Third practice – object manipulation

At this laboratory practice you will try to work with object in this manner:

- Object creation
- Assign attributes to object
- Constructor creation
- Dynamic object creation
- Class usage and object creation through class
- Access to attributes and methos of another object

Task 1:

Create three JavaScript files. First file is responsible for object creation and loop maintenance. In second file, you will create class *Player*. In this class create attributes:

- Strength
- Health
- Armor

For object and attributes initialization use random value in range <50-200>. Class *Player* has methods:

- Defense – based on *Armor* attribute will decrease *Health* or block attack,
- Attack – attack another object.

In third file, you will create class *Enemy* with same specification as *Player* class. One difference is in *attack* method. This method remains undefined.

Task 2:

Create two another two files – one with class *Wrathful Bunny* and second with class *Pinky Ass Destroyer*.

A) Inheritance

Classes *Wrathful Bunny* a *Pinky Ass Destroyer* will inherit methods and attributes from class *Enemy*.

B) Polymorphism

To classes *Wrathful Bunny* a *Pinky Ass Destroyer* define different behavior for *attack* method.

C) Encapsulation

For accessing attributes *Strength, Health and Armor* create get and set methods in classes *Player, Wrathful Bunny* a *Pinky Ass Destroyer*.

D) Object interaction

Object interaction is simulated in loop. In loop, enemies will attack at player. Loop will end when these conditions are met:

- 100 attacks happen between *Player* and enemies,
- Player has 0 or less health,
- All enemies have 0 health.

Task 3:

To classes *Player* and *Enemy* add attributes *position* and *attack range*. *Position* attribute is nested object with *X* and *Y* attributes. *X* and *Y* are in range 0 – 100. *Player* and *Enemy* classes extend with method *Jump*. With method *Jump*, will objects randomly jumps in any direction (max 5 point in *X* and *Y* parameter, but objects need to stay at "map"). *Player* and enemies can attack each other only in *attack range*. For attributes *position* and *attack range* create get and set methods.

RECOMMENDATION TO BACK-UP JAVASCRIPT PROGRAM. AT NEXT LABORATORY PRACTICES YOU WILL EXTENDS THIS PROGRAM WITH NEW FUNCIONALITIES.