

## 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 (или сразу), автоматически восстанавливать прошлое состояние. Для этого предусмотреть хранение информации на жестком диске. Обязательна внимательная обработка ошибок.*

### Задания:

1. Реализовать приложение, позволяющее управлять колониями членистоногих: управление набором колоний — добавление, удаление, изменение, просмотр; управление выбранной колонией — добавление, редактирование, удаление существ; выборка отфильтрованных существ в колонии с возможностями печати и перемещения. Каждая колония имеет свое название. Добавить «массовое вымирание», при котором подходящие под критерий создания удаляются из всех колоний.

Предусмотреть вывод статистики по всем колониям: таблицу вида «название колонии — количество существ», суммарное число колоний, суммарное количество членистоногих, среднее количество существ на колонию (2 знака после запятой), число крылатых и число бескрылых созданий, подробная информация о существах с максимальной массой и максимальной длиной.

2. Реализовать систему управления храмового комплекса: управление набором храмов — добавление, удаление, изменение, просмотр; управление выбранным храмом — добавление, редактирование, удаление рукописей; выборка и печать отфильтрованных рукописей. Добавить возможность перемещения рукописи в другой храм, при этом, с вероятностью N%, монах сделает копию и оставит ее себе. Каждый храм характеризуется названием, именем монаха, проживающего в нем, а также усердностью монаха N.

Предусмотреть создание отчета для великого императора по всем храмам: таблицу вида «подробная информация о храме — количество древних рукописей», суммарное число рукописей, суммарное количество храмов, суммарное число уникальных рукописей, суммарное число: табличек, пергаментов, свитков, книг... по категориям.

3. Реализовать приложение, работающее с параллельными мирами: управление набором миров — добавление, удаление, изменение, просмотр. Также имеется один древний мир, который не может быть удален. Должна быть реализована возможность управления выбранным миром — добавление, редактирование, удаление существ (демонов); выборка, печать и удаление отфильтрованных демонов. Выбранных по критерию демонов можно перемещать между текущим миром и древним миром (или наоборот между древним и указанным, но не между собой).

Предусмотреть создание отчета для старших богов по всем мирам: таблицу вида «информация о мире — количество населяющих его демонов», суммарное число демонов, суммарное число миров, суммарное число рогов по всем мирам, суммарное число демонов по каждому цвету.

4. Реализовать систему управления лабораториями: добавление, удаление, изменение и просмотр лабораторий. Каждая лаборатория относится к своему городу с названием и количеством жителей. Должна быть реализована возможность управления выбранной лабораторией: добавление, редактирование, удаление и просмотр заболеваний; выборка отфильтрованных заболеваний с возможностями печати и перемещения в другую лабораторию.

Добавить функцию «катастрофа», при которой в населенном пункте, содержащем лабораторию, происходит вымирание (выбрать случайную болезнь из контейнера) пропорциональное населению и числу путей распространения заболевания.

Предусмотреть вывод статистики по всем лабораториям: таблицу вида «информация о лаборатории — количество изучаемых заболеваний — среднее число путей распространения на заболевание», суммарное число лабораторий, суммарное количество заболеваний, среднее число заболеваний на лабораторию (2 знака после запятой).

5. Создать консольное приложение, имитирующее работу сети, осуществляющей производство и доставку заказов блюд из различных точек. Каждая такая точка имеет свои название, адрес и меню (набор блюд). Клиент должен мочь просмотреть список точек, выбрать нужную, выбрать из нее подходящие блюда по различным критериям (см. задачу 2) и сортировать. При входе клиент указывает свой адрес.

Кроме того, должно быть осуществлено консольное меню для администратора (вместо адреса указывается «admin»), который может редактировать (добавлять, изменять и удалять) список таких точек и редактировать сами точки (добавлять, изменять и удалять блюда, копировать выбранную по фильтру часть меню в другую точку).

Администратор также может получить статистику по всей сети: таблицу, состоящую из названия точки, количества блюд в ее меню, максимальное и минимальное время приготовления блюд, максимальная и минимальная

калорийность блюд в точке. Под таблицей приводится строка итогов: всего точек, всего блюд, среднее время готовки, средняя калорийность.

6. Каждый злодей должен сидеть в тюрьме. Для грамотного распределения злодеев по тюрьмам нужно создать приложение-общую тюремную систему, которое позволяет управлять набором подключенных к нему тюрем. Каждая тюрьма характеризуется названием и набором заключенных. Тюрьмы можно добавлять, удалять и изменять. Можно управлять отдельно взятой тюрьмой: добавлять, удалять (злодей оказывается на свободе) и редактировать заключенных в ней злодеев, а также просматривать списки злодеев по тому или иному критерию (см. задачу 2) и иметь возможность переместить заключенных в другое место.

Однако, часть злодеев по-прежнему находится на свободе. Имена их неизвестны (только прозвища). Детективы могут попытаться поймать злодея. Для этого плохиш выбирается из списка всех свободных и детектив называет его предполагаемое имя. Если догадка верна, злодея помещают в тюрьму (предоставить пользователю возможность выбора заведения из списка).

Помимо прочего, цифровая тюремная система позволяет просматривать статистику: число заключенных в каждой тюрьме, общее число тюрем, общее число заключенных, среднее число заключенных.

7. Реализовать консольное приложение для управления компанией, состоящей из множества отделов — добавление, удаление, изменение, просмотр отдела; управления выбранным отделом — добавление, редактирование, удаление сотрудников; выборка по критерию (см. задачу 2) по отделу с возможностями печати и перемещения сотрудников в другой отдел.

Добавить «формулу полезности» сотрудника, которая переводит все его поля в некоторое числовое значение (на свой вкус). Добавить возможность удалять самого бесполезного сотрудника из отдела и возможность удалять самого бесполезного сотрудника из компании.

Предусмотреть вывод статистики по всей компании: таблицу вида «название отдела — количество сотрудников — суммарный оклад», а также строку «наибольший и наименьший оклады сотрудников компании», информацию о лучшем сотруднике.

8. Чтобы войти в систему, пользователю нужно указать логин и пароль. Каждый пользователь хранит список постов (максимум N, самые старые автоматически удаляются). Юзер имеет возможность опубликовать новый пост (дата, время и автор заполняются автоматически, оценка = 0). Пользователь имеет возможность просмотреть все свои посты либо вообще все посты, либо вообще все посты, отсортированные по дате и времени (сначала свежие), а также поставить оценку не своим постам (+1 или -1 к оценке) [невозможность повторного оценивания

рассматривать не требуется], выбрать только интересующие посты (по критериям задачи 2).

Добавить возможность зарегистрировать нового юзера с уникальным логином и пустым набором постов.

Модератор также входит в систему с логином и паролем admin. Он имеет возможность удалять отдельно взятых пользователей со всеми их постами (банить), может удалить определенные посты. Кроме этого он может просматривать статистику сайта: общее число пользователей, общее число постов, число и средний рейтинг всех постов за последние сутки, неделю и месяц.

9. Реализовать приложение, позволяющее управлять системой складов стройматериала. Склад характеризуется уникальным номером. Предусмотреть управление набором складов — добавление, удаление, изменение, просмотр; управление выбранным складом — добавление, редактирование, удаление материала; выборка отфильтрованных существ (см. лабу 2) с возможностями печати и перемещения. Если количество товара стало равно нулю, он НЕ удаляется.

Добавить в главное меню пункт, который позволяет посмотреть информацию о товарах, которые закончились или заканчиваются ( $<N$ ) по всем складам. При этом должна выводиться информация и о товаре, и о складе, на котором его мало.

Предусмотреть вывод статистики по всем складам: таблицу вида «номер склада — количество товара на нем (не видов товара) — суммарная масса товаров».

10. Есть несколько игроков, имеющих собственный инвентарь, а также «брошенное» оружие — то, которое находится не у какого-либо из игроков. Игроки могут подбирать и выбрасывать оружие, обмениваться им между собой (по условиям из лабы 2). Список игроков можно редактировать — добавлять, изменять, удалять. Игроки характеризуются уникальным никнеймом. Если подбирается то же самое оружие, которое уже есть у игрока, оно не заменяется — вместо этого суммируется количество патронов.

Предусмотреть в меню возможность добавлять, удалять и изменять оружие «на земле».

Добавить вывод статистики: имя игрока, занято слотов. свободно. Также выводить любимое оружие — то, что наиболее часто встречается в арсенале у игроков.

11. Реализовать «лавку наемников» — набор юнитов, который можно редактировать с консоли: добавлять, удалять и изменять. Реализовать «игроков», имеющих никнейм и свой отряд юнитов. Игрок может пополнять свой отряд из лавки, продавать обратно неподходящих бойцов, обмениваться юнитами с другими игроками (замена 1 на 1). Можно просматривать всех юнитов у игрока или в лавке по критериям задачи 2. Игроков можно добавлять и удалять.

Добавить вывод статистики по игрокам: имя игрока, имеется юнитов. средний уровень, среднее здоровье, средний запас маны. Добавить вывод статистики по юнитам: раса, количество юнитов данной расы, средний уровень, среднее здоровье, средний запас маны.

12. В голове каждого студента находится некоторое число языков программирования. Этот список можно пополнить, посетив лекции. Для этого из списка лекций (я.п.) нужно выбрать определенную (одну или несколько) по критериям задачи 2. Когда студент посещает лекцию, новый язык прибавляется к его списку изученных. Если посетить лекцию по уже изученному языку ничего не произойдет. Добавить возможность редактировать список лекций.

Каждый студент, помимо списка языков характеризуется также ФИО, номером студенческого и годом поступления. Должна быть возможность добавлять, удалять и подробно рассмотреть студента (вывести всю информацию о нем).

Добавить возможность выбрать сразу несколько студентов для посещения занятий.

Добавить вывод информации по всем языкам из списка лекций: процент изученности языка студентами.

13. В том ужасном мире, в котором мы все вынуждены существовать, на каждом углу нет-нет и наткнешься на какую-либо тварь. И просто так от нее уже не отдалиться — герой вынужден вступить в схватку. Никогда не знаешь заранее кто выйдет победителем — все зависит от характеристик (здоровье, урон) героя и твари, ее слабых сторон и готовности героя ими воспользоваться (оружие героя: серебряное, огонь, яд..., которое наносят К-кратный урон уязвимым к данному типу оружия противникам), и конечно от удачи (шанс N% любому существу нанести двойной урон).

Пользователь может настроить своего героя в главном меню и отправиться на встречу приключениям (встретиться на поле битвы с 10 случайными тварями из predetermined списка (герой восстанавливает силы между боями)). Если тварь побеждена, она попадает в Бестиарий (только если записи о ней еще нет ни в одном разделе, попадает в дефолтный раздел). Если же герой повержен, он возвращается в главное меню.

Также из меню можно пролистать сам Бестиарий. Каждый раздел Бестиария хранит «название категории» и свой набор тварей. Разделы можно редактировать: создавать, удалять (только пустой и не дефолтный), менять. Записями о тварях в разделах можно управлять: просматривать, сортировать, выбирать по фильтру (просто просматривать или перемещать выбранное в другой раздел).

Кроме того, из меню можно просмотреть список всех тварей, с которыми герой может столкнуться в своем путешествии. К сожалению, в данном списке видны лишь названия монстров без подробностей. Этот список можно редактировать.