Interfaces

# Before Class

1. Familiarise yourself with the concept of interfaces.
   1. <https://www.w3schools.com/java/java_interface.asp>
   2. <https://www.javatpoint.com/interface-in-java>
2. What is the difference between an interface and an abstract class.

* Interface ma tylko abstrakcyjne metody a klasa i takie i takie
* Klasa może wykorzystywać wiele interfaceow, a może extendować tylko jedną klasę
* Interface nie ma konstruktora, klasa może mieć
* Interface implementuje klasa, a abstrakcyjna klasa jest extendowana przez subclasse

1. How to create interfaces in programming language.

Public interface Klasa {}

Public class Klasa1 implements Klasa

1. What attributes and methods can interfaces contain.

Statyczne, default, abstrakcyjne

1. How an interface is represented in a UML class diagram.

<<interface>>

# During Class

1. The basic functionalities of the TV set are switching it on and off, changing the channel and adjusting the volume level. Create a CanOnOff interface that contains methods to turn on and off the TV set: on(), off().
2. A TV class describes TV sets. Each TV can be turned on or off. Define the class with the appropriate attribute. Add a method that displays TV status.
3. In the TV class, implement the CanOnOff interface. Define methods by which the TV can be turned on and off. Then write a program that:
   1. Turns on TV
   2. Displays TV status
   3. Turns off TV
   4. Displays TV status
4. In addition to switching on and off, TV sets also allow you to change the number of the displayed channel. Create a CanChangeChannel interface which contains the methods for changing the channel: channelUp(), channelDown(), setChannel(channelNo).
5. In the TV class, add an attribute that describes the channel number. Then implement the CanChangeChannel interface. Define methods in the class for changing the channel. Take into account that:
   1. The allowed channel number is 1 through 99.
   2. The channel can only be changed when the TV is on.
   3. The default channel number after turning on the TV is 1.
   4. TV status displays channel number only when the TV is on.

Then write a program that:

* 1. Turns on the TV
  2. Displays TV status (whether TV is on and channel number)
  3. Changes channel number to the next one.
  4. Changes channel number to 7.
  5. Changes channel number to 142.
  6. Displays TV status.
  7. Changes channel number to the previous one.
  8. Displays TV status.
  9. Turns off the TV.
  10. Displays TV status.

1. An important functionality of each TV set is the volume control. Create a CanChangeVolume interface that contains the methods for changing the volume level: volumeUp(), volumeDown().
2. In the TV class, add an attribute that describes the volume level. Then implement the CanChangeVolume interface. Define in the class the methods for adjusting the TV volume level. Take into account that:
   1. The valid range for the volume level is 1 to 10.
   2. The default volume level after turning on the TV is 1.
   3. The volume level can only be adjusted when the TV is on.

Then write a program that:

* 1. Turns on the TV.
  2. Displays TV status (whether TV is on, channel number, volume level)
  3. Changes channel number to 15.
  4. Sets the volume level to 7.
  5. Displays TV status.
  6. Turns off the TV.
  7. Displays TV status.

# After Class

1. In a UML class diagram, show the TV class definition with all created interfaces. Pay attention to the relationships between the class and interfaces.
2. You can assign the selected TV station to a channel number on each TV set. Complete the TV class with the list of TV stations (list of channel names). You can use an array with a size equals to the maximum channel number. Assign TV station names for the first ten array elements. Then write a program that:
   1. Displays a list of TV stations with their channel numbers.
   2. Displays the TV status along with the name of the TV station (if available).