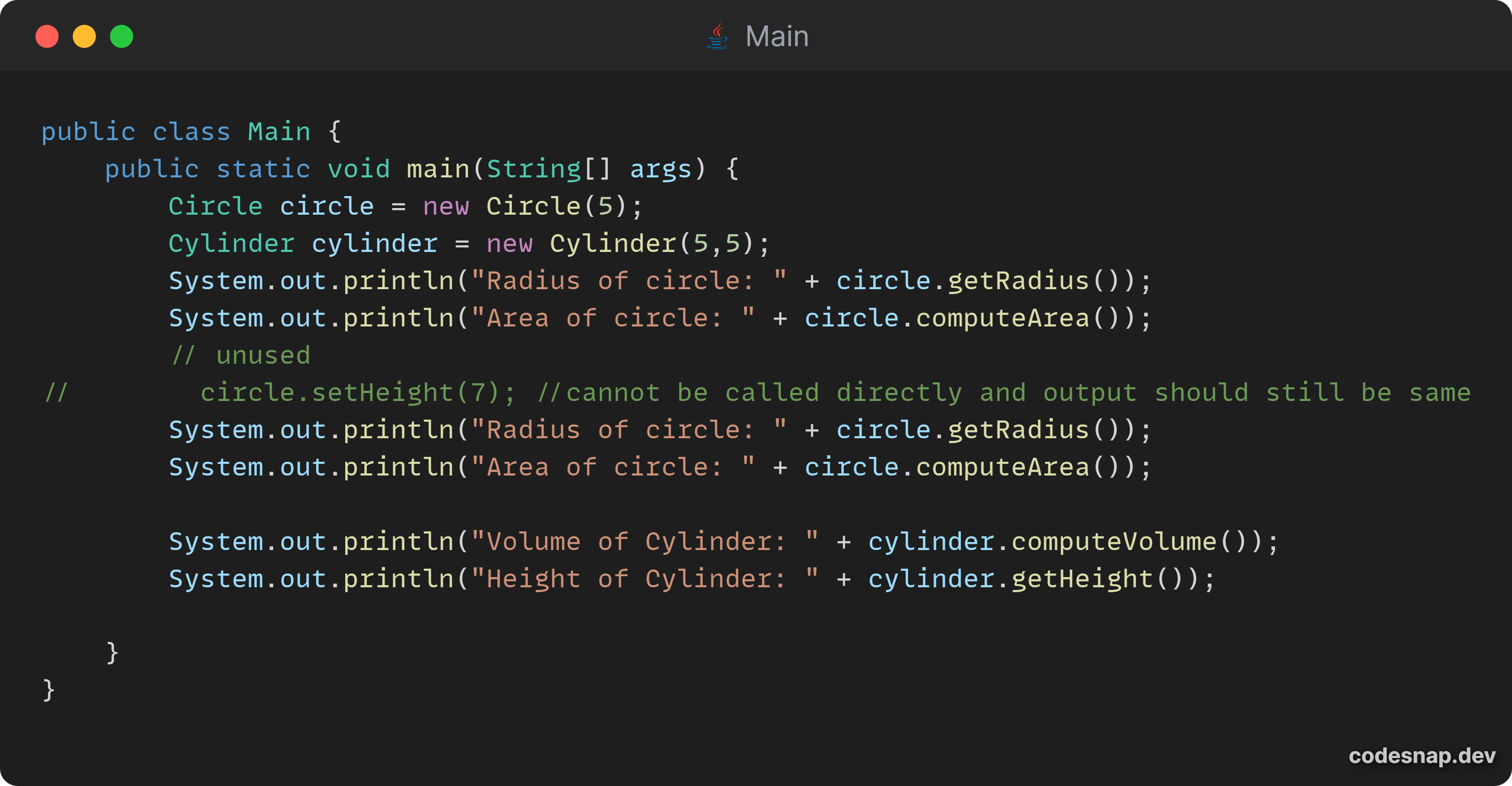
Lab3 - uml/Prob3/Composition-Circle-Cylinder.png



Source:

Lab3 - code/code\_for\_prob3/composition

|  |  |
| --- | --- |
|  |  |



4.

Source:

Lab3 - code/code for prob4/

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

