**Ministry of education and science of the Kyrgyz Republic**

**Kyrgyz State Technical University named after I.Razzakov**

**Faculty of Information Technologies**

**Department of Software of Computer Systems**

**Major: 710400 «Software Engineering»**

Report

Discipline: «**Object-Oriented Design**»

Software requirements

Done by**:** student of the SE(eng)- 1- 21

Rami Alshalabi

Checked by: Musabaev E.B.

Bishkek 2024

**Task1**

**Class Timer**

class Timer  
{  
private:  
 int initialTime;  
 int remainingTime;

**Functional requirements**

1. **Initialization:**
   * The Timer class should allow initialization with an initial time value specified in seconds.
   * It should also support initialization using a string representing time in seconds.
   * Initialization should be possible with minutes and seconds separately.
2. **Countdown Timer Function:**
   * The Timer class should have a **run()** function that initiates the countdown.
   * During the countdown, the Timer should display the remaining time in seconds at regular intervals (e.g., every second).
   * The countdown should pause for one second between updates.
3. **Timer Completion:**
   * Once the countdown reaches zero, the Timer should notify the user that the time is up.
   * A notification, such as "Time's up! Bell rings," should be displayed.

**Flowchart**

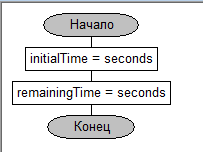
****

Figure 1. Timer(int seconds)

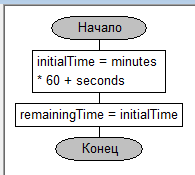
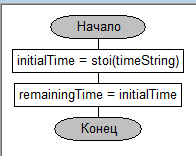
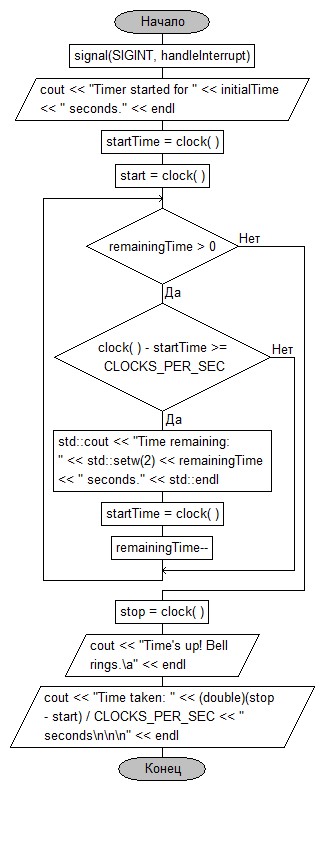
****

Figure 2. Timer(const string& timeString)

Figure 3. Timer(int minutes, int seconds)

**Task 2**

**Class Nomenclature**

class Nomenclature  
{  
private:  
 string productName;  
 double wholesalePrice;  
 double retailMargin;  
 double quantityInStock;

**Product Class (Nomenclature):**

* Store and display product details, including name, wholesale price, retail margin, and quantity in stock.
* Calculate net income from the sale of the product.
* Provide user interface for entering product details.

**Task 3**

class Software  
{  
private:  
 string programName;  
 string developer;  
 double volumeOccupied;  
 time\_t licenseExpirationDate;

**Software Class (Software):**

* Store and display software details, including program name, developer, volume occupied, and license expiration date.
* Calculate days until license expiration.
* Provide user interface for entering software details with a formatted expiration date.

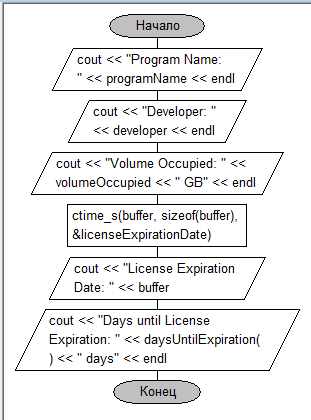
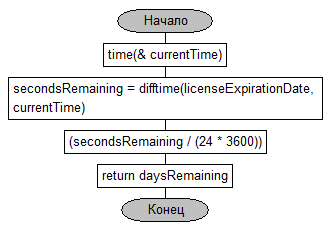
****

Figure 4. displaySoftwareData()

Figure 5. daysUntilExpiration()

**Non-functional Requirements:**

1. User Interface:
   * Console-based user interface.
   * Easy-to-understand prompts and messages.
2. Data Validation:
   * Validate user inputs to ensure correctness and prevent errors.
3. **What is a constructor? How is it called for execution?**
   * A constructor is a special member function of a class in C++ that is automatically called when an object of that class is created. It is used to initialize the object's data members and allocate necessary resources. The constructor has the same name as the class and is called automatically when an object is instantiated.
4. **Features of a constructor that distinguish it from other class methods?**
   * The constructor has the same name as the class.
   * It has no return type, not even void.
   * It cannot be called explicitly; it is automatically invoked during object creation.
   * It is responsible for initializing the object's data members.
5. **What is the default constructor? In what case should it be used?**
   * A default constructor is a constructor that can be called without any arguments. If a class does not have any constructor defined, the compiler provides a default constructor. It initializes the object with default values or leaves the members uninitialized.
   * It should be used when you want to create an object without providing specific initialization values.
6. **Default constructor initialization methods? Which initialization method is preferable?**
   * Default constructor initialization methods include using an initialization list or assigning values in the constructor body.
   * Using an initialization list is preferable because it allows initializing constant and reference members and ensures that initialization happens before the constructor body.
7. **What is a constructor with arguments? When to use a constructor with arguments?**
   * A constructor with arguments is a constructor that accepts parameters to initialize the object with specific values.
   * It is used when you want to provide initial values to the object at the time of creation, allowing more flexibility in object initialization.
8. **What are the advantages of a constructor with arguments over the Set() method?**
   * Constructor with arguments allows for initializing object members at the time of creation, making it a one-step process.
   * It ensures that an object is always in a valid state right from the start.
   * It simplifies the program code and avoids the need for separate initialization methods like Set().
9. **What is a destructor? What is it used for?**
   * A destructor is a special member function with the same name as the class, preceded by a tilde (~). It is automatically called when an object goes out of scope or is explicitly deleted.
   * It is used to release resources, perform cleanup operations, or deallocate memory that was allocated during the object's lifetime.
10. **Features of a destructor:**
    * It has the same name as the class, preceded by a tilde (~).
    * It has no return type and no arguments.
    * It is automatically called when the object is destroyed.
11. **In what order are constructors executed and in what order are class destructors executed?**
    * Constructors are executed in the order in which they are declared in the class.
    * Destructors are executed in the reverse order of their declaration.
12. **Is it possible to overload a constructor?**
    * Yes, it is possible to overload a constructor by defining multiple constructors in a class with different parameter lists. This allows creating objects with different sets of initialization values.
13. **What is a copy constructor?**
    * A copy constructor is a constructor that takes an object of the same class as a parameter and creates a new object, copying the values of the members from the passed object. It is used to initialize an object with the values of another object of the same type.