Conceptual Modeling

# Before Class

1. Watch the videos dealing with the UML and creating diagrams:
   1. What's UML and Why Do You Need It?

<https://youtu.be/8CBnAmYnwk0>

* 1. UML model:

<https://youtu.be/OmbZpJrlrbk>

1. Familiarize yourself with class diagrams. Consider when this type of diagram is used.
2. Learn about drawing a class diargram with draw.io. Watch the video:

‘Creating UML Class Diagrams and Objects Diagrams with Draw.io’:  
<https://youtu.be/dcsvl3YqAEk>

1. Using the draw.io application (<https://draw.io>), try to create a class diagram for a class describing your smartphone. Create at least 5 properties and 3 methods.

# During Class

1. Watch the video ‘UML Class Diagram Tutorial’:

<https://youtu.be/UI6lqHOVHic>

1. Answer the questions:
   1. What are the three sections of a single class diagram?
   2. How access modifiers are marked in a class diagram?
   3. How static fields and static methods are tagged in a class diagram?
2. Working in a group, create a class diagram for:
   1. Computer file
   2. Bank account
   3. Any object
3. Write a program that performs the following actions for the bank account:
   1. Deposit PLN 500
   2. Display balance
   3. Deposit PLN 200
   4. Display balance
   5. Withdraw PLN 300
   6. Display balance
4. If your account balance is less than the withdrawal amount, no withdrawal is possible. Include this condition in the program. When you try to withdraw, display a message.
5. In line with the bank's policy, you can withdraw no more than PLN 500 from your account at one time. Include these limitation in your program. Display a message when trying to withdraw a larger amount.
6. In the defined class, add the ability to display the three most recently performed transactions on the bank account. Define a 3-element array and store the 3 most recently performed transactions in it.

# After Class

1. Write a program for the class Computer File, defined in the “During Class” section.
2. The Best Books publishing house releases crime, drama, fantasy, and science fiction books. Consider what attributes can be used to describe the books. Then, create a class diagram containing books attributes. Finally, define a class based on the created class diagram.
3. Competitors are judged by five judges during the competition. Each judge can score 1, 2, 3, 4 or 5 points. Then, the highest score and the lowest score are thrown out. The arithmetic mean of the remaining three scores is calculated and this is the competitor's final result that is displayed. Create a class diagram for the competition scoring system. Define a class and write a program that calculates the final results for three players.
4. Create a class diagram for a shopping list. You can use, for example, an ArrayList as the data structure for storing products. Then, based on the class diagram, define a class. Finally, create an object representing a shopping list and perform the following actions:
   1. Display the list of products (should be empty)
   2. Display information about the number of products to be purchased
   3. Add three products to your shopping list
   4. Display the list of products
   5. Display information about the number of products to be purchased
   6. Add one product to your shopping list
   7. Display the list of products
   8. Display information about the number of products to be purchased
5. Add to the list of products the ability to enter product names from the keyboard. Then, run the program, create a shopping list and display a list of products to buy.
6. The sales system consists of three categories of objects: a seller, a customer and a purchased product. The seller can be a company while the customer is a person. Create class diagrams for each of the object categories. Then, based on the created class diagrams, define corresponding classes. Save all three diagrams in one file. Finally, define classes based on the created class diagrams and write a program that creates objects, assign them attribute values, and call the defined methods.
7. For any object of your choice, create a class diagram. When creating a diagram, take into account the principles of encapsulation. Then, based on the diagram you have created, define a class. Using a defined class, write a program that creates two objects, assign them attribute values, and call the defined methods.