All code must be coded by you and you alone. If you base your code on existing code, then you must clearly show (in comments) what the source was.

You should put all the files that the delivery contains in one folder and zip.

The solution must be able to run in IntelliJ (no other special technology/IDE is needed)

The delivery will contain two things:

1. The project folder with all relevant files to be able to run and evaluate the code you have created.

2. A small report (pdf or word document) which briefly describes how you have handled

- exceptions

- encapsulation

- heritage

and any assumptions you have made as a basis for your solution (if you believe the assignment text contained ambiguities).

The assignment

The case deals with the discovery of historical objects. A group of hobby archaeologists have collected historical objects over a period of time. Information about who has found which objects is collected in a text file (funn.txt). The file also contains information about museums where some of the objects are located. For a description of the file, see Appendix 1.

Part 1 - importing data into database:

You will create a program that reads from the file funn.txt and enters the data into a database. A proposal for the database structure can be found in the file funn.sql. You can choose to make adjustments in funn.sql, but it should not be necessary.

When you have finished part 1 you should have:

- A well-functioning database that has information on discoveries of historical objects.

The tables in the database are filled with data based on the file funn.txt.

- One or more classes that can communicate with the database.

- One or more classes that can hold data from the database. It is expected that inheritance is used for classes that include different types of findings as they have many common properties.

ATTENTION! Remember that the time is limited. If you are unable to finish Part 1 within a reasonable time, then you will have to move on to Part 2. In that case, keep the code you wrote for Part 1 so that it can be reviewed.

Part 2 – using the database:

You will create a small program that offers the user a menu with some menu options that deal with the historical objects that are now placed in the database (after part 1 has been completed). You can choose yourself whether you want two main methods (one for part 1 and one for part 2), or whether you want to comment out the code for part 1 when you develop part 2. Or you can choose to have part 1 and part 2 in the same program.

The program must offer the following functionality:

- See information about all found objects.

- See information on all found objects older than <year>. (The user specifies the year).

- Get information on the number of found objects registered. - Exit the program.

Remember to include the file that describes how you have handled exceptions, encapsulation and inheritance before zipping and delivering.

Appendix 1 – about the file funn.txt

Overall, the file contains information about three different entities: Person, Museum and Found Objects. There are three different types of Found Items: jewellery, weapons and coins.

The file starts with information about people:

Persons: -> Headings that show that the information now coming concerns persons.

49 -> How many people are listed in the file.

1 -> Id for person

Ingrid Halvorsen -> Name of person

91675235 -> Telephone number for person

Ingrid.Halvorsen@hotmail.com -> E-mail address of person

[More people]

Next comes information about museums:

Museums: -> Headings that show that the information that is now coming deals with museums.

10 -> How many museums are listed in the file.

1 -> Id for museum

Engerdal Museum -> Name of museum

Trysil -> The place of the museum

[More museums]

At the end of the file comes information about all the findings. It starts with a headline, and then information about the findings. There are three different types of finds: Jewelry, Weapons, and Coins. They have some common characteristics, and some characteristics that are different. The first find is a coin:

Findings: -> Headings that show that the information that is now coming deals with findings.

1 -> Id of the found item – in this case a coin.

66.79195598068002, 29.861744277190 -> Coordinates of the discovery site.

4 -> Id of the person who found the item.

2023-08-17 -> The date when the item was found.

1348 -> Assumed year for when the item was made.

7 -> Id of the museum where the object is now located. Not all of the objects are in the museum.

Coin -> What type of find item it is.

28 -> The diameter of the coin.

Silver -> The metal of the coin.

------- -> A separator showing that the information about the find item is complete.

[More findings]

Here is an example of a weapon that is not in a museum:

10 -> Id of the found object – in this case a weapon.

62.55779037017215, 29.66555626918067 -> Coordinates of the discovery site.

37 -> Id of the person who found the item.

2023-04-28 -> The date when the item was found.

1703 -> Assumed year for when the item was made.

There is no ID for the museum here. This weapon is therefore not in a museum.

Weapon -> What type of artifact it is.

Ax -> What type of weapon it is.

Bronze -> What kind of material the weapon is made of.

1606 -> The weight of the weapon (in grams).

------- -> A separator showing that the information about the find item is complete.

And here is an example of the last type of found object; jewelery:

12 -> Id for the found item – in this case a piece of jewellery.

63.3082538637922, 23.70062482964193 -> Coordinates of the discovery site.

32 -> Id of the person who found the object.

2024-01-09 -> The date when the item was found.

1567 -> Assumed year for when the object was made.

Here, there is no ID for the museum either. This piece of jewelery is therefore not in a museum.

Jewelry -> What type of found object it is.

Brooch -> What type of jewelery it is.

3234 -> Estimated value of the jewelery (in NOK).

119960192460.jpg -> filename for image of the jewellery.

--------> A separator showing that the information about the found object is complete