Dna Project.

Main: runs the cli.

UTIL:

CLI:

The cli serves as an observable, while the controller serves as an observer.

When a user enters an input – the cli notifies the controller, and sends him the input passed from the user.

Controller:

Checks while the command is a cmd command or a batch commands, and runs the appropriate command.

In case of a cmd command:

Cmd:

Holds as a variable a CommandImplementation object(a Factor – Design Pattern).

When receiving a command from the user, the CommandImplementation creates an object of the specific command, and executes it.

In case of batch command:

Batch:

The controller creates a BatchObserver object, and runs it.

BatchObserver:

The batchobserver serves an observable, while all the commands are used as observers.

The batchObserver creates a batch object, while, on each command the batch receives,

The batchObserver adds the command to his command list.

The batch run command, calls the batch observer notify function, and runs the commands stored in the command list.

Batch:

Receives from the user a batch command, and adds it to the command list,

When ending the batch mode, it saves the batch name and batchobserver in the data base.

BatchImplementation:

A factor (design pattern) for the batch commands.

Commands:

Every command is under a different category.

CommandImplementation:

When receiving a command, the commandImplementation send the command to the right category factory.

Every category contains a ‘category’_Implementation class, a ‘category’_command class, and the commands

‘category’Implementation:
A factory(design – pattern) who creates the command.

‘category’commans:

A class where all the command inheritors from.

DNA:

Data Base:

Holds the data the program needs to store through the run of it.

Dna:

A class designed to add the dna_sequences and the relevant data to the data base.

Dna_sequence:

A class who serves as a dna, and implements the relevant functions.