Sixth practice – Observer

At this laboratory practice you will try to handle implement Observer design pattern.

For work at this practice, use already programed tasks from third and fourth laboratory practices.

Task 1:

This task is focused on work with Observer design pattern. Modify you program with Observer design patter in this way:

- *Player* is subject part in this program. Modify player with Observer array addition. Add method *notify*. This method *notifyObservers* will notifies all Observers in the Observer array.
- *Enemies* are the Observers. Observers have the method *notify*. This method is called by the subject in method *notifyObservers*.

Extension A – knowledge repetition → class creation and inheritance:

As an extension, you will add new enemy class. This class will inherit from *Enemy* class. You can name new class with your imagination. Then add new type of enemy at the map. New type of enemy will chase *Player* and will be implemented as Observer.

Extension B – knowledge repetition → polymorphism:

Modify all enemy classes (*Pinky Ass Deystroyer*, *Wrathful Bunny* and new created class) with polymorphed method *chase*. Each class will implement own style of *Player* chasing.

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