Programming

New Austrian Coding School

week 2 - exercises

Exercise 1

Matilda has finished writing an article and wants to print it.
Develop the code that she uses to write an article and print it.
Follow the instructions below:

- You need the following classes: *Article, Printer* and *PrintApplication.*
- An *Article* has a *title* and a *text*. It also has the methods *getTitle* and *getText*.
- A *Printer* can *print* an *article,* which means it just writes the *title* and the *text* on the screen.
- In the *PrintApplication* class *main* method, create an *Article* and a *Printer* and *print* the *article* with the *printer*.

Exercise 2

Hansel arrived home after a hard day of work and he reaches his key to open the door.
Develop the code that Hansel uses to create a key and open the door.
Follow the instructions below:

- You need the following classes: *Key, Door* and *DoorApplication*.
- The *Door* has a *number*. It also has a *getNumber* method.
- The *Key* has no attributes. It has the *open* method that receives a *door* and says that it's opening the door with its door number.
- In the *DoorApplication* class *main* method, create a *Door* and a *Key*, and open the *door* with the *key*.

Exercise 3

Derek wants to hang a painting at home but he realized that for that he needs to buy a hammer.
Develop the code that is used to give him the tool and ask him to hang the painting.
Follow the instructions below:

- You need the following classes: *Derek*, *Tool* and *PaintingApplication*.
- A *Tool* has a name. It also has a *getName* method.
- *Derek* has a *tool* as attribute, but he does not receive it through the constructor. Instead he receives it with a *setTool* method. He also has the methods canHangPainting and *hangPainting*.
- In the *canHangPainting* Derek checks if he has a tool called hammer. If so he returns true, otherwise false.
- In the *hangPainting* method, if Derek *canHangPainting* it will say he can hang the painting. Otherwise he will say that he's missing the hammer.
- In the *PaintingApplication* class *main* method, create a *Derek* object and ask him to hang the painting. He should say he's missing the hammer. Later, create a hammer *Tool*, give it to *Derek* and ask him again to hang the painting. He should say that he can hang the painting.

Exercise 4

Mugatu is thinking about buying another car but he's not sure which brand he should choose. He goes to a car shop and the seller says that every brand is particularly good, which confuses him.
Develop the code that the seller uses to confuse Mugatu saying that every car is particularly good.
Follow the instructions below:

- You need the following classes: *Seller, Car* and *CarShopApplication*.
- A *Car* has a brand. It also has the *getBrand* method.
- A *Seller* has no attributes. It has the *describe* method that receives a *Car* and it says that that car's brand is particularly good.
- In the *CarShopApplication* class *main* method, create a *Seller* and three *Car*s with different brands and use the *Seller* to describe the three of them. He should say that the three of them are particularly good.

Exercise 5

Matilda's company wants to publish a magazine and they want to see a preview of the articles included by the editors.
Develop the code that the editors use to write the magazine and have a preview of it.
Follow the instructions below:

- You need the following classes: *Article, Magazine, Editor, Printer* and *MagazineApplication*.
- Reuse the *Article* from exercise 1 (don't copy it, reuse the existing one from the package where it already is).
- A *Magazine* has a *title* and a list of *Article*. Its constructor only asks for the *title*. Make sure its *articles* are initialized with an empty collection. It also has the method *add* to add an *Article* to its list and the method *getArticles* that returns the list. It also has the method *getTitle*.
- An *Editor* has no attributes. It has the *writeMagazine* method that returns a *Magazine*. In order to create a *Magazine* it creates first three *Articles*, it adds them to the *Magazine* and then it returns it.
- Reuse the *Printer* from exercise 1 (don't copy it, reuse the existing one from the package where it already is). Add the new method *printMagazine* to the *Printer* that receives a *Magazine* and displays its *title* and then all the *articles*.
- In the *MagazineApplication* class *main* method, create an *Editor,* use it to write a *Magazine*, create a *Printer* and use it to print a preview of the *Magazine.*

Exercise 6

After hanging his painting on the wall, Derek realizes how few tools he has at home and he decided to buy a toolbox.
Develop the code that Derek uses to ask the shop assistant to find him a toolbox with the three tools he needs.
Follow the instructions below:

- You need the following classes: *Tool, Toolbox, ShopAssistant* and Tool*ShopApplication*.
- Reuse the *Tool* from exercise 3 (don't copy it, reuse the existing one from the package where it already is).
- A *Toolbox* has a *color* and a list of *Tool*. Its constructor only asks for the *color*. Make sure its *tools* are initialized with an empty collection. It also has the method *add* to add a *Tool* to its list and the method *getTools* that returns the list. It also has the *getColor* method.
- A *ShopAssistant* has no attributes. It has the *findToolbox* method that receives a list of String representing the names of the tools the customer asks for and returns a *Toolbox.* In order to create a *Toolbox* it creates one tool per tool requested adding them to the *Toolbox* then it returns it.
- In the Tool*ShopApplication* class *main* method, create a list of String with three tool names inside, create a *ShopAssistant and* use it to find a red *Toolbox* with those three tools inside.