|  |  |
| --- | --- |
| 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 **09.04.01/07 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 9**

**Вариант № 12**

**Название:** Stream API

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Т.М. Курохтин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2024

**Цель:** изучить и освоить принципы работы с Stream API в Java.

**Задание 1:** Использовать ТОЛЬКО методы Stream API. 1. Задана коллекция строк. Вернуть последний элемент и третий элемент коллекции.

Код программы:

fun main() {

val strings = listOf("first", "second", "third", "fourth", "fifth")

// Получаем третий элемент (индекс 2)

val thirdElement = strings.drop(2).firstOrNull()

// Получаем последний элемент

val lastElement = strings.drop(strings.size - 1).firstOrNull()

println("Third element: $thirdElement")

println("Last element: $lastElement")

}

Результат работы программы представлен на рисунке 1.

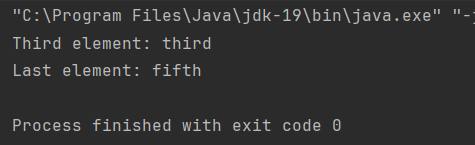


Рисунок 1 – Результат работы программы

**Задание 2:** Использовать ТОЛЬКО методы Stream API. 2. Задана коллекция строк. Вернуть первый элемент коллекции, а также существуют ли все совпадения с шаблоном. Шаблон можно выбрать произвольно.

Код программы:

fun main() {

val collection = listOf("Hello123", "World456", "NoDigitsHere", "Another789")

val pattern = Regex("\\d+")

val firstElement = collection.firstOrNull()

val allMatchPattern = collection.all { it.contains(pattern) }

println("Первый элемент коллекции: $firstElement")

println("Все ли строки совпадают с шаблоном: $allMatchPattern")

}

Результат работы программы показан на рисунке 2.

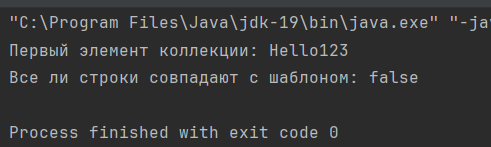


Рисунок 2 – Результат работы программы

**Задание 3:** Использовать ТОЛЬКО методы Stream API. Задана коллекция строк. Отсортировать значения по алфавиту и убрать повторы.

Код программы:

fun main() {

val collection = listOf("Banana", "Apple", "Banana", "Cherry", "Apple", "Date")

val sortedDistinctCollection = collection.distinct().sorted()

println("Отсортированная коллекция без повторов: $sortedDistinctCollection")

}

Результат работы программы представлен на рисунке 3.

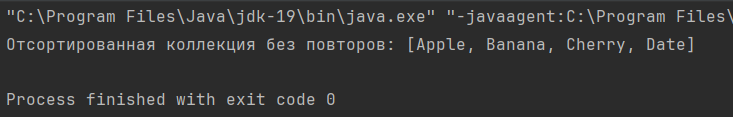


Рисунок 3 – Результат работы программы

**Задание 4:** Использовать ТОЛЬКО методы Stream API. 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 имени в обратном алфавитном порядке..

Код программы:

data class People(val name: String, val age: Int)

fun main() {

val peoples = listOf(

People("Ivan", 16),

People("Petr", 23),

People("Maria", 42)

)

val sortedPeoples = peoples.sortedByDescending { it.name }

println("Отсортированная коллекция по имени в обратном алфавитном порядке:")

for (person in sortedPeoples) {

println("${person.name}, ${person.age}")

}

}

Результат работы программы представлен на рисунке 4.

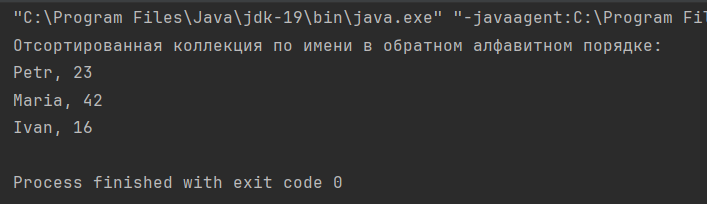


Рисунок 4 – Результат работы программы

**Вывод:** были освоены принципы работы со Stream API в Java.