### **Question {abstract}**

- + Header of the question : string -> (automatic property)
- + Body of the question : string -> (automatic property)

# Mark : double # rightAnswerld: int

- + RightAnswerld: int -> (Full Property)
- + Mark: double -> (Full Property)
- + NumberOfChoices int -> (Full Property)

#### Exam {abstract}

- # TimeOfExam: int
- # NumberofQuestions: int
- + McqQuestions : List<MCQ\_Questions>
- + TOFQuestions : List<TrueOrFalse Questions>
- + fullMark : double
- + TimeOfExam: int
- + NumberOfQuestions : int
- +FullMark:double

#### **Answer**

- id: int
- #Text: string
- + Answer(int , string )(type)
- + Id: int
- + ToString(): string

## **Helper{static}**

- + ReadInt(int , int ): int
- +ReadString():string
- + ReadDouble(): double
- + MainWindow(): void
- +Delay(): void

# HelperExam{static}

- + CreateObjectForPracticalExams(int, int
- ): PracticalExam
- +CreateObjectForFinalExams(int , int ):FinalExam
- + ReadInfoForExams(int): (int , int )
- + GetAllFinalExams(int id): void
- +GetAllPracticalExams(int): void
- + CreateExam(): (Exam ,int )
- + ShowExamImediatelyTaken

(Exam , double ,int[] , int[], DateTime , DateTime )

+ TakeExam(Exam): void

# HelperQuestion{static}

- + ReadInfoAndCreateObject
- ForTrueOrFalse Question()
- :TrueOrFalse Question
- +ReadInfoAndCreateObjectForMCQ\_Question()
  :MCQ Question
- + GetAnswerByld(Answer[],int): string

#### **TrueOrFalse Question**

- + TrueOrFalse Question()
- + TrueOrFalse Question(string, string, double, int)
- + ToString():string

#### MCQ Question

- + answers : Answer[]
- + MCQ\_Question()
- +MCQ Question

(string,string,double,int,Answer[],int)

- + printAnswers():string
- + ToString(): string

#### **PracticalExam**

- + PracticalExam()
- + PracticalExam(int ,int )
- + ToString():string

#### **FinalExam**

- + FinalExam()
- + FinalExam(int ,int )
- + ToString():string

### Subject

- # Id: int
- + Name: string
- + finalExamsList:List<Exam>
- + practicalExamsList: List<Exam>
- + Subject(int , string ):
- + Id : int
- + ToString(): string

#### <u>HelperSubject{static}</u>

- + subjectsList: List<Subject>
- + createSubject(): void
- + GetAllSubjects(): void
- + FoundId(int) : bool
- + ReadIdSubject(int , int ):int
- + CreateExamForSpecificSubject(int ):void
- + <u>TakePracticalExamForSpecificSubject(int id)</u>: void
- +<u>TakeFinalExamForSpecificSubject(int id)</u>: void