Classes And Objects

# Before Class

1. Watch the film:
   1. Programming paradigms – wzorce: <https://youtu.be/Wt4FPjkCNaU>
   2. Programming Paradigms in 6 Minutes: <https://youtu.be/B1p5OlO5tWg>

Then list the main features of the following programming paradigms (you can search the Internet for some details):

* 1. Imperative programming - Programowanie imperatywne – paradygmat programowania, który opisuje proces wykonywania jako sekwencję instrukcji zmieniających stan programu. Programy imperatywne składają się z ciągu komend do wykonania przez komputer. Rozszerzeniem (w sensie wbudowanych funkcji) i rodzajem (w sensie paradygmatu) programowania imperatywnego jest programowanie proceduralne.
  2. Procedural programming - paradygmat programowania zalecający dzielenie kodu na procedury, czyli fragmenty wykonujące ściśle określone operacje. Procedury nie powinny korzystać ze zmiennych globalnych (w miarę możliwości), lecz pobierać i przekazywać wszystkie dane (czy też wskaźniki do nich) jako parametry wywołania.
  3. Structured programming - jeden z paradygmatów programowania, w którym celem jest konstruowanie programu zmierzające do osiągnięcia takiej jego struktury, aby stanowiła ona odzwierciedlenie struktury rozwiązywanego problemu.
     + - 1. Iteracja
         2. Wybór
         3. Sekwencja
         4. Pętla
         5. Podprogram
         6. Instrukcja

1. Familiarise yourself with primitive data types available in Java. How is the type of a variable determined in Java.
   1. int, byte, short, long, float, double, boolean and char.
   2. These aren't considered objects and represent raw values.
   3. <https://www.baeldung.com/java-primitives>

piszemy sb typ przed zmienna

* 1. Obraz zawierający stół

     Opis wygenerowany automatycznie

1. Familiarise yourself with the basic operators available in Java.
   1. \* Multiplication
   2. / Division
   3. % Modulus (returns reminder)
   4. ++ increment
   5. – decrement
   6. == equal to
   7. ! = not equal to
   8. > < >= <=
   9. && and
   10. || or
   11. ! logical not
2. Familiarise yourself with the CamelCase – a naming convention for variables, constants, methods, and classes in Java.
   1. Variables - camelCase
   2. Constants - CM\_PER\_INCH snake\_case
   3. Methods – camelCase
   4. Classes - PascalCase

# During Class

1. Watch the film:
   1. What is object-oriented language: <https://youtu.be/SS-9y0H3Si8>
   2. Object Oriented Programming: <https://youtu.be/0yEBBCfaIZk>

Then answer the questions:

* 1. What is a class
  2. What is an object
  3. What are object components

1. Specify at least five attributes and three behaviors for objects that represent students.
2. In the Java programming language, define a Student class. Pay attention to the class name and the file where the class is defined.
3. In the Student class, add name and age fields to represent the object's attributes. Pay attention to the necessity to specify the attribute types.
4. Create two Student class objects. Assign values to the object fields. What are the object field properties.
5. In the Student class, add methods (method headers only) that represent the behaviors of the object:
   1. sayHello()
   2. displayName()
   3. displayAge()

Then enter the statements in the methods to display the appropriate information.

1. Create three Student class objects. Assign values to the object fields. Then call the methods.
2. Add the following fields and methods to the Student class to represent object attributes and behaviors:
   1. Object attributes: (1) student ID card, (2) whether the student ID card is valid or not, (3) semester number, (4) average grade
   2. Object bahaviors: (1) display the student’s name, semester number and average grade, (2) change the status of student ID card (valid/invalid), (3) display the student’s name, ID card number and whether the ID card is valid

Then create two Student class objects, assign them some values and call methods.

# After Class

1. Specify at least five attributes and three behaviors for the objects:
   1. Smartphone
   2. Bank account
2. Define a class for describing books. Then, define fields and methods in the class that represent the attributes and behaviors of the book objects. Finally, create two books, assign them attribute values, and call defined methods.
3. Define a class for describing lamps. Then define the following object attributes and behaviors:
   1. attributes: whether a lamp is on
   2. behaviors: (1) switch on the lamp, (2) switch off the lamp, (3) display information whether the lamp is on or off

After that, create two lamps, switch on the first one and switch off the second one. Display information whether the lamps are on or off.

1. Define a class for describing rectangles. Then create two objects that represents rectangles with dimensions of 3x4 and 2x7 respectively. Display their dimensions, perimeters and surface areas.
2. Design a simple counter. The initial value of the counter is 0. The counter can be increased or decreased by one, and increased and decreased by 10. It is possible to reset the counter to its initial value. Then define a class for counters. Create two counters. Set the first counter to 23 and the second one to -47.