|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 9**

**Вариант № 13**

**Название:** stream API

**Дисциплина:** языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | С.П.Пантелеев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:** изучить stream API в java.

**Задание 1:** использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция чисел. Вернуть сумму нечетных чисел.

Код класса Main:

package Zadanie\_3;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Collection;  
import java.util.stream.Stream;  
  
public class Main {  
 public static void main(String[] args) {  
 Collection<Integer> collection = Arrays.*asList*(1, 2, 3, 4, 5);  
 Stream<Integer> steam = collection.stream();  
  
 int a = steam.filter(o -> o % 2 != 0).reduce((s1, s2) -> s1 + s2).orElse(0);  
  
 System.*out*.println(a);  
 }  
}

Работа программы показана на рисунке 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Работа программы

**Задание 2:** использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция чисел. Разделить числа на четные и нечетные.

Код класса Main:

package Zadanie\_4;  
  
import java.util.Arrays;  
import java.util.Collection;  
import java.util.stream.Stream;  
  
public class Main {  
 public static void main(String[] args) {  
 Collection<Integer> collection = Arrays.*asList*(1, 2, 3, 4, 5);  
 Stream<Integer> steam\_1 = collection.stream();  
 Stream<Integer> steam\_2 = collection.stream();  
 Stream<Integer> steam\_3 = collection.stream();  
  
 double a = steam\_1.filter(o -> o % 2 != 0).reduce((s1, s2) -> s1 + s2).orElse(0);  
 double sum = steam\_2.reduce((s1, s2) -> s1 + s2).orElse(0);  
 double b = steam\_3.filter(o -> o % 2 == 0).reduce((s1, s2) -> s1 + s2).orElse(0);  
 double rez\_1 = sum/a;  
 double rez\_2 = sum/b;  
  
 System.*out*.println("Сумма деленная на четные");  
 System.*out*.println(rez\_1);  
 System.*out*.println("Сумма деленная на нечетные");  
 System.*out*.println(rez\_2);  
 }  
}

Работа программы показана на рисунке 2.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Работа программы

**Задание 3:** использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. 2. Задана коллекция:

(Класс People: имя и возраст)

Collection<People> peoples = Arrays.asList(

new People("Ivan", 16),

new People("Petr", 23),

new People("Maria", 42)

);

Отсортировать по возрасту.

Код класса Main:

package Zadanie\_1;  
  
import java.util.Arrays;  
import java.util.Collection;  
import java.util.Comparator;  
import java.util.List;  
  
class People{  
 public String name;  
 public int age;  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public People(String name, int age) {  
 this.name = name;  
 this.age = age;  
 }  
}  
public class Main {  
 public static void main(String[] args) {  
 Collection<People> people = Arrays.*asList*(  
 new People("Ivan", 16),  
 new People("Petr", 23),  
 new People("Maria", 42)  
 );  
  
 List<People> sortedList = people.stream()  
 .sorted(Comparator.*comparingInt*(People::getAge))  
 .toList();  
  
 sortedList.forEach(System.*out*::println);  
 }  
}

Работа программы показана на рисунке 3.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Работа программы

**Задание 4:** использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция строк. Преобразовать в MAP, где первый символ ключ, второй – значение.

Код класса Main:

package Zadanie\_2;  
  
import java.util.\*;  
import java.util.stream.Collectors;  
import java.util.stream.Stream;  
  
public class Main {  
 public static void main(String[] args) {  
 List<String> collection = Arrays.*asList*("pel=0", "me=1", "ni=2");  
 Stream<String> steam = collection.stream();  
  
 Map<String, String> result = steam  
 .map( e -> e.split("=") )  
 .filter( e -> e.length == 2 )  
 .collect( Collectors.*toMap*(e -> e[0], e -> e[1]) );  
  
 result.entrySet().stream().forEach(System.*out*::println);  
 }  
}

Работа программы показана на рисунке 4.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Работа программы

**Вывод:** во время выполнения лабораторной работы были изучена stream API в java.