**Лабораторная работа №18 Часть 2**

*«Обобщения и коллекции»*

**Вариант 1**

1. Напишите программу, обрабатывающую очередь пассажиров, желающих купить авиабилет. Элементы очереди содержат следующую информацию: фамилия пассажира, пункт прибытия, дата вылета. Информация об имеющихся билетах хранится в списке, элементы которого содержат номер рейса, пункт прибытия, дату, число непроданных билетов. Пассажир, обеспеченный билетом, удаляется из очереди, остальные остаются там. Содержимое списка по мере обеспечения пассажиров билетами корректируется. Продемонстрировать работу такой очереди в консоли.
2. Создать класс Студент, у которого будут поля Фамилия с большой буквы и средний балл от 0 до 10. Создать словарь, содержащий список Студентов (не менее 10).

* Упорядочить список по алфавиту. Результат вывести на экран консоли.
* Добавить в словарь еще одного студента, с полями, введенными с консоли. Вывести результат на экран.
* Удалить студентов, фамилия которых начинается на А. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 2**

1. За один просмотр файла, элементами которого являются целые числа, и без использования дополнительных файлов переписать его элементы в другой файл так, чтобы первоначально были записаны все числа, меньшие заданного, а затем все числа из отрезка [a, b] (пользователь вводит с консоли) и все остальные. Взаимный порядок чисел в каждой из групп должен быть сохранен. Указания. В решении задачи числа последовательно считываются из файла. Если очередное число меньше а, то оно записывается в файл, если оно принадлежит отрезку [a, b], то заносится в первую очередь, иначе – во вторую. После завершения чтения в выходной файл записываются числа из первой и второй очередей.
2. Создать класс Продукт, у которого будут поля название с большой буквы и масса в граммах. Создать словарь, содержащий список Продуктов (не менее 10).

* Упорядочить список по убыванию массы продуктов. Результат вывести на экран консоли.
* Добавить в словарь еще один продукт, с полями, введенными с консоли. Вывести результат на экран.
* Удалить продукты, название которых начинается на М. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 3**

1. Вычислите арифметическое выражение. Выражение содержит операции + и -. В качестве операндов для простоты используются цифры от 0 до 9. Выражение является синтаксически правильным и заканчивается знаком =. Например, 5+3-9=. Используйте два стека: стек операций, элементы которого имеют тип char и стек операндов, содержащий элементы целого типа.
2. Создать класс Страна, полями которого будет ее название с большой буквы и численность населения. Создать словарь, содержащий список Стран (не менее 10).

* Упорядочить список по алфавиту. Результат вывести на экран консоли.
* Добавить в словарь еще одну страну, с полями, введенными с консоли. Вывести результат на экран.
* Удалить страны, название которых начинается на В. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 4**

1. Создайте алгоритм сложения двух больших целых чисел, заданных в виде набора цифр хранимых с помощью списка (очереди). Числа вводятся в виде строк с дальнейшим разбиением на цифры и помещением этих цифр внутрь списка.
2. Создать класс Машина, у которого будут поля название с большой буквы и скорость. Создать словарь, содержащий список Машин (не менее 10).

* Упорядочить список по возрастанию скоростей автомобилей. Результат вывести на экран консоли.
* Добавить в словарь еще одну машину, с полями, введенными с консоли. Вывести результат на экран.
* Удалить машину, название которой начинается на S. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 5**

1. С помощью стека решить следующую задачу. Дана строка текста которая в том числе содержит множество скобок: “(”, “{”, “[” как открывающихся, так и закрывающихся. Строка считается корректной в том случае, если всем открывающим скобкам есть соответствующие им закрывающие скобки. Например, строка “abc(as)[]{aa[z]}” – считается корректной, а строка ”[[09]({]})” – не является таковой. Проверьте правильность расстановки скобок в строке.
2. Создать класс Ткань, у которого будут поля цвет с большой буквы и длина полотна. Создать словарь, содержащий список Тканей (не менее 10).

* Упорядочить список по алфавиту. Результат вывести на экран консоли.
* Добавить в словарь еще одну Ткань, с полями, введенными с консоли. Вывести результат на экран.
* Удалить ткань, с самым длинным названием цвета. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 6**

1. Дана строка вида s\*w (s, w – строки из символов, не содержащих символа \*). Чтение разрешено по одному символу. Проверить, является лм строка w обратной строке s. Например, для случая s = ABCDEF, w = FEDCBA ответ положительный.
2. Создать класс Кружка, у которого будут поля цвет с большой буквы и объем. Создать словарь, содержащий список Кружек (не менее 10).

* Упорядочить список по возрастанию объема. Результат вывести на экран консоли.
* Добавить в словарь еще одну кружку, с полями, введенными с консоли. Вывести результат на экран.
* Удалить кружку, с самым коротким названием цвета. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 7**

1. Содержимое текстового файла f, разделенного на строки, переписать в текстовый файл g, перенося при этом в конец каждой строки все входящие в него цифры, с сохранением взаимного исходного порядка.
2. Создать класс Праздник, у которого будут поля название с большой буквы и количество выходных. Создать словарь, содержащий список Праздников (не менее 10).

* Упорядочить список по алфавиту. Результат вывести на экран консоли.
* Добавить в словарь еще один праздник, с полями, введенными с консоли. Вывести результат на экран.
* Удалить праздники, с самыми короткими выходными. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 8**

1. Пусть имеется файл действительных чисел и некоторое число *C,* введенное с консоли. Используя очередь, записать в новый файл сначала все элементы, меньшие числа *C*, а затем все остальные элементы.
2. Создать класс Клавиатура, у которого будут поля цвет с большой буквы и количество кнопок. Создать словарь, содержащий список Клавиатур (не менее 10).

* Упорядочить список по убыванию количества кнопок. Результат вывести на экран консоли.
* Добавить в словарь еще одну клавиатуру, с полями, введенными с консоли. Вывести результат на экран.
* Удалить клавиатуры, у которых название цвета содержит букву «С». Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 9**

1. Даны две непустые очереди (вывести их на экран консоли), которые содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются. Вывести полученный результат на консоль.
2. Создать класс Игрушка, у которого будут поля название с большой буквы и тип (кукла, конструктор, пазл и т.д.). Создать словарь, содержащий список Игрушек (не менее 10).

* Упорядочить список по алфавиту типов игрушек. Результат вывести на экран консоли.
* Добавить в словарь еще одну игрушку, с полями, введенными с консоли. Вывести результат на экран.
* Удалить игрушки, у которых название типа содержит букву «К». Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 10**

1. Напишите программу, определяющую, является ли введенная строка символов палиндромом (то есть читается ли она одинаково слева направо и справа налево). Используйте для решения одну из рассмотренных структур.
2. Создать класс Книга, у которого будут поля название с большой буквы и количество страниц. Создать словарь, содержащий список Книг (не менее 10).

* Упорядочить список по убыванию количества страниц. Результат вывести на экран консоли.
* Добавить в словарь еще одну книгу, с полями, введенными с консоли. Вывести результат на экран.
* Удалить книгу, с самым коротким названием. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

**Вариант 11**

1. Даны две непустые очереди (вывести их на консоль). Элементы каждой из очередей упорядочены по возрастанию. Объединить очереди в одну с сохранением упорядоченности элементов. Вывести полученный результат на экран консоли.
2. Создать класс Мебель, у которого будут поля название (стул, стол и т.д.) с большой буквы и масса в килограммах. Создать словарь, содержащий список Мебели (не менее 10).

* Упорядочить список по убыванию массы. Результат вывести на экран консоли.
* Добавить в словарь еще одну мебель, с полями, введенными с консоли. Вывести результат на экран.
* Удалить мебель, с самой маленькой массой. Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.