

Задача 6. Сохранение состояния

Требования к программам:

- Язык программирования — C++, в чистом виде, без расширения типа .net.
- Программа — консольное приложение с меню.
- В коде обязательно условие задачи и автор (в комментариях).
- Комментарии к коду обязательны.

Общее условие:

Выполнение основано на задании 2.

Завершение программы не должно сбрасывать состояние. При очередном запуске программы, по мере необходимости, автоматически восстанавливать прошлое состояние. Для этого предусмотреть хранение информации на жестком диске. Обязательна внимательная обработка ошибок.

Задания:

1. Реализовать систему управления грузовыми портами: управление списком портов — добавление, удаление, изменение, просмотр; управление выбранным портом — добавление, редактирование, удаление контейнера; вывод отфильтрованных контейнеров как в отдельном порту, так и по всем портам вместе; перемещение группы выбранных по фильтру контейнеров из порта в порт. Предусмотреть вывод сводного отчета по всем грузовым портам вида «всего портов, всего контейнеров, контейнеров в порте 1, контейнеров в порте 2 ..., суммарная масса контейнеров, количество обычных, содержащих взрывчатые вещества, содержащих газы, содержащих легковоспламеняющиеся жидкости...».
2. Реализовать систему управления плейлистами: управление списком плейлистов — добавление, удаление, изменение, просмотр; управление выбранным плейлистом — добавление, редактирование, удаление трэка; вывод отфильтрованных композиций как в отдельном листе, так и по всем плейлистам вместе; перемещение группы выбранных по фильтру треков между плейлистами. Предусмотреть вывод информации о медиатеке вида «всего композиций, всего различных альбомов, всего исполнителей, суммарная продолжительность».
3. Реализовать взаимодействие пользователей (читателей) с библиотекой. Пользователи выбираются из списка доступных. Можно добавить нового пользователя. Удалить пользователя из списка можно только, если у него на руках нет книг. Читатель может брать доступные книги из библиотеки и возвращать их. Также читатель может просматривать список всех доступных книг и фильтровать их. Добавить второй режим — администратора (библиотекаря), который также может смотреть и фильтровать список книг в библиотеке, добавлять, удалять и редактировать книги, которые не выданы на руки, получать статистику по библиотеке: всего читателей, книг в библиотеке, всего книг на руках.

4. Реализовать класс «Чародей», который имеет свой запас маны, а также хранит несколько магических книг, в которых записаны заклинания. Чародей должен иметь возможность просматривать любую имеющуюся книгу, выбирать из нее подходящие по фильтру заклинания, удалять их, править и добавлять новые. Также чародей может завести новую книгу или избавиться от старой. Кроме этого, чародей может колдовать, если ему хватает маны на данное заклинание и время суток подходящее, а также восстанавливать запас маны отдохнув (при этом текущее время увеличивается на N часов). Еще чародей может припомнить сколько раз он использовал заклинания из каждой отдельной книги и сколько раз он применял заклинания каждой отдельной стихии.
5. Реализовать систему игровых сохранений, где каждому игроку по его имени соответствует некоторый набор сохранений. Для каждого игрока хранится лишь самые свежие N сохранений. Можно создавать или удалять новый профиль, делать сохранения, загружать или удалять. Также можно просматривать информацию по всем сохранениям игрока в т. ч. с различными фильтрами. Добавить возможность просмотреть общую информацию «всего игроков, всего сохранений, дата и время последнего изменения; игрок «...» дата-время последнего сохранения, игрок «...» дата-время последнего сохранения...».
6. Реализовать файловую систему, состоящую из корня, папок и файлов (добавление папки в папку не предусмотрено, как и добавление файлов в корень). Файлы внутри папки можно добавлять, удалять, редактировать, сортировать, а также выбирать по критерию для просмотра или перемещения между папками. Папки можно создавать, удалять и переименовывать. Предусмотреть вывод информации: количество и объем файлов в каждой папке, количество папок и суммарный объем файлов.
7. Газовая компания хранит записи о платежах по группам (например, «текущий месяц», «прошлый год», «старее» или «коминтерновский», «центральный», «юго-западный») — группы можно добавлять, редактировать, удалять (при этом удаляя все входящие в нее записи, просматривать, в т. ч. с фильтрацией по критериям. В каждой отдельно взятой группе можно добавлять, удалять или редактировать квитанции, а также перемещать их между группами по заданному фильтру. Предусмотреть вывод сводного отчета по компании: общее число квитанций и сумма платежей, а также следующую информацию по каждой группе: квитанций, общая сумма, дата самого первого и самого последнего платежа.
8. Реализовать систему управления учебными группами: управление списком групп — добавление, удаление, изменение, просмотр; управление выбранной группой — зачисление, редактирование, отчисление студента; вывод отфильтрованных студентов по группе; перемещение выбранных по фильтру студентов из группы в группу. Предусмотреть вывод сводного отчета по всем группам вида «всего студентов, общий средний балл, всего на красный диплом, группа №1: студентов, средний балл, на красный диплом, группа №2: студентов, средний балл, на красный диплом...».

9. Реализовать систему управления сетью оружейных магазинов у каждого из которых есть свой оружейный склад, а также существует один распределительный склад, с которым могут взаимодействовать магазины. Добавить возможность управления списком магазинов — добавление, удаление, изменение, просмотр; возможность управления выбранным магазином — продажа (печать с удалением) оружия и просмотр списка доступного вооружения с различными фильтрами; возможность перемещения выбранного по фильтру оружия из магазина в распределительный склад и обратно; склад предоставляет возможность добавлять, изменять и просматривать оружие (и перемещать, но не продавать). Предусмотреть вывод сводного отчета по всему вооружению в сети вида «всего оружия, всего магазинов, оружия на складе, оружия в магазине 1, оружия в магазине 2 ...».
10. Реализовать электронный автомагазин — клиент может выбрать через него автосалон из списка, в каждом из которых может просмотреть любой автомобиль, а также фильтровать для себя выводимый список автомобилей. Администратор имеет все возможности обычного пользователя, а также редактировать список автосалонов, редактировать (добавлять, удалять, изменять и перемещать выбранные автомобили между салонами) список автомобилей в салоне и просматривать отчеты всего магазина: всего авто, всего салонов, список из уникальных марок авто, собранный по всем салонам, список из уникальных моделей, собранный по всем салонам, количество авто в каждом салоне.
11. Реализовать систему управления предприятиями: управление списком рабочих — добавление, удаление, изменение, просмотр; управление выбранным предприятием — добавление, редактирование, удаление рабочего; вывод отфильтрованных работников в предприятия; перевод группы выбранных по фильтру работников между предприятиями. Все предприятия хранят желаемую профессию работника и ожидаемое число рабочих. Предусмотреть операцию автоматического распределения сотрудника в подходящее предприятие, учитывая эти два поля. Добавить вывод сводного отчета по всем предприятиям «всего предприятий, всего рабочих, предприятие 1: желаемая профессия, желаемое число сотрудников, всего работников, подходящих работников; предприятие 2: ...».
12. Реализовать систему управления сетью питомников для животных (каждый питомник имеет название и набор подходящих видов, животные другого вида не могут находиться в нем): управление списком питомников — добавление, удаление, изменение, просмотр; управление выбранным питомником — добавление, редактирование, удаление животного; вывод отфильтрованных животных как в отдельном питомнике, так и по всем питомникам вместе. Предусмотреть вывод сводного отчета по всем питомникам вида «всего питомников, всего животных, питомник '...' самок, самцов, хищных, травоядных, всеядных, падальщиков, питомник '...' самок, самцов, хищных, травоядных, всеядных, падальщиков...».

13. Каждый раздел Бестиария хранит «название категории» и свой набор тварей. Разделы можно редактировать: создавать, удалять, менять. Записями о тварях в разделах можно управлять: просматривать, добавлять, удалять, редактировать, выбирать по фильтру (просто просматривать или перемещать выбранное в другой раздел). Предусмотреть вывод отчета о текущем состоянии: всего тварей, средний уровень, изучено тварей, не изучено тварей (нет даты изучения или дефолтное значение), после этого те же данные нужно вывести по каждому разделу, указав перед этим его название.