**Exercise: 4**

**Name: Pekka Lehtola**

**How many tasks did you do: 8**

**Were the tasks easy, ok, difficult: Ok**

**Do you need help/comments in any task (if yes, to which ones):**

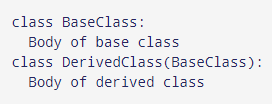
1. Explain the following terms and what they are used for:

a. Inheritance (in object-oriented programming)

* It refers to defining a new class with little or no modification to an existing class. The new class is called derived (or child) class and the one from which it inherits is called the base (or parent) class.

Derived class inherits methods and attributes from parent/base class

* Example of inheritance:

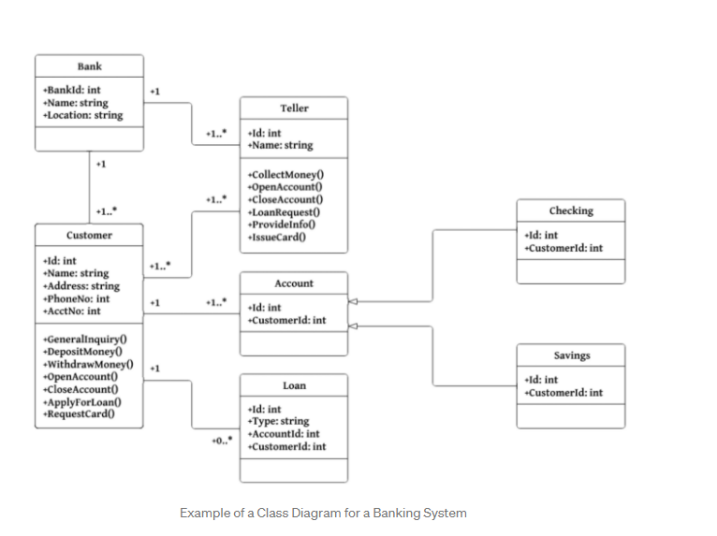


b. UML

* UML or Unified Modeling Language is way of modeling and documenting software.
* It is based on diagrammatic representations of software components. with visual representations, possible flaws or errors in software are detected.

c. UML class diagram

* a class diagram in UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
* Example of class diagram:

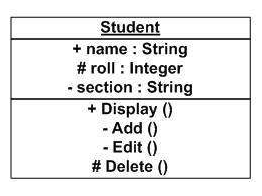
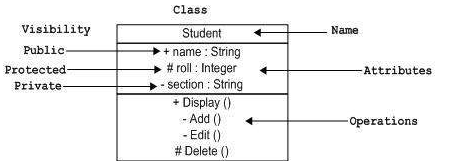


2. Answer the following question.

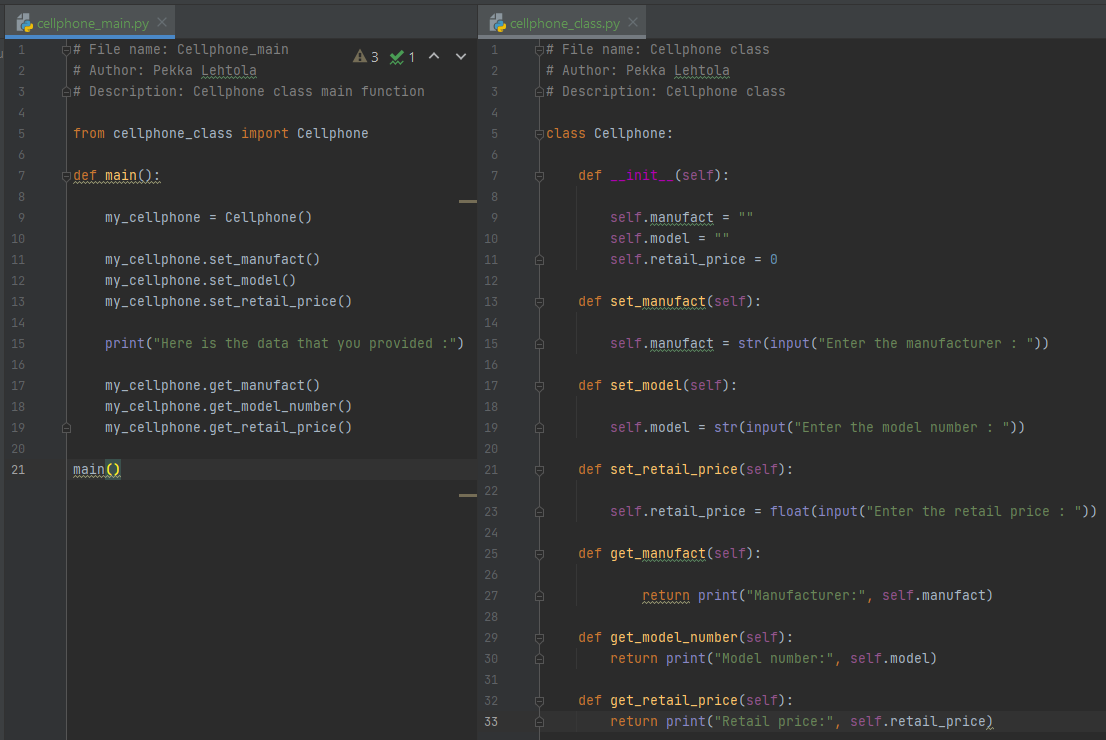
When you model using UML diagrams, why is it important to folllow the UML syntax strictly?

Efficient and appropriate use of notations is very important for making a complete and meaningful model. The model is useless, unless its purpose is depicted properly.

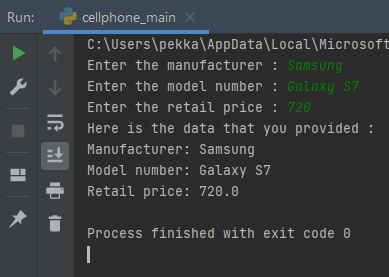
This model useless with out knowing the correct syntax. UML is very barebones regarding text so every model needs to follow correct syntax to make it readable to everyone.



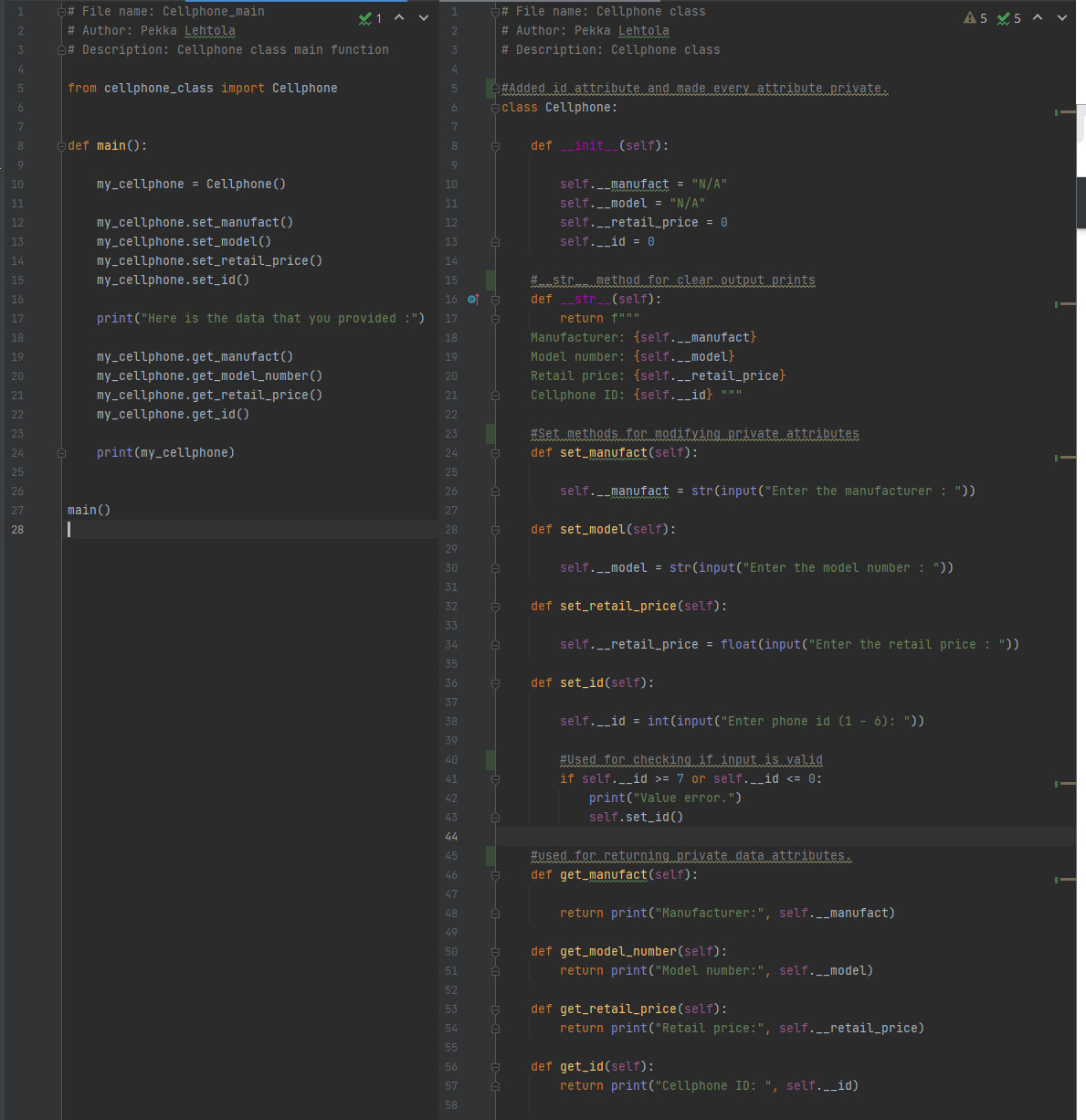
3. Take the cell phone class of last week and divide the cell phone class into another file (name the file clearly). Leave the main function in the original file. Test, that your code still works.

Screen capture of Task 3

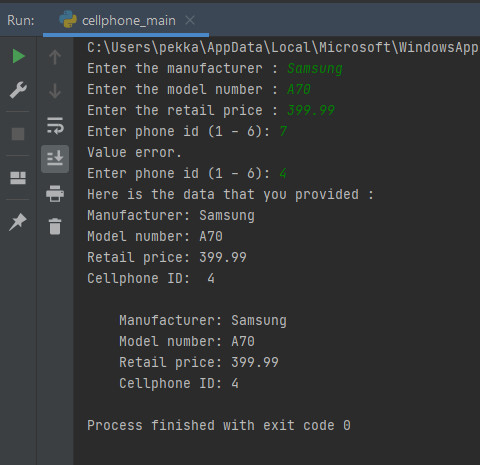
Screen capture of the output of Task 3



4. Add and ID data attribute (integer between 1-6) to the cell phone. Cell phone class shall have accessor and mutator methods for all data attributes. Also check the \_\_str\_\_ method is up to date.

Screen capture of Task 4

Screen capture of the output of Task 4



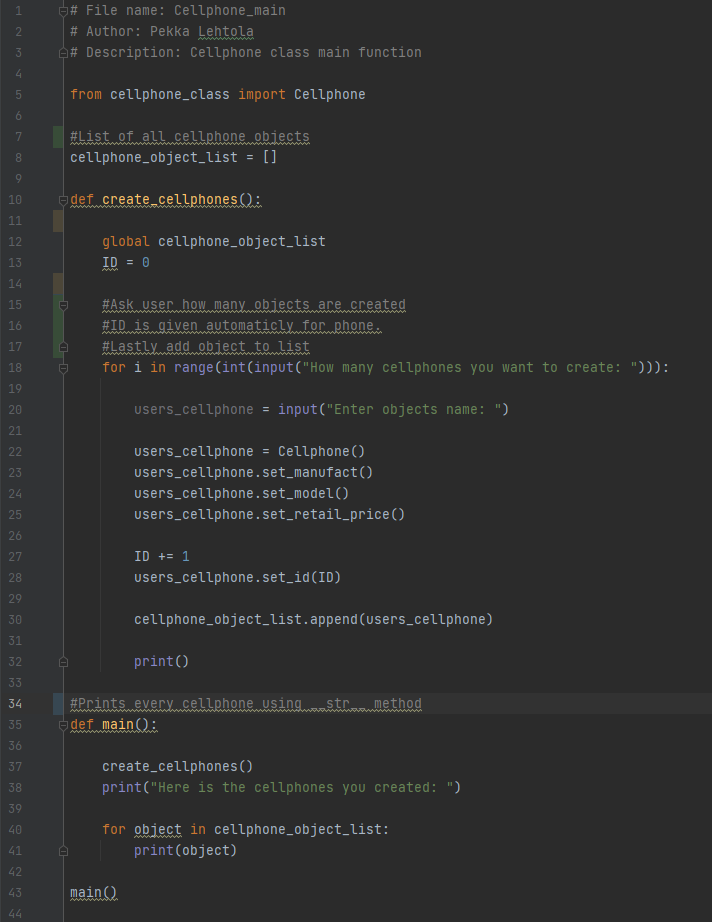
I had outputs with accessor and with \_\_str\_\_ method.

5. Create different cell phone objects (which have different data attribute values, use

mutator methods to change the data attribute values). Print out each object’s state (use

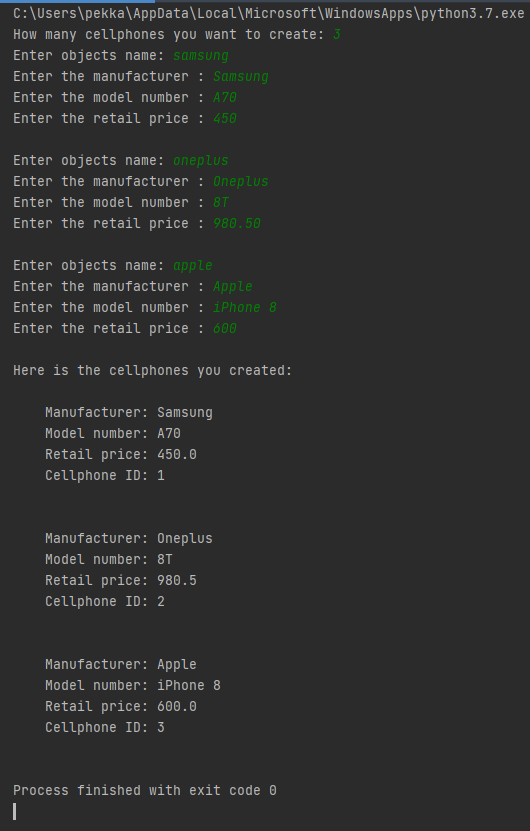
the \_\_str\_\_ method in the cell phone class).

Screen capture of Task 5

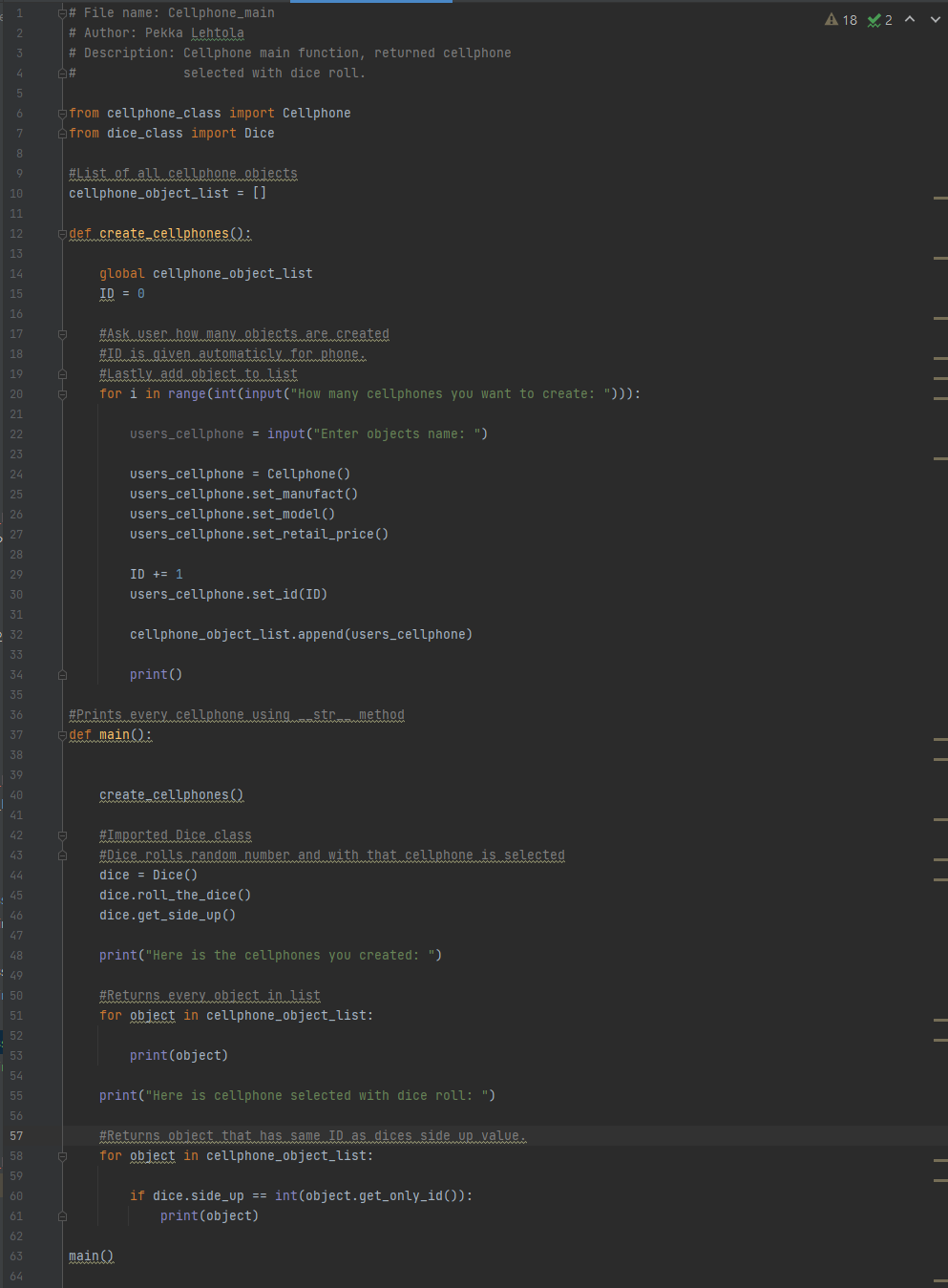


No changes to Cellphone class.

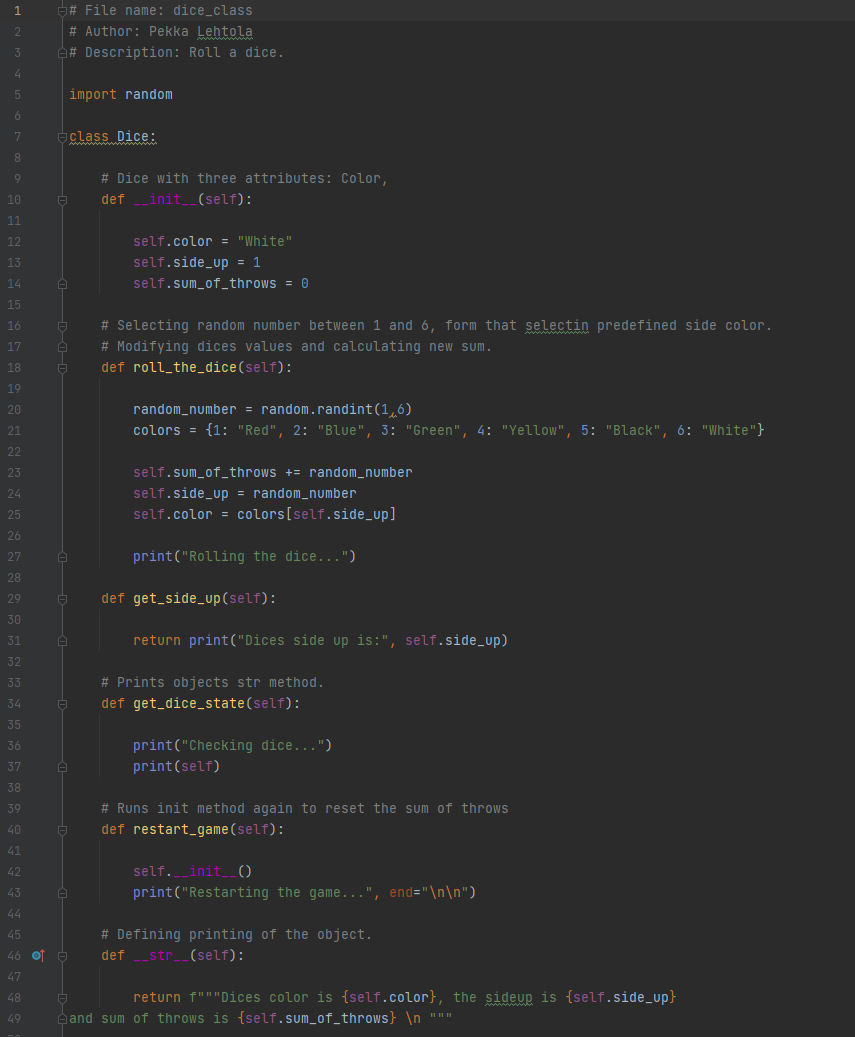
Screen capture of the output of Task 5



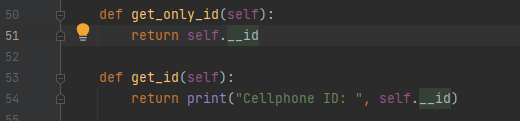
6. Take the Dice class from your earlier exercises and place that to its own file. Then in main function roll a dice and based on the result choose the correct cell phone based on the ID. Print out the chosen cell phone object’s state.

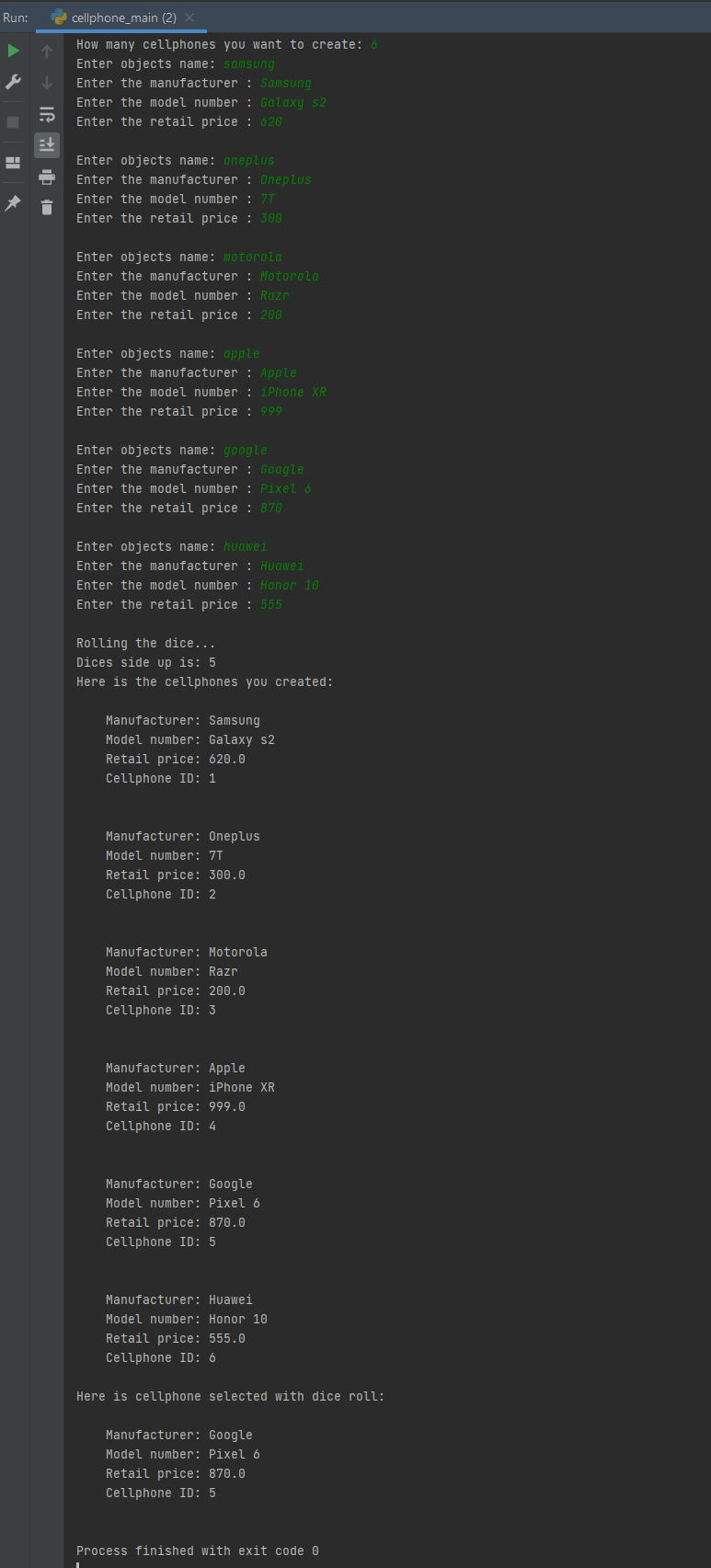
Screen capture of Task 6

Modified Dice class.



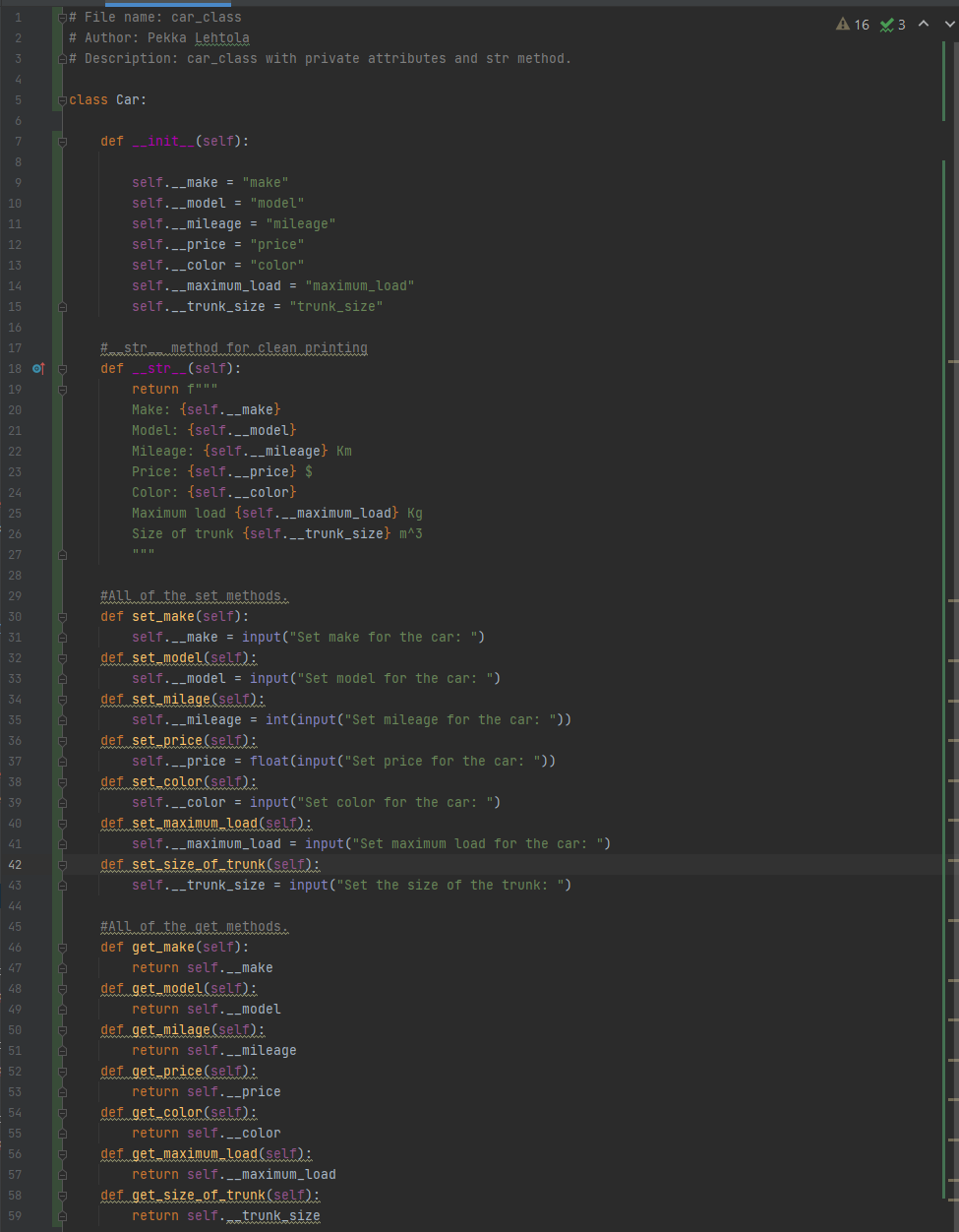
Added get\_only\_id to Cellphone class.

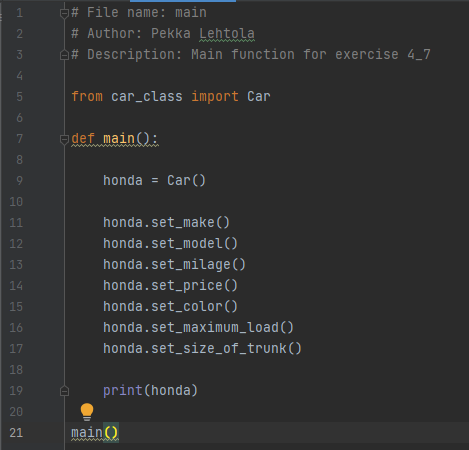


Screen capture of

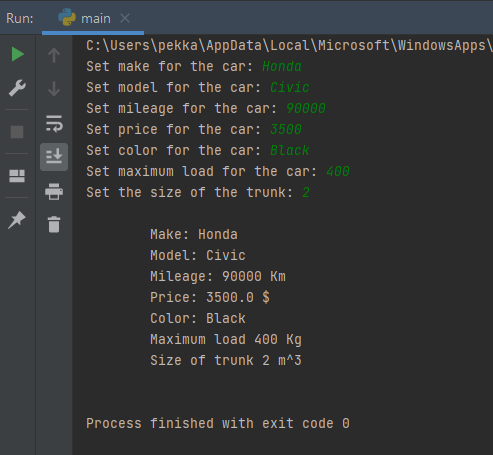
the output of Task 6

7. Create a car object. It has the following data attributes: make, model, mileage, price, color, maximum load limit, size of trunk. Make them private. Write accessor and mutator methods to change them. Add \_\_str\_\_ method to print the state of the car.

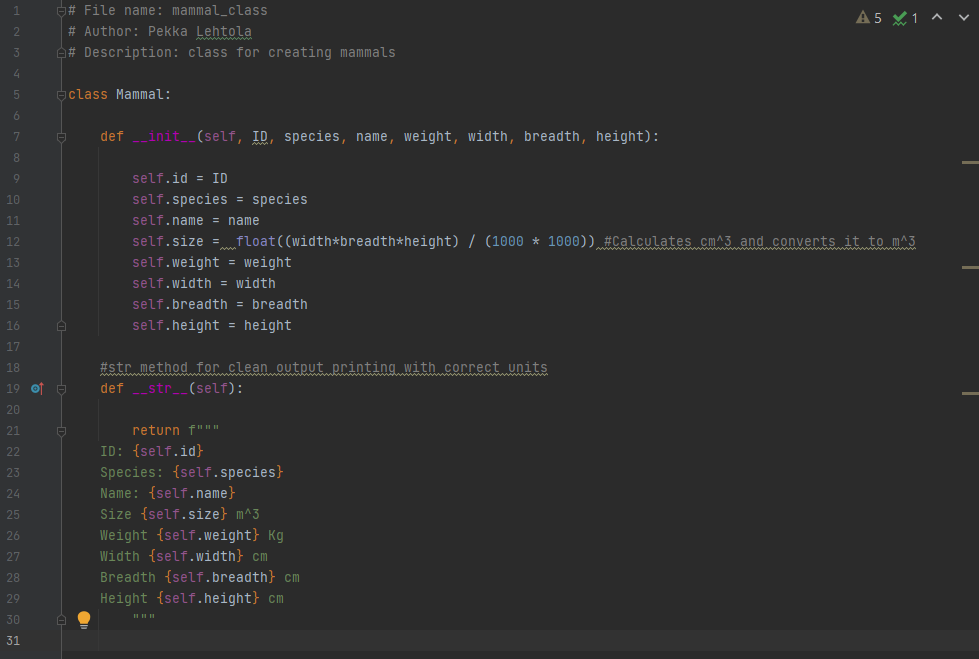
Screen capture of Task 7

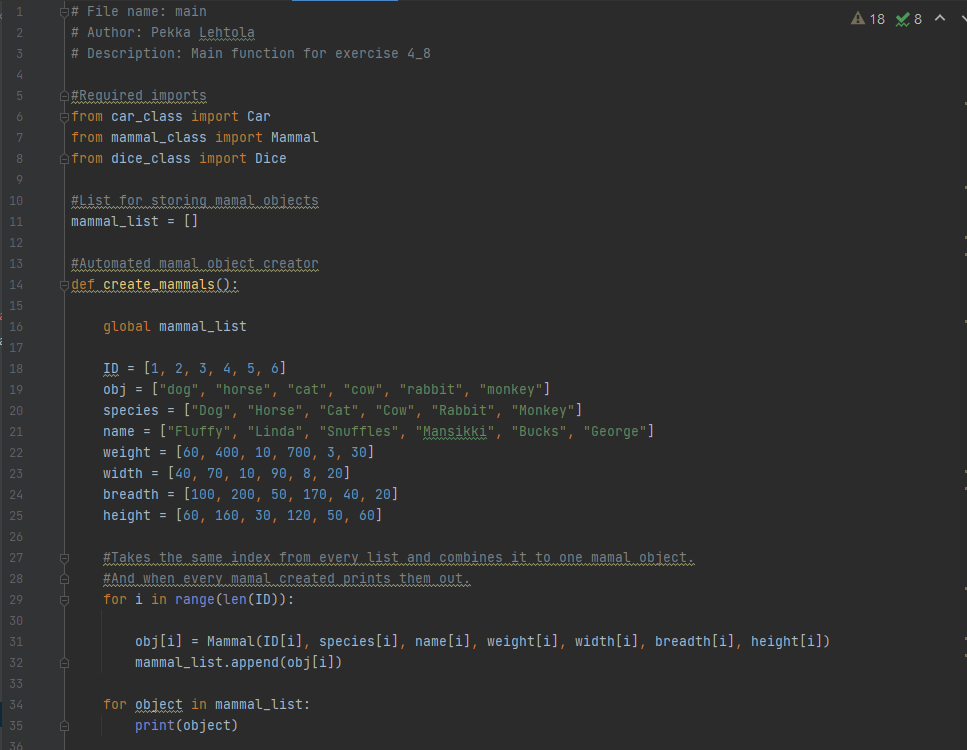


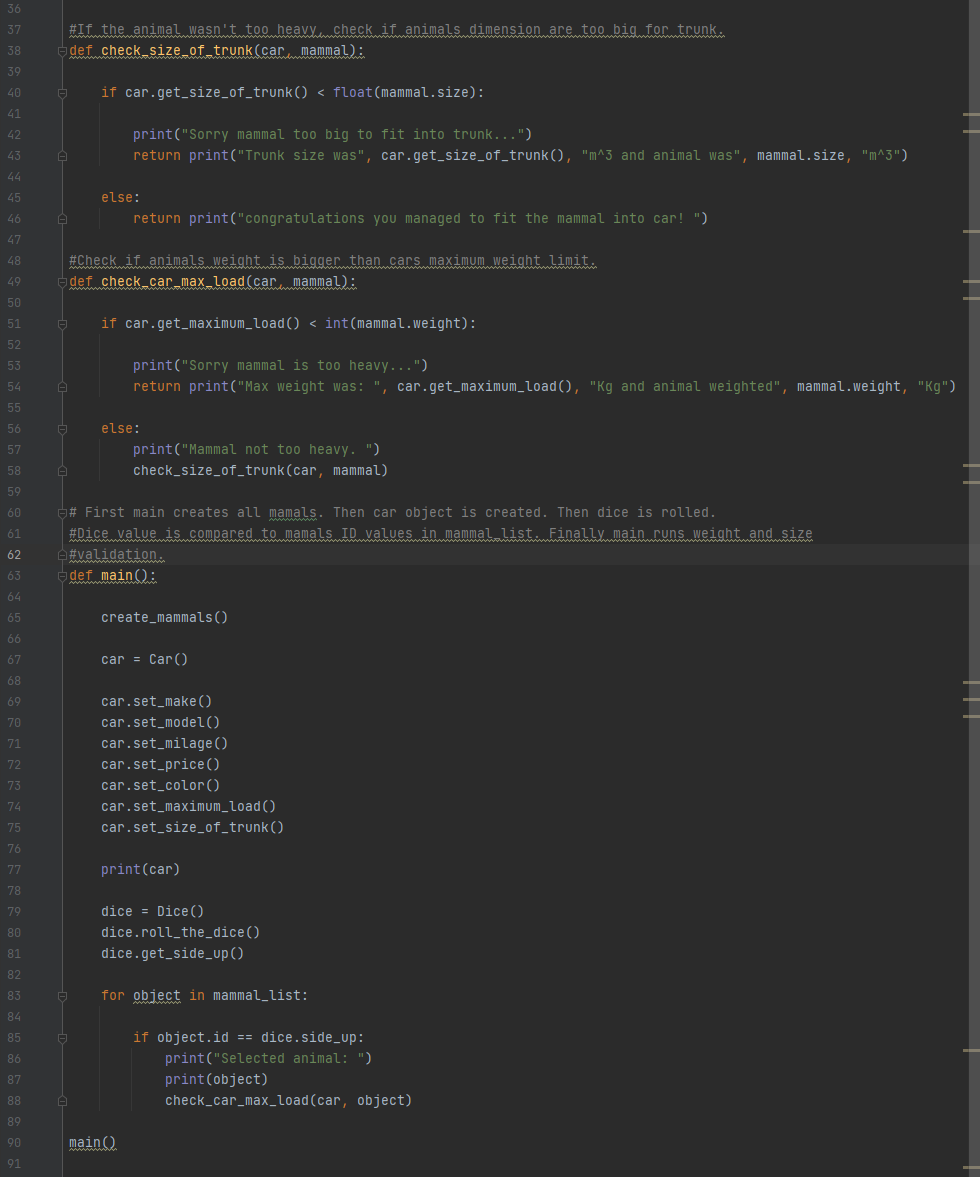
Screen capture of the output of Task 7



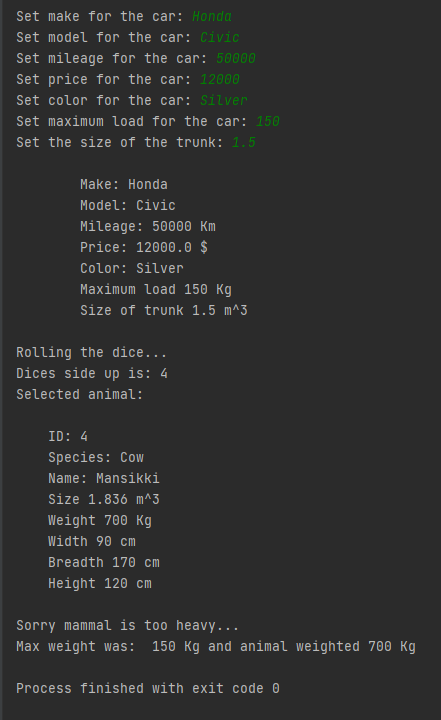
Screen capture of Task 8

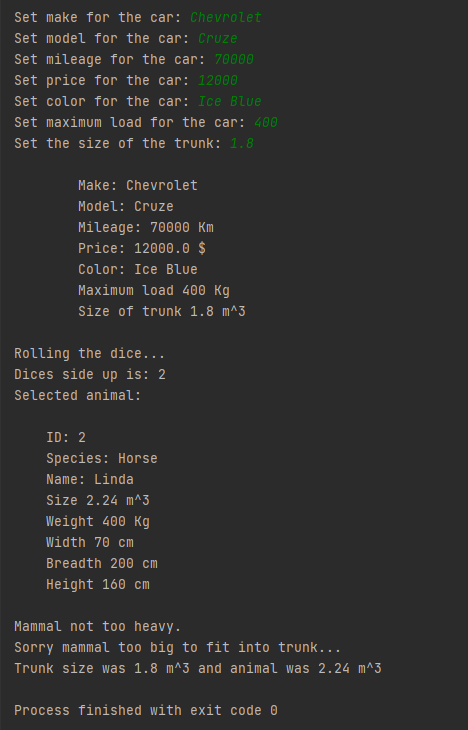
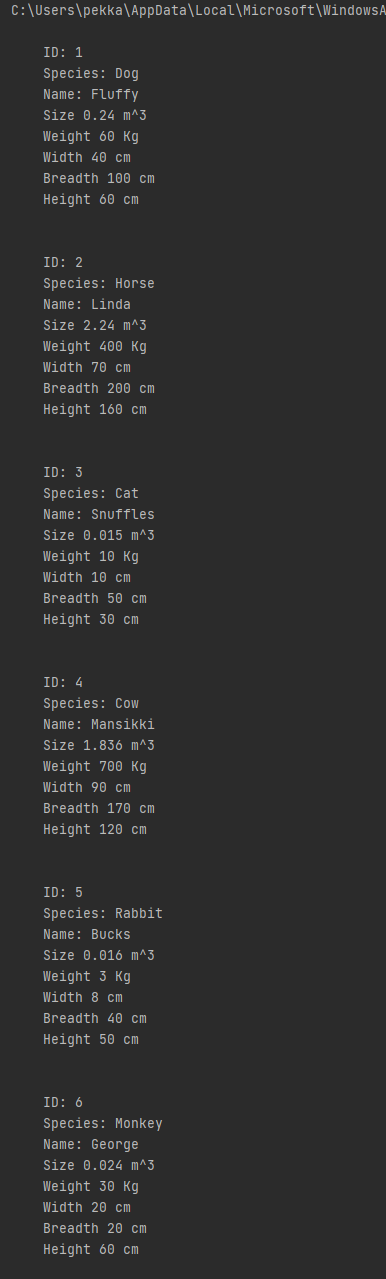
Mammal class: 

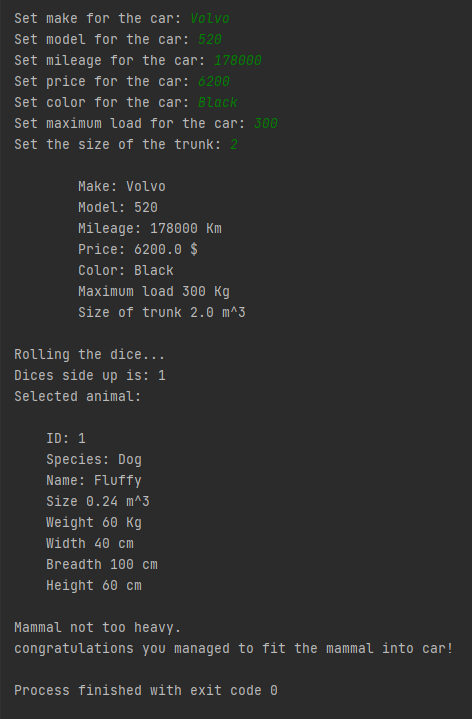
Main:



Dice and Car class unchanged.

Screen capture of the output of Task 8

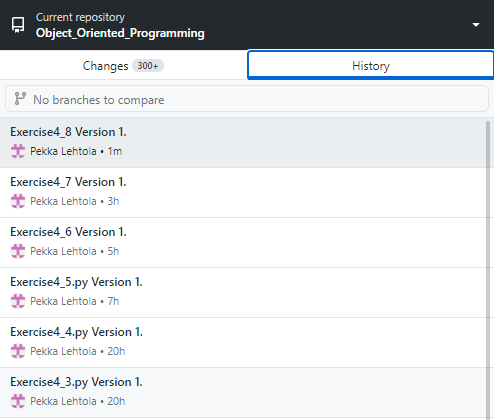




Mammal creating was the same in every instance.

Mammals were printed First.

Screen capture of git log (showing that you made a commit after every task).



Self-assessment:

This exercise was easy/difficult/ok/etc. for me because…

Tällä viikolla tehtävät olivat ihan ok, tietenkin oli joitakin haasteita, mutta vikastakin tehtävästä selvisin ihan hyvin.

Doing this exercise, I learned…

Objectien sijoitusta listaan ja objection kommunikointia toistensa kanssa.

I am still wondering…

Vaikka onnistuin sijoittamaan objecteja listoihin ja käyttämään niitä sieltä, olen aika epävarma kyseisen käytännön kanssa vielä. En tiedä missä muodossa ne tallentuvat sinne ja miten käyttäisin niitä listasta tehokkaasti. En keksinyt esimerkiksi miten kutsuisin niitä listasta objectin nimen mukaan.

I understood/did not understand that… ; I did/did not know that… ; I did/did not manage to do…

Onko olemassa järkevämpää tapaa hakea objecti listasta kuin tämä. Esimerkiksi jos olen luonut kolme objectia (Kissa, Koira, Lisko) onko suoraa tapaa kutsua Liskoa listasta ilman for looppia tai käyttämällä indexia missä kohtaa sijaitsee listassa.

