# **Федеральное государственное бюджетное профессиональное образовательное учреждение**

**Торжокский политехнический колледж**

# **Федерального агентства по государственным резервам**

**ОТЧЁТ**

# о зимней работе

**по профессиональному модулю МДК.01.01**

«**РАЗРАБОТКА ПРОГРАМНЫХ МОДУЛЕЙ**»

наименование профессионального модуля

## Специальность ***09.02.07 «Информационные системы и программирование»***

Период с «28» декабря 2024г. по «08» февраля 2025г.

## Выполнили студенты

Глагольев Е.В., Венидиктов Д.Р.

(ФИО, подпись)

|  |  |
| --- | --- |
| Преподаватель  Бойков А.М.  (ФИО, подпись) | оценка |

## 

## 2024

**Введение**

Цель работы – получить практический опыт парного программирования и разработки модульной структуры программы.

Задачи:

1) выявить практическую значимость проектируемого программного средства (далее – ПС) (с точки зрения использования, с точки зрения развития продуктов большего масштаба и т.д. и т.п.);

2) обосновать функции разрабатываемого программного средства и варианты его использования (какие задачи позволит решить и кому);

3) спроектировать модульную структуру программного средства (структура классов: данные и методы);

4) описание нескольких алгоритмов методов (как минимум – два), отвечающие непосредственно за решение поставленной задачи;

5) разработать программное средство в среде Visual Studio.

**1. Практическая значимость программного средства**

Программа предназначена для эффективного планирования и организации рабочего времени пользователей. С ее помощью можно структурировать задачи, предстоящие на конкретный день, установить приоритеты, следить за выполнением дел и повысить продуктивность рабочего процесса.

Система позволяет добавлять и удалять задачи, отображать задачи на определённую дату, завершать задачи и сохранять список задач в файл.

Пользователями ПС могут быть люди всех профессиональных сфер и возрастов, включая студентов, офисных работников, предпринимателей, домохозяек и всех, кто стремится к организованности, планированию своего времени и управлению задачами.

В случае отсутствия данного ПС, пользователи могли бы столкнуться с необходимостью ручного создания и ведения списка задач, что потенциально привело бы к неструктурированности в планировании рабочих задач, забыванию важных дел и упущению сроков выполнения задач. Отсутствие такого инструмента могло бы снизить эффективность управления временем, привести к риску упустить важные задачи и снизить общую производительность в выполнении рабочих обязанностей.

Использование данной программы позволяет пользователям структурировать свои задачи и планировать их выполнение на конкретный день, устанавливать приоритеты, контролировать выполнение дел, а также эффективно организовывать свое рабочее время. Благодаря этой программе, пользователи могут быть более организованными, улучшить управление своими задачами и повысить свою продуктивность и эффективность в достижении поставленных целей.

**2. Функциональные требования к программному средству**

Программа для составления списка дел должна обеспечивать выполнение следующих функций:

1. Основные возможности ПС:

• Добавление задачи

Пользователь вводит описание задачи и срок выполнения. Задача добавляется в список к остальным задачам.

Требуется привести описание формата ввода задачи, в т.ч. форматов каждого отдельного атрибута

• Удаление задачи

Пользователь выбирает задачу, и она удаляется из списка

Требуется описать формат вывода списка задач

Требуется указать, каким образом пользователь может осуществить удаление задачи

• Отображение задач на определённую дату

Пользователь вводит дату, и отображаются все задачи на эту дату.

Требуется указать форматы поддерживаемых дат

Требуется указать, в каком виде должен быть представлен ответ на запрос пользователя

• Завершение задачи

Пользователь выбирает выполненную задачу, и она удаляется из списка.

В разделе №1 описано, что с помощью данного ПС можно отслеживать выполненные и невыполненные задачи. Если выполненная задача удаляется, то отследить ее невозможно. Требуется не удалять задачу, а отслеживать ее статус: в работе, завершена.

Описать, как пользователь должен отметить задачу выполненной

• Сохранение списка задач в файл

Пользователь может сохранить весь список задач в отдельный файл.

Описать формат такого выходного файла и привести примеры

2. Взаимодействие пользователя с программой:

Описание пунктов перенести и соотнести по каждому требованию

Программа работает в консольном режиме. Пользователь взаимодействует с ней через меню:

1. Добавить задачу.

2. Удалить задачу.

3. Отобразить задачи на дату.

4. Завершить задачу.

5. Сохранить в HTML.

6. Выход.

3. Ошибки и валидация данных:

Описание пунктов перенести и соотнести по каждому требованию

Программа должна обрабатывать ошибки ввода и некорректные данные:

• Если пользователь вводит неверный номер задачи при её удалении или завершении, программа уведомляет пользователя об ошибке.

Привести пример сообщения, которое должно выдавать ПС

• Если пользователь вводит неверный номер действия, программа уведомляет пользователя об ошибке.

Что такое номер действия и чем он отличается от номера задачи? Оставить один какой-то термин в рамках документа.

Дать определение, что такое «неверный» номер

4. Пример работы программы:

1. Пользователь запускает программу.

2. Выбирает пункт “Отобразить задачи на дату”.

3. Программа выводит все задачи с этой датой на экран пользователя.

4. При желании пользователь может завершить или удалить некоторые задачи.

5. Пользователь может сохранить список задач в файл, при необходимости.

6. При завершении работы программа закрывается.

Программа обеспечивает структурирование, управление и отслеживание списка дел, повышая организованность, эффективность и продуктивность пользователей.

**3. Модульная структура проектируемого программного средства**

**Раздел должен начинаться диаграммой классов!!!**

Программное средство построено на основе модульной структуры, включающей несколько классов, каждый из которых выполняет строго определенные функции. Это обеспечивает удобство расширения, поддержку кода и его повторное использование.

**Структура классов:**

Программа состоит из следующих классов:

• Класс TaskItem – хранит данные о задаче. Он содержит поля для описания задачи, даты и статусе. Этот класс необходим для хранения и обработки информации о задачах.

• Класс TaskManager – основной управляющий класс, который использует список товаров??? и реализует методы добавления и удаления задачи, получения задач на определённую дату, сохранения списка задач в файл и завершения задачи.

Класс TaskManager перегружен функцией сохранения задач в файл, это должен быть отдельный класс

Нет класса, реализующий функционал навигации пользователя в программе

Нет класса или описания методов, отвечающих за обработку ошибок ввода пользователя

**Описание классов и их методов:**

Класс TaskItem включает в себя следующие поля:

• description\_ – строковое поле, представляющее описание задачи.

• dueDate\_ – поле типа DateTime, представляющее дату, к которой должна быть выполнена задача.

• isCompleted\_ – булевое поле, представляющее статус товара. По умолчанию имеет значение false.

Нет атрибута, позволяющего однозначно идентифицировать задачу в системе

Класс TaskManager содержит список объектов класса TaskItem и реализует методы для работы с задачами:

• AddTask – метод, добавляющий новый ~~элемент~~ в список.

• RemoveTask– метод, удаляющий выбранный пользователем ~~элемент~~ из списка.

• GetTasksByDate– метод, возвращающий ~~элементы~~ списка, в которых ~~значение dueDate\_~~ совпадает со ~~значением~~, введённым пользователем.

• SaveToHtml – метод, сохраняющий список в файл формата HTML.

• CompleteTask – метод, ~~устанавливающий полю isCompleted\_ значение true в выбранном пользователем элементе списка~~.

**Описание интерфейса метода должно базироваться на том, что произойдет с точки зрения поведения программы, а не изменения каких-то конкретных полей. В данном случае: метод, позволяющий сделать выполненной задачу X, передаваемую в качестве аргумента метода или выполненной задачу с номером X, передаваемый в качестве аргумента.**

**Проверить остальные описания методов (слова отмечены красным и зачеркнуты)**

Описание методов приводить в нотации C#.

Нет описания типов возвращаемых значений и списка аргументов методов

**Взаимосвязь классов:**

Класс TaskManager является управляющим классом, который использует и координирует работу с классом TaskItem, который в свою очередь хранит данные конкретной задачи. TaskManager использует экземпляры класса TaskItem для реализации методов. Таким образом, TaskManager отвечает за управление списком задач и реализацию операций, используя данные, предоставляемые классом TaskItem.

**~~Логика работы классов:~~**

~~1. При запуске программы создается объект List<TaskItem> tasks.~~

~~2. Пользователь выбирает действие из меню, которое соответствует методу класса TaskManager.~~

~~3. После выполнения выбранного действия программа возвращается в главное меню, предлагая выбрать следующее действие.~~

Разделы 4 и 5 отсутствуют