

tor<Employee *> ng, int, unsigned long int, string, unsigned long int, unsigned long int) (Employee*) :void yee (string) :bool Interface DataTypeT : class library: Library Ex cepti on Interface() msg :string menu() :void value :DataTypeT displayMenu():void clearScreen():void Exception(string, DataTypeT) displayContainer(vector<T*>) :void {query getError() :string {query} displayHeader(string):void getMessage() :string {query} confirm Operation (string &): bool getValue() :DataTypeT {query getInput():int readInputString(string) :string pressAnyKey():void is_NON_ASCII_Letter(int&):bool is_All_ASCII_Letter(string&):bool ~Interface() Library s :vector<Book*> lib rary entBorrows:vector<Current*> iousBorrows:vector<Previous*> ers:vector<Reader*> loyees :vector<Employee*> rvisors: vector<Supervisor*> erID :un signed long int ary() rary () ooks() :vector<Book*> {query} currentBorrows():vector<Current*> {query} reviousBorrows():vector<Previous*> {query} eaders() :vector<Reader*> {query} mployees():vector<Employee*> {query} upervisors(): vector<Supervisor*> {query} ooks(vector<Book*>) :void urrentBorrows(vector<Current*>) :void reviousBorrows(vector<Previous*>) :void eaders(vector<Reader*>) :void mployees(vector<Employee*>) :void upervisors(vector<Supervisor*>):void Book(Book*) :void CurrentBorrow(Current*) :void PreviousBorrow(Previous*):void

Superv isor

Reader(Reader*) :void Employee(Employee*) :void Supervisor(Supervisor*) :void

ove CurrentBorrow(Current*): bool ove PreviousBorrow(Previous*): bool ove Reader(Reader*): bool ove Employee (Employee*): bool ove Supervisor(Supervisor*): bool

Readers() :void Employees() :void Supervisors() :void oveBook(Book*) :bool