Object-oriented programming task list

The second semester of 2019-2020 academic year

**Task 1 （25 points）**

Design the program according to the following requirements, draw the corresponding UML diagram, program flow chart, and list the complete program code.

(1) Read in a sequence of Numbers separated by commas.

(2) Design a method to complete sequence inversion.

(3) Design a method to complete the sequencing of the sequence.

(4) Design a method to complete the calculation of the average value of the sequence.

(5) Design an algorithm to find the number closest to the average value, and display its position in the sequence.

**Task 2（25 points）**

**(A)** Design a binary *ArithmeticExpression* class, which includes:

(1) Two double private data members named x and y, representing left and right operands respectively.

(2) A double private data member named *operator*, representing the operator.

(3) A constructor for creating objects based on concrete binary expressions. Where the operator is qualified as "+, -, \*, \" and the constructor throws an exception if the argument does not qualify.

(4) Accessor method of three members.

(5) *getResult()* method to evaluate the expression’s value. This method throws an exception when it encounters a division operation with a zero divisor.

**(B)** Write the main program, if the disk file named *out.txt* does not exist, then create the file, and create 100 binary arithmetic expression objects with random Numbers, and write the expression and operation results into the file; If the file exists, all expressions are read from the file and displayed in line.

**Task 3 (25 points)**

**(A)** Design a class called *Policema*n, Which includes the following information:

(1) Three private property: police number, name and salary.

(2) A private property which is date type, and represents the date of work.

(3) The constructor to create the object according to the specified number, name and salary (the time when the object is created is the date of starting work).

(4) Accessors for all properties.

(5) Modifier of salary property.

(6) Modifier for date of work property.

(7) Override the inherited *toString()* method and output "Number + name + years of service + salary".

**(B)** Design an interface named *Employee* that contains an abstract method called *Raise ()*

**(C)** Design a class named *ArmedPolice*, inherit the police class and implement the employee interface, and include the following information:

(1) A String type property, representing the name of the army to which it belongs.

(2) A String type property , representing the rank.

(3) The constructor to create the object according to the specified number, name, salary, army and rank (the time to create the object is the date to start work).

(4) Realize the abstract method *raise* to increase the salary attribute by 30%.

(5) Override the inherited *toString()* method and output "Number + name + years of service + salary + army + rank".

**(D)** Draw the UML diagram of the above class structure.

**(E)** Compile the test program and create a police officer with the police number 10034, the name is "Jackie chan", the salary is 3000, the army is the 7th Brigade of the Armed Police, the rank is the third Police supervisor. Set the working time as April 1, 2004.Assume a raise every two years, modify his current salary according to the current system time, and output his personal information.

**Task 4 (25 points)**

Combining the C language and Java language we have learned, Please summarize the differences between object-oriented programming idea and process-oriented programming idea. (You can use examples to illustrate your idea, and the total number of words between 300 and 500 words. Warning: If you direct copy of the content on Baidu then you will get 0 points)