

Задачи на лабораторные работы
(Задачи со звездочками необязательные)

Java collections framework

Напишите класс CollectionsDemo со следующим набором методов.

1. Вход: список строк и символ. Выход: количество строк входного списка, у которых первый символ совпадает с заданным.
2. Создайте класс Human с полями: фамилия, имя, отчество, возраст и методами: конструкторы, геттеры/сеттеры, equals и hashCode.
Напишите метода класса ListDemo, который получает на вход список объектов типа Human и еще один объект типа Human. Результат — список однофамильцев заданного человека среди людей из входного списка.
3. Вход: список объектов типа Human и еще один объект типа Human. Выход — копия входного списка, не содержащая выделенного человека. При изменении элементов входного списка элементы выходного изменяться не должны.
4. Вход: список множеств целых чисел и еще одно множество. Выход: список всех множеств входного списка, которые не пересекаются с заданным множеством.
5. Создайте класс Student, производный от Human, новое поле — название факультета, к нему геттер, сеттер и конструктор.
Напишите метод класса ListDemo, который получает на вход список, состоящий из объектов типа Human и его производных классов. Результат — множество людей из входного списка с максимальным возрастом.
6. *По множеству объектов, расширяющих Human, постройте список так, чтобы итератор списка перебирал его элементы по возрастанию ФИО людей без дополнительной сортировки списка.
7. Имеется набор людей, каждому человеку задан уникальный целочисленный идентификатор. Напишите метод, который получает на вход отображение (Map) целочисленных идентификаторов в объекты типа Human и множество целых чисел. Результат — множество людей, идентификаторы которых содержатся во входном множестве.
8. Для отображения из задачи 7 постройте список идентификаторов людей, чей возраст не менее 18 лет.
9. Для отображения из задачи 7 постройте новое отображение, которое идентификатору сопоставляет возраст человека.
10. По множеству объектов типа Human постройте отображение, которое целому числу (возраст человека) сопоставляет список всех людей данного возраста из входного множества.

11. *С использованием решения задачи 10 по множеству людей постройте отображение, которое возрасту сопоставляет новое отображение, которое букве алфавита сопоставляет список всех людей из входного множества, имеющих данный возраст, и фамилия которых начинается на эту букву. Т.е.

(возраст ->

(буква -> <список людей данного возраста с фамилией на эту букву>)

)

Списки внутри должны быть отсортированы по ФИО по убыванию.

12. *Напишите класс PhoneBook (телефонная книга). Книга представляет собой отображение человека в список его номеров телефонов (номер телефона — строка). Методы: добавить телефон, удалить телефон, получить список телефонов по человеку, найти человека по номеру телефона, найти всех людей с их телефонами по началу фамилии человека (результат — новое отображение такой же структуры, но содержащее только отобранные записи).

13. *Имеются целочисленные данные, объединенные в группы. Группы хранятся в виде единого массива.

- 1) Напишите класс Group (группа данных). Класс должен содержать идентификатор группы (целое число) и сами данные (массив целых чисел). Методы: конструкторы, геттеры и сеттеры полей, длина массива. Должна быть возможность создать объект так:

```
Group group = new Group(100, 1, 2, 3); // 100 – номер группы
```

- 2) Напишите класс Data (набор групп). Класс должен содержать название набора (строка символов) и сам набор в виде массива. Методы: конструкторы, геттеры и сеттеры полей, длина массива. Должна быть возможность создать объект так:

```
Data data = new Data("Test data", new Group(...), new Group(...));
```

- 3) Напишите итератор, который для объекта класса Data перебирает все числа во всех его группах. Добавьте в класс Data метод iterator().

- 4) Напишите класс DataDemo со статическим методом getAll, который получает на вход объект класса Data и возвращает список целых чисел. Метод с помощью итератора строит список всех чисел, входящих во все группы данных.