## HARDWARE RELATED CLASSES

#### IWATER

IWATER is the interface class containing the virtual functions relating which is implemented in the hardware class.

|  |  |
| --- | --- |
| IWATER | |
| Functions | Description |
| *virtual void SetWaterLevel(int level) = 0;* | The purpose of this function will be to set the appropriate water lever for the washing process. It takes one parameter for the level that we want to check. |
| *virtual boolean CheckWaterLevel(int level) = 0;* | The purpose of this function will be to check if the washing machine has reached the desired water level. It takes one parameter for the level that we want to check. |
| *virtual void SinkWater() = 0;* | The purpose of this function will be to |
| Remarks: This interface is complete. All the functions are implemented.  To discuss: I just wonder if we need any more functions for this interface or are those 3 enough?  (Please describe here any problems, remarks, current states of your functions) | |

#### IMOTOR

IMOTOR is the interface class containing the virtual functions relating which is implemented in the hardware class.

|  |  |
| --- | --- |
| IMOTOR | |
| Functions | Description |
| *virtual void SetDirection(char dir) = 0;* | The purpose of this function will be to indicate the direction of the washing machine. It takes one parameter to allow us to indicate the direction that we want to set. |
| *virtual void SetSpeed(int level) = 0;* | The purpose of this function will be to indicate the speed of the washing machine. It takes one parameter to allow us to indicate the speed that we want to set. |
| *virtual void StopMotor() = 0;* | The purpose of this function will be to stop the motor. |
| *virtual void StartMotor() = 0* | The purpose of this function will be to start the motor at an initial state assuming that all the prerequisites are handled properly within the IProgram interface. |
| *virtual void CheckLoadingLevel(int level) = 0;* | The purpose of this function will be to check le loading level of the washing machine. It takes one parameter for the level that we want to check. |
| Remarks: This interface is complete. All the functions are implemented.  To discuss: To start the motor can we just set the initial speed, or we also should consider some other states? Should we take care of the prerequisites in this function of in the IProgram interface?  (Please describe here any problems, remarks, current states of your functions) | |