

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Тольяттинский государственный университет»

**Институт Математики, Физики и информационных технологий**

(институт)

Прикладная математика и информатика

(кафедра)

**Практическое задание №10**

по учебному курсу «Разработка приложений на платформе Java 1»

Вариант \_\_\_\_ *(при наличии)*

|  |  |  |
| --- | --- | --- |
| Студент | С.Р.Хакимов  (И.О. Фамилия) |  |
| Группа | ПМИп-1702б  (И.О. Фамилия) |  |
| Преподаватель | E.A. Ерофеева  (И.О. Фамилия) |  |

Тольятти 2020

Ссылка на github: <https://github.com/Rich-alien/java>

**Тема:** Java Collection Framework

**Цели:**

− закрепить знания в области обработки даты и времени;

− научиться использовать технологии Java Collection Framework для работы

с датой и временем.

**Задание 1.**

Исследование контейнера List.

Выполните следующие задания, используя статические методы класса

Collections, и осуществляя контрольный вывод по каждому пункту.

1. Создайте массив из N чисел.
2. На основе массива создайте список List.
3. Отсортируйте список в натуральном порядке.
4. Отсортируйте список в обратном порядке.
5. Перемешайте список.
6. Выполните циклический сдвиг на 1 элемент.
7. Оставьте в списке только уникальные элементы.
8. Оставьте в списке только дублирующиеся элементы.
9. Из списка получите массив.



Рисунок 1-Фрагмент кода задания 1

**Задание 2.**

Генератор простых чисел.

Напишите класс PrimesGenerator, генерирующий заданное количество (N)

простых чисел, и проверочный класс PrimesGeneratorTest, выводящий на

консоль первые N чисел в прямом и обратном порядке. Для этого:

1. Продумайте структуру класса PrimesGenerator, сигнатуру методов и тип возвращаемого значения.
2. Создайте новый класс компаратора для обхода коллекции в обратном
3. порядке (вариант 1).
4. Используйте анонимный класс итератора для обхода в обратном порядке
5. (вариант 2) – проверочный класс должен реализовывать оба варианта.

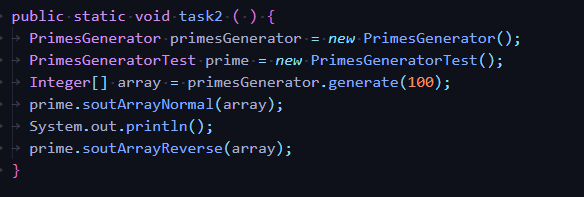


Рисунок 2-Фрагмент кода задания 2

**Задание 3.**

Разработайте набор классов для работы со списком сотрудников.

1. Разработайте класс Human, имеющий поля Имя, Фамилия, Возраст в

соответствии со спецификацией JavaBeanes и требованиям контракта

класса Object. Для того, что бы объекты класса Human могли

сортироваться, объявите в классе Human реализацию интерфейса

Comparable, пример кода:

@Override

public int compareTo(Human o) {

int rez;

if(this.equals(o)) return 0;

rez = fName.compareToIgnoreCase(o.getfName());

if(rez != 0) return rez;

rez = lName.compareToIgnoreCase(o.getlName());

if(rez != 0)

return rez;

else

return age-o.getAge();

}

2. Напишите класс компаратора HumanComparatorByLName

сравнивающий два человека только по Фамилии

3. В проверочном классе создайте коллекцию на основе HashSet, включив

в нее минимум три человека. Выведите коллекцию на консоль, пример

кода

Set<Human> s = new HashSet<Human>();

s.add(new Human("Andy", "Wolf", 25));

s.add(new Human("Ritta", "SKitter", 24));

s.add(new Human("Petr", "Petrov", 31));

System.out.println("HashSet");

System.out.println(s);

4. В проверочном классе создайте коллекцию LinkedHashSet на основе

существующей коллекции s. Выведите коллекцию на консоль.

Сделайте вывод об упорядоченности и сортировке элементов!

5. В проверочном классе создайте коллекцию TreeSet на основе

существующей коллекции s. Выведите коллекцию на консоль.

Сделайте вывод об упорядоченности и сортировке элементов!

6. В проверочном классе создайте пустую коллекцию TreeSet с

компаратором HumanComparatorByLName. Вставьте в новую

коллекцию элементы коллекции s. Выведите коллекцию на консоль.

Сделайте вывод об упорядоченности и сортировке элементов!

7. В проверочном классе создайте пустую коллекцию TreeSet с

анонимным компаратором по возрасту. Вставьте в новую коллекцию

элементы коллекции s. Выведите коллекцию на консоль.



Рисунок 3-Фрагмент кода задания 3

**Задание 4.**

Первая строка – первый вопрос к зачету, вторая – второй. Проведите

частотный анализ слов и символов из первой строки. Определите символы, которые:

* входят и в первую и во вторую строку,
* входят в первую и не входят во вторую,
* содержатся хотя бы в одной строке.

Сделайте те же задания, символы должны выводиться:

* в обычном порядке;
* в обратном порядке;
* в порядке возрастания циклического сдвига влево на n разрядов хеш-
* функции символа.

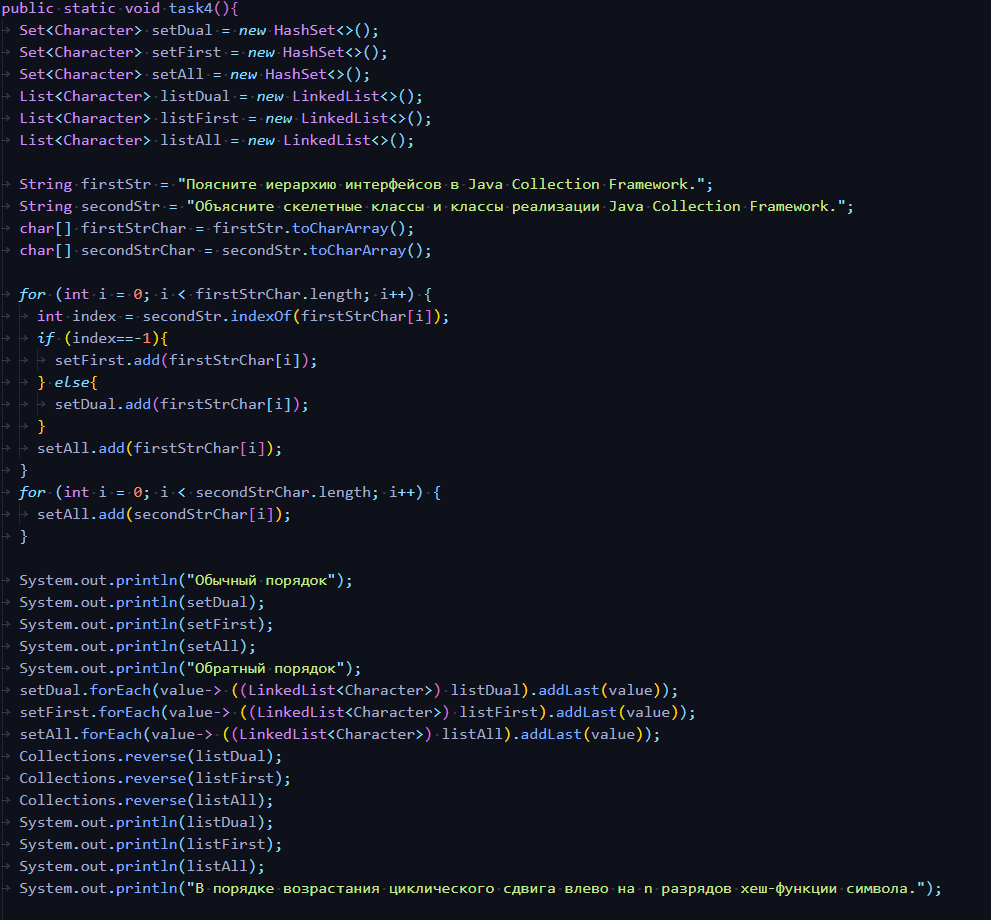


Рисунок 4-Фрагмент кода задания 4