

## Задача 5. Универсальные шаблоны, передача функций

### 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:

*Программа должна содержать шаблонный класс-контейнер для управления данными: структурой или классом, указанными в задании. Для хранения данных необходимо выбрать оптимальный с точки зрения задания контейнер. Реализовать следующие функции:*

- *Добавление элемента контейнера;*
- *Удаление элемента контейнера;*
- *Изменение элемента контейнера;*
- *Ввод элементов контейнера с использованием потоковых итераторов на экран и в файл;*
- *Вывод элементов контейнера с использованием потоковых итераторов на экран и в файл;*
- *Выборка подмножества элементов по заданным критериям с помощью алгоритма линейного поиска.*

*Выбранные подмножества должны мочь выводиться на экран и в файл.*

### Задания:

1. «Морской контейнер» представляет собой структуру с полями: уникальный номер контейнера, дата поступления на склад (использовать одну из предлагаемых языком структур), размер (перечисление [20-футовый, 40-футовый...]), масса груза, собственная масса, максимальная допустимая масса груза, категория опасного груза (перечисление [нет, взрывчатые вещества и изделия, газы...]). Выборка контейнеров производится путем указания пользователем подходящего интервала по одному из критериев: дата поступления, размер, фактическая масса. Также выборка по фильтру наличия-отсутствия категории опасности. Шаблонный класс должен иметь дополнительный метод — двоичный поиск по условию. Реализовать поиск по id контейнера.
2. «Музыкальная дорожка» представляет собой структуру с полями: уникальный номер трэка, альбом (может не быть), порядковый номер в альбоме, дата выпуска (использовать одну из предлагаемых языком структур), исполнитель, жанры (один или несколько), продолжительность. Выборка треков производится путем указания пользователем подходящего интервала по одному из критериев: дата выпуска, альбом, исполнитель (например, от «Sa» до «Sl» включительно). Также выборка всех треков, относящихся к (одному) указанному жанру. Шаблонный класс должен

иметь дополнительный метод — удаление по условию. Реализовать удаление по: id трэка, альбому, исполнителю.

3. «Библиотечная книга» представляет собой структуру с полями: уникальный номер книги, название, язык (перечисление), дата написания (использовать одну из предлагаемых языком структур), автор, издательство, год выпуска, объем (в страницах), дата выдачи и срок возврата (может не быть, если книга не на руках). Выборка книг производится путем указания пользователем подходящего интервала по одному из критериев: название (например, от «Alice» до «Rumpelstilzchen» включительно), автор, объем. Также выборка всех книг по фильтру «выдана-не выдана». Шаблонный класс должен иметь дополнительный метод слияния (перегрузку оператора «+») с другим экземпляром того же класса. Добавить пункт меню, демонстрирующий работу этого метода.
4. «Заклинание» представляет собой структуру с полями: название, краткое описание, стихия (перечисление), затрачиваемая мана, сила и продолжительность эффекта, минимальный уровень чародея для использования, время суток «с» и «по» которое оно может быть применено (использовать одну из предлагаемых языком структур). Выборка заклинаний производится путем указания пользователем подходящего интервала по одному из критериев: название (например, от «Mutilatio» до «Somnium» включительно) и расход маны. Также выборка всех заклинаний, доступных в указанное время, выборка всех заклинаний, относящихся к указанной стихии, выборка всех заклинаний, доступных на указанном уровне. Шаблонный класс должен иметь дополнительный метод сортировки. Добавить в меню возможность отсортировать заклинания по минимальному уровню по возрастанию и по названию по алфавиту.
5. «Игровое сохранение» представляет собой структуру с полями: уникальный номер, дата и время (использовать одну из предлагаемых языком структур), здоровье, мана, запас сил персонажа, название текущей локации, цифровой код последней выполненной миссии. Выборка сохранений производится путем указания пользователем подходящего интервала по одному из критериев: дата-время и по коду миссии. Также выборка всех сохранений по названию локации. Шаблонный класс должен иметь дополнительный метод «извлечь последний». Добавить в меню возможность загружать последнее из оставшихся сохранений.
6. «Файл» представляет собой структуру с полями: имя файла; расширение; дата и время создания (использовать одну из предлагаемых языком структур); дата и время последнего изменения; атрибуты: только на чтение, скрытый, системный; объем. Выборка файлов производится путем указания пользователем подходящего интервала по одному из критериев: создан, изменен, объем. Также выборка всех файлов по расширению, выборка всех файлов по фильтру только на чтение/нет, по фильтру системный/нет, по фильтру скрытый/нет. Шаблонный класс должен иметь дополнительный метод сортировки. Добавить в меню возможность отсортировать файлы по дате изменения и по названию в прямом и обратном порядке.

7. «Квитанция за газ» представляет собой структуру с полями: адрес (р-н, улица, дом, квартира), получатели платежа, лицевой счет, сумма к оплате, последние показания, текущие показания, срок оплаты (для даты использовать одну из предлагаемых языком структур). Выборка квитанций производится путем указания пользователем подходящего интервала по одному из критериев: получатели (например, от «А» до «К» включительно), сумма к оплате, срок. Также выборка всех квитанций по: району, улице. Шаблонный класс должен иметь дополнительный метод вычисления числовой агрегатной функции суммирования. Добавить в меню вычисление суммы к оплате по всем квитанциям.
8. «Карта студента» представляет собой структуру с полями: ФИО студента, ступень (перечисление: бакалавр, специалист, магистр), направление обучения, дата поступления (использовать одну из предлагаемых языком структур), номер студенческого, набор записей вида «дисциплина – оценка». Выборка карт производится путем указания пользователем подходящего интервала по одному из критериев: ФИО (например, от «М» до «Ш» включительно), дата поступления. Также выборка всех карт по ступени образования. Шаблонный класс должен иметь дополнительный метод — двоичный поиск по условию. Реализовать поиск по: номеру студенческого, по ФИО.
9. «Оружие» представляет собой структуру с полями: серийный номер, калибр (перечисление), модель, производитель, дата изготовления (использовать одну из предлагаемых языком структур), длина, масса. Коллекция, хранящая оружие всегда отсортирована по дате изготовления. Выборка оружия производится путем указания пользователем подходящего интервала по дате изготовления. Также выборка всего оружия по: калибру, модели или производителю.
10. «Автомобиль» представляет собой структуру с полями: серийный номер, модель, марка, тип кузова (перечисление), мощность, цвет, привод (передний/задний/полный), дата сборки (для даты использовать одну из предлагаемых языком структур). Выборка автомобилей производится путем указания пользователем подходящего интервала по одному из критериев: мощность, дата сборки. Также выборка всех автомобилей по: марке, модели, типу привода. Удаление (извлечение) из шаблонного класса-контейнера всегда применяется к самому первому из попавших в него элементов.
11. «Рабочий» представляет собой структуру с полями: имя, пол, профессия (перечисление), дата рождения (для даты использовать одну из предлагаемых языком структур), черта характера (перечисление, например, дружелюбие, апатия, алкоголизм...). Выборка рабочих производится путем указания пользователем подходящего интервала по одному из критериев: имя (например, от «Владислав» до «Степан» включительно), дата рождения. Также выборка всех рабочих по: профессии, черте характера или полу. Шаблонный класс должен иметь дополнительный метод выборки всех сотрудников, которые не подходят под заданный критерий. Продемонстрировать это, выбирая сотрудников, исключая определенную черту характера.

12. «Зверушка» представляет собой структуру с полями: кличка, вид, пол, тип питания (травоядный/хищный/падальщик/всеядный), дата рождения (для даты использовать одну из предлагаемых языком структур). Выборка животных производится путем указания пользователем подходящего интервала по одному из критериев: кличка (например, от «Изма» до «Салем» включительно), дата рождения. Также выборка всех животных по: вид, пол, тип питания. Шаблонный класс должен иметь дополнительный метод удаления всех животных по критерию. Продемонстрировать это, удаляя животных по виду.
13. «Тварь» представляет собой структуру с полями: название, здоровье, урон, уровень, места обитания (строка), уязвимости (несколько значений из фиксированного набора, например, серебро ☒, дневной свет ☒, огонь ☒, яды ☒, магия ☒, электричество ☒, холод ☒, награда за убийство (целочисленная), дата изучения твари (для даты использовать одну из предлагаемых языком структур). Выборка тварей производится путем указания пользователем подходящего интервала по одному из критериев: название (например, от «Визирь саламандр» до «Совиный медведь» включительно), награда, дата изучения, уровень. Также выборка всех тварей по полю «категория», выбранному пользователем. Шаблонный класс должен иметь дополнительный метод сортировки по указанному условию. Добавить сортировку по: названию, дате изучения, отношению награды к уровню твари.