Workshop 8

5 февраля 2019 г.

Consider the tasks from the Workshop 6. Remaster the solution using the OOP approach.

1. Define a class named `Employee` which describes an employee. The fields are:
   a. Name
   b. Date of Birth (in dd.mm.yyyy format)
   c. Array of wages
   d. An auxiliary variable for storing cache of calculated total wages. Use -1, if no total wage is calculated.
   e. (Additional fields if needed).

   All fields are private. Create all necessary methods to provide a public interface to these fields. Respect constancy.

2. Create methods to calculate a total and average salary with caching the calculated sum.
3. Overload the `operator<<` for inputting data of an individual employee.
4. Overload the `operator>>` for outputting data of an individual employee. Output wages as a vertical table, as in the following example:

1	12.50
2	13.10
3	1.15

5. Create a method for inputting data of several employees using the new `operator<<` approach.
6. Create a method for outputting data of several employees using the new `operator>>` approach.
7. Create a method which sorts an array of employees by name, then by age, then by *average* salary. Use `std::sort method()`. Create a predicate for comparing two employee objects if needed.
8. Create a `main()` method which brings the things together:
   a. inputs a collection of employees;
   b. outputs the initial collection;
   c. creates a copy of a collection of employees, sorts it and outputs a sorted collection.

   Draw a UML Sequence Diagram depicting the call flow of the developed methods and free functions.