****

**COMP 3140 - Object Oriented Design & Prog**

**Project Stage 2:Interface Files Document**

**Team Member:**

**ZiCheng Ren T00548429**

**JiaRen Li T00573084**

**Lukas Bowden T00040951 11/23/2020**

* **Class Person**

**Data members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Data members** | **Type** | **Access** | **Definition** |
| **name** | **string** | **protected** | **Person’s name** |
| **gender** | **string** | **protected** | **Person’s gender** |
| **age** | **int** | **protected** | **Person’s age** |

**Constructor and Desctructor summary:**

|  |  |
| --- | --- |
|  | **Definition** |
| **Person(const string& newName,const int& newAge,const string& newGender);** | **Parameter constructor** |
| **Person()** | **Default constructor** |
| **virtual~Perosn** | **Destructor** |

**Member Fucntions:**

|  |  |  |
| --- | --- | --- |
|  | **Definition** | **Access** |
| **void setName(string& newName);** | **Mutators(set value of name)** | **public** |
| **void setAge(int& newAge);** | **Mutators(set value of age)** | **public** |
| **void setGender(string& newGender);** | **Mutators(set value of gender)** | **public** |
| **int getName() const;** | **Accessors(get name)** | **public** |
| **int getAge() const;** | **Accessors(get age)** | **public** |
| **int getGender() const** | **Accessors(get gender)** | **public** |
| **Virtual void printInfo() const** | **printInfo() function that print all member data** | **public** |

* **Class Tenant : public Person**

**Data members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Data members** | **Type** | **Access** | **Definition** |
| **job** | **string** | **private** | **Tenant’s job** |
| **unitNumber** | **int** | **private** | **Tenant’s unit Number** |
| **monthly\_rent** | **double** | **private** | **Tenant’s Monthly\_rent** |
| **movingIn\_Date** | **Date** | **private** | **Tenant’s moving in date** |
| **payment\_status** | **Char[6]** | **rivate** | **Tenant’s payment status for six month** |

**Constructor and Desctructor summary:**

|  |  |
| --- | --- |
|  | **Definition** |
| **Tenant(const string& newName,const int& newAge, const string& newGender,const string& newJob, const int& newUnitNumber, Date newMoveIn\_Day, const double& newMonthly\_rent, char newPayment\_status[6]);** | **Parameter constructor** |
| **Tenant();** | **Default constructor** |
| **virtual ~Tenant();** | **Destructor** |

**Member Fucntions:**

|  |  |  |
| --- | --- | --- |
|  | **Definition** | **Access** |
| **void setJob(const string& newJob);** | **Mutators(set value of job)** | **public** |
| **void setMovingIn\_Date(const Date& date);** | **Mutators(set value of movingIn\_Date)** | **public** |
| **void setUnitNumber(const int& newUnitNumber);** | **Mutators(set value of unitNumber)** | **public** |
| **void setMonthlyRent(const double& newMonthly\_rent);** | **Mutators(set value of monthly\_rent)** | **public** |
| **void setPaymentStatus(char newPayment\_status[6]);** | **Mutators(set value of payment\_status)** | **public** |
| **string getJob() const;** | **Accessors(get job)** | **public** |
| **int getUnitNumber() const;** | **Accessors(get unitNumber)** | **public** |
| **Date getmovingIn\_Date() const;** | **Accessors(get movingIn\_Date)** | **public** |
| **double getMonthlyRent() const;** | **Accessors(get monthly\_rent)** | **public** |
| **char\* getPaymentStatus();** | **Accessors(get payment\_status)** | **public** |
| **virtual void printInfo() const;** | **Overridden printInfo() function from base class** | **public** |

* **Date Class**

**Data members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Data members** | **Type** | **Access** | **Definition** |
| **day** | **string** | **private** | **Current day** |
| **month** | **int** | **private** | **Current month** |
| **year** | **double** | **private** | **current year** |

**Constructor and Desctructor summary:**

|  |  |
| --- | --- |
|  | **Definition** |
| **Date(const int& newDay, const int& newMonth, const int& newYear);** | **Parameter constructor** |
| **Date();** | **Default constructor** |
| **Date(const Date& date);** | **Copy Constructor** |
| **~Date();** | **Destructors** |

**Member Fucntions:**

|  |  |  |
| --- | --- | --- |
|  | **Definition** | **Access** |
| **void setDay(const int& newDay);** | **Mutators(set value of day)** | **public** |
| **void setMonth(const int& newMonth);** | **Mutators(set value of month)** | **public** |
| **void setYear(const int& newYear);** | **Mutators(set value of year)** | **public** |
| **int getDay() const;** | **Accessor(get value of day)** | **public** |
| **int getMonth() const;** | **Accessor(get value of month)** | **public** |
| **int getYear() const;** | **Accessors(get value of year)** | **public** |
| **String getDate() const;** | **Return full date information** | **public** |
| **string convertMonth(int m) const;** | **Converts a numbered month into a worded one. i.e. January, Febuary, March, April... etc** | **private** |
| **bool checkValidDate();** | **Checks if the given numbers is a valid date by checking if days is between 1 - 31, Months is 1 - 12, and Year is correct.** | **private** |

* **Manager Class: Public Person**

**Data members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Data members** | **Type** | **Access** | **Definition** |
| **salary** | **double** | **private** | **Manager’s salary** |
| **bouns** | **double** | **private** | **Manager’s bonus** |
| **monthly\_expense** | **double[6]** | **private** | **Manager’s expense for each six month** |
| **hired\_date** | **Date** | **private** | **Manager’s hired date** |

**Constructor and Desctructor summary:**

|  |  |
| --- | --- |
|  | **Definition** |
| **Manager(const string& newName,const int& newAge,const string& newGender,const Date& newHired\_date, const double& newSalary,const double& newBouns,double newMonthly\_expense[6]);** | **Parameter constructor** |
| **Manager();** | **Default constructor** |
| **virtual ~Manager();** | **Destructor** |

**Member Fucntions:**

|  |  |  |
| --- | --- | --- |
|  | **Definition** | **Access** |
| **void setHired\_Date(const Date& date);** | **Mutators(set value of hired day)** | **public** |
| **void setSalary(const double& newSalary);** | **Mutators(set value of salary)** | **public** |
| **void setBouns(const double& newBouns);** | **Mutators(set value of bonus)** | **public** |
| **Void setMonthly\_expense(double newMonthly\_expense[6]);** | **Mutators(set value of monthly\_expense )** | **public** |
| **Date getHired\_Date() const;** | **Accessor** | **public** |
| **double getSalary() const;** | **Accessor** | **public** |
| **double getBouns() const;** | **Accessor** | **public** |
| **double\*getMonthly\_expense();** | **Accessor** | **public** |
| **virtual void printInfo() const;** | **Overridden printInfo() function from base class** | **public** |

* **Management Class**

**Data members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Data members** | **Type** | **Access** | **Definition** |
| **fileName** | **string** | **protected** | **File’s name** |
| **totalManager\_Expense** | **double** | **protected** | **Total manager expense for six month** |
| **totalRent** | **double** | **protected** | **Total rent that collected for six month** |
| **tenantArr** | **Tenant[100]** | **protected** | **Store information of tenant from Excel** |
| **managerArr** | **Manager[2]** | **protected** | **Store information of Manager from Excel** |
| **totalSalarayBonus\_Expense** | **double** | **protected** | **Total salary and bonus for six month** |

**Constructor and Desctructor summary:**

|  |  |
| --- | --- |
|  | **Definition** |
| **Management(const string& newFileName);** | **Parameter constructor** |
| **Management();** | **Default constructor** |
| **virtual ~Manager();** | **Destructor** |

**Member Fucntions:**

|  |  |  |
| --- | --- | --- |
|  | **Definition** | **Access** |
| **void profile\_tenant();** | **Print out all tenants' profile** | **public** |
| **void profile\_manager();** | **Print out all Manager' profile** | **public** |
| **void setFileName(const string& newFileName);** | **Mutators** | **public** |
| **string getFileName();** | **Accessors** | **public** |

* **NewManagement class: public Management**

**Data members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Data members** | **Type** | **Access** | **Definition** |
| **tenantFileName** | **string** | **private** | **File name of tenant** |
| **managerFileName** | **string** | **private** | **File name of manager** |
| **month\_name** | **string[6]** | **private** | **Each month name(read from file)** |
| **total\_missRent** | **double** | **private** | **Total missing rent in six month** |
| **totalIncome** | **double** | **private** | **Total income in six month** |

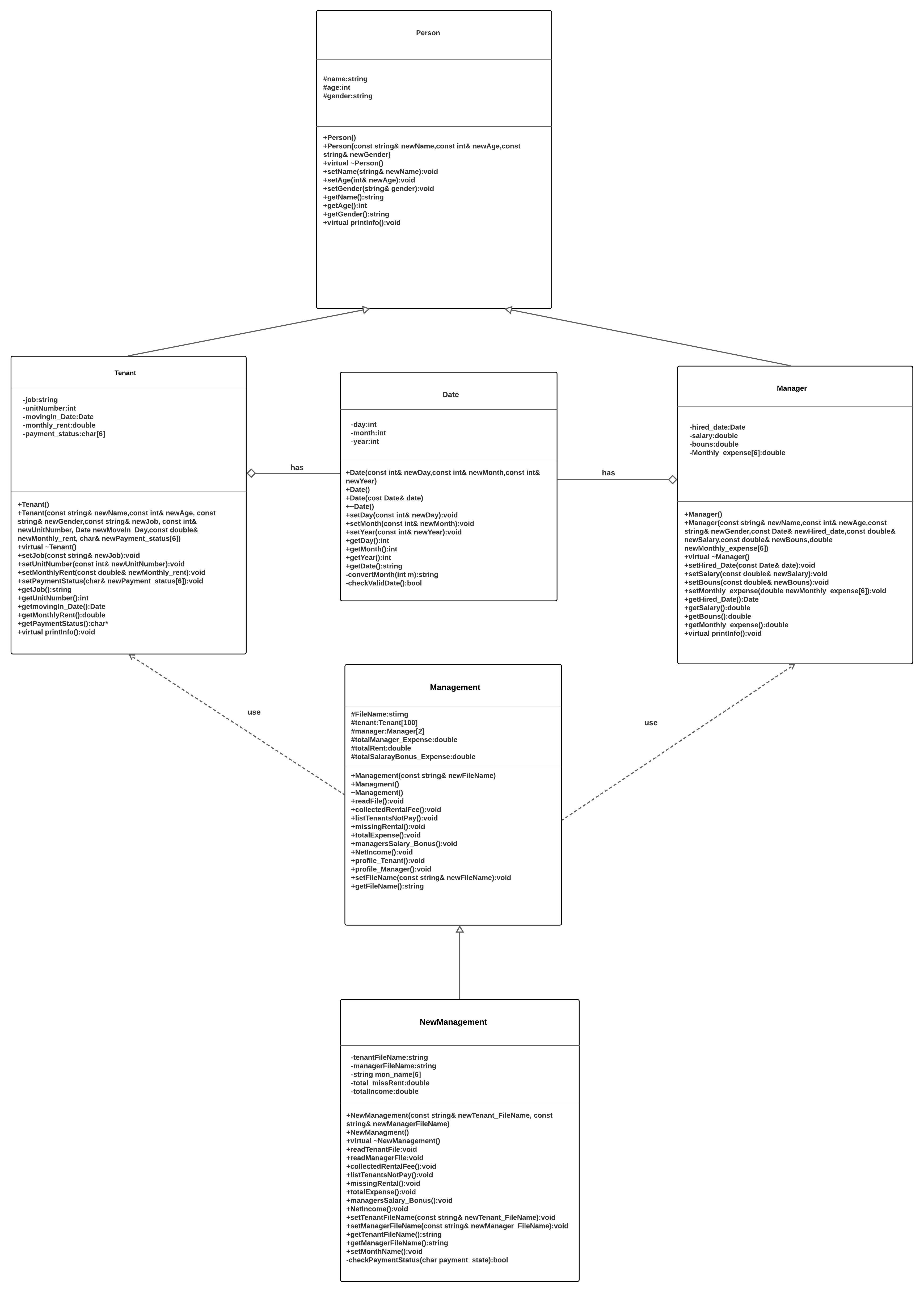
**Constructor and Desctructor summary:**

|  |  |
| --- | --- |
|  | **Definition** |
| **NewManagement(const string& newTenant\_FileName,const string& newManagerFileName);** | **Parameter constructor** |
| **NewManagement();** | **Default constructor** |
| **virtual ~NewManagement();** | **Destructor** |

**Member Fucntions:**

|  |  |  |
| --- | --- | --- |
|  | **Definition** | **Access** |
| **void readTenantFile();** | **Takes in an input file based on the tenantFileName variable, scans through it and put data into tenantArr[100] with any tenats.** | **public** |
| **void readManagerFile();** | **Takes in an input file based on the managerFileName variable, scans through it and put data into managerArr[2] with any tenats.** | **public** |
| **void collectRentalFee();** | **Scans through tenantArr[100] and checks their payment status, if it's true it checks their rental fee and adds it to a total for the month as well as a total for 6 months. Does this process 6 times for each month.** | **public** |
| **void listTenantsNotPay();** | **Scans through the tenantArr[100] and lists any tenants, along with their current index, with payment status set to false(Not Pay).** | **public** |
| **void missingRental();** | **This function to determine who hasn't paid their rent, then adds up the total missing rent for the month then continues to add the total of missing rent for the past 6 months before printing it out.** | **public** |
| **void totalExpense();** | **Scans through the managerArr[100] and grabs the total expenses data held in every manager then adds them together and displays the total. It continues to do this for every month as well as the past 6 months.** | **public** |
| **void managerSalary\_bouns();** | **Scans through the managerArr[100] and displays the total salary and bonus of every month as well as 6 months. It does this for every individual manager.** | **public** |
| **void netIncome();** | **The function that caculate the total Net income for each month and for six months.**  **NetIncome = (total rent collection)- (managers expense)-(salary and bonus expense)** | **public** |
| **void setTenantFileName(const string& newTenant\_FileName);** | **Mutators** | **public** |
| **void setManagerFileName(const string& newManager\_FileName);** | **Mutators** | **public** |
| **string getTenantFileName();** | **Accessors** | **public** |
| **string getManagerFileName();** | **Accessors** | **public** |
| **bool checkPaymentStatus(char payment\_state);** | **Check the tenant's payment status** | **private** |

* **UML for Stage 2**

****