Оленчикова Т.Ю.

**Методические указания и задачи к выполнению лабораторной работы по дисциплине «**Программирование на языке Java для анализа данных**»**

**Лабораторная работа 5 Лямбда-выражения, StreamIP**

Цель: Научиться разрабатывать программы на Java, используя потоковцю обработку данных (framework StreamIP), получить навыки работы с функциональными интерфейсами

**Теоретические материалы**

## [Обобщенные функциональные интерфейсы и лямбда-выражения](https://www.bestprog.net/ru/2020/12/08/java-lambda-expressions-basic-concepts-functional-interface-generalized-functional-interfaces-and-lambda-expressions-examples-ru/)

## Глубокое [погружение](https://vc.ru/u/1389654-machine-learning/665389-glubokoe-pogruzhenie-v-lyambda-vyrazheniya-na-java) в лямбда-выражения на Java

## [Потоки данных. Stream API](https://www.bestprog.net/ru/2020/12/25/java-data-streams-stream-api-general-information-ru/)

**Индивидуальные задания для лабораторной работы.**

В работе два задания:

1 Использование лямбда-выражений

2. Потоковая обработка данных StreamIP.

**Задание 1 Лямбда-выражения**

В задании 1, согласно варианту задания, требуется:

1) создать интерфейс и класс, при необходимости. В методе main главного класса программы Program1 требуется создать лямбда-выражение, согласно варианту, и продемонстрировать его работу;

2) написать ответ на контрольный вопрос в соответствии с вашим вариантом

|  |  |
| --- | --- |
| Вар=т | Задание |
| 1 | Написать интерфейс Expression, который содержит метод. Этот метод принимает в качестве параметров два вектора и возвращает вектор. Создать класс Expressions, содержащий статические методы: сложение векторов, разность векторов. Создать метод operation, который принимает в качестве параметров ссылку на элемент интерфейса Expression (выражение) и два вектора и который выполняет данное выражение. В главном классе программы вызвать методы класса Expressions, передав выражение и вектора методу operation в качестве параметров. |
| 2 | Написать интерфейс, который содержит метод. Этот метод принимает в качестве параметра строку и возвращает строку. Создать метод, возвращающий лямбда-выражение различных операций над строкой: удаление подряд идущих одинаковых символов, инвертирование содержимого строки, инвертирование регистра строки. В главном классе программы продемонстрировать выполнение каждой операции. |
| 3 | Написать интерфейс, который содержит обобщенный метод. Метод принимает в качестве параметра массив значений и возвращает значение указанного типа. Создать два лямбда-выражения: нахождение минимального элемента массива, сумма положительных и отрицательных элементов вещественного массива. |
| 4 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметра массив строк и возвращает сроку. Создать лямбда-выражение, находящее строку с наибольшим количеством символов в верхнем регистре. |
| 5 | Написать интерфейс Expression, который содержит метод. Метод принимает в качестве параметров две строки и возвращает строку. Создать класс Expressions, содержащий статические методы: объединения строк, удаление из первой строки всех символов, содержащихся во второй строке. Создать метод operation, принимающий в качестве параметров ссылку на элемент интерфейса Expression, массив строк, и выполняющий выражение для соседних элементов массива. В главном классе программы объявить и инициализировать массив строк и вызвать методы класса Expressions, передав метод и массив методу operation в качестве параметров. |
| 6 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметров два целых числа и возвращает целое число. Создать метод, возвращающий лямбда-выражение различных операций между двумя числами. В главном классе программы продемонстрировать выполнение каждой операции. |
| 7 | Написать интерфейс, который содержит обобщенный метод для целочисленного типа. Метод принимает в качестве параметра массив значений и возвращает значение указанного типа. Создать два лямбда-выражения: подсчет суммы по модулю два элементов целочисленного массива, вычисление произведения элементов вещественного массива. |
| 8 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметров два вектора целых чисел и возвращает целое число. Создать лямбда-выражение, вычисляющее сумму элементов, образованных в результате произведения первого транспонированного вектора на второй. |
| 9 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметров матрицу целых чисел, целое число и возвращает целое число. Создать лямбда-выражение, вычисляющее сумму чисел из столбца матрицы с номером, переданным в качестве второго параметра. |
| 10 | Написать интерфейс, который содержит обобщенный метод. Метод принимает в качестве параметра массив значений и возвращает значение указанного типа. Создать два лямбда-выражения: объединение строк массива в одну, нахождение самой короткой строки. |
| 11 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметра массив целых чисел и возвращает истину или ложь. Создать лямбда-выражение, определяющее принадлежность массива последовательности Фибоначчи. |
| 12 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметра матрицу целых чисел и возвращает матрицу целых чисел. Создать лямбда-выражение, осуществляющее сортировку строк матрицы в зависимости от значения произведения элементов строки. |
| 13 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметров два массива строк (оба массива одинакового размера) и возвращает словарь. Создать лямбда-выражение, сопоставляющее слову перевод. |
| 14 | Написать интерфейс, который содержит метод. Метод принимает в качестве параметров массив строк, строку и возвращает строку. Создать лямбда-выражение, объединяющее массив строк в одну, используя в качестве разделителя другую строку. |
| 15 | Написать интерфейс, который содержит метод. Метод принимает в качестве аргументов массив целых чисел и возвращает массив вещественных чисел. Создать лямбда-выражение, инвертирующее порядок элементов массива и приводящее их к вещественному типу. |

**Контрольные вопросы**

1. Какие требования предъявляются к интерфейсу для использования его в качестве лямбда-выражения?
2. Как осуществляется передача параметров в лямбда-выражение?
3. Как локальные переменные могут быть использованны в лямбда-выражениях?
4. Как используется интерфейс с обобщенными типами в лямбда-выражениях?
5. Как лямбда-выражение может быть использованно в качестве параметра метода?
6. Как лямбда-выражение может быть использованно в качестве ссылки на метод в параметрах метода?
7. Как с помощью лямбда-выражения можно создавать объекты класса?
8. Как лямбда-выражения могут быть использованы в качестве возвращаемых значений метода?
9. Как использование лямбда-выражений может повысить производительность крупномасштабного приложения для обработки данных и каковы некоторые рекомендации по достижению этого?
10. Как использование лямбда-выражений может улучшить читаемость и ремонтопригодность приложения, и каковы некоторые рекомендации для достижения этого?
11. Как использование потоков в сочетании с лямбда-выражениями может повысить производительность задач обработки данных в Java? Приведите пример того, как этого можно достичь.

**Задание 2 StreamIP**

В ходе выполнения лабораторной работы должно быть реализовано:

1) Класс Human(int age, String firstName, String lastName, LocalDate birthDate, int weight);

2) приложение, которое создает список из объектов класса Human, а затем производит действия в соответствии с вариантом индивидуального задания (список после каждого этапа должен выводиться в консоль).

Все действия должны производиться только с использованием StreamIP

|  |  |
| --- | --- |
| Вар=т | Задание |
| 1 | Сортировка по имени; фильтрация по дате рождения большей, чем 24 июня 2000; сортировка по фамилии; нахождение суммы всех возрастов. |
| 2 | Сортировка по дате рождения; фильтрация по возрасту меньше, чем 50; сортировка по весу; конкатенация всех имен в одну большую строку через пробел. |
| 3 | Сортировка по весу в обратном порядке; фильтрация по фамилии “не Иванов”; сортировка по возрасту, произведение всех возрастов. |
| 4 | Сортировка по второй букве имени; фильтрация по весу кратно 10; сортировка по произведению веса на возраст; произведение всех весов. |
| 5 | Сортировка по возрасту в обратном порядке; фильтрация по имени «начинается с А»; сортировка по дате рождения; расчет среднего веса. |
| 6 | Уменьшение веса каждого объекта на 5; фильтрация по дате рождения меньшей, чем 3 февраля 1999; конкатенация фамилий в строку через пробел. |
| 7 | Выбор первых 5 элементов списка; сортировка по дате рождения от старых к новым; фильтрация по весу меньше, чем 60; вывод имени и фамилии через пробел. |
| 8 | Фильтрация по возрасту больше чем 20; сортировка по последней букве имени; увеличение возраста каждого на 3; вычисление среднего возраста всех элементов. |
| 9 | Фильтрация по признаку «вес больше, чем возраст»; сортировка по фамилии в обратном порядке; сумма всех весов. |
| 10 | Сортировка по второй букве имени в обратном порядке; фильтрация по весу больше, чем 60; сортировка по возрасту; произведение всех возрастов |
| 11 | Сортировка по имени в обратном порядке; фильтрация по возрасту больше. чем 20; выбор первых 3 элементов списка; конкатенация имен в строку через пробел. |
| 12 | Сортировка по последней букве фамилии; фильтрация по признаку «возраст больше, чем вес»; сортировка по дате рождения; произведение всех возрастов. |
| 13 | Сортировка по возрасту; фильтрация по возрасту меньше, чем 20; фильтрация по имени «содержит ‘е`»; конкатенация первых букв имен. |
| 14 | Сортировка по сумме веса п возраста; фильтрация по весу кратно 5; выбор первых четырёх элементов; конкатенация имён через пробел. |
| 15 | Увеличение веса каждого объекта на 3; сортировка по весу в обратном порядке: фильтрация по дате рождения меньшей, чем 01.01.2000; сумма всех весов. |

**Контрольные вопросы**

1. Какие есть методы в интерфейсе Stream?
2. Чем отличается метод map от flatMap?
3. Какой функциональный интерфейс использует метод filter?
4. Какие существуют способы создания стрима?
5. В чем разница между Collection и Stream?
6. Для чего нужен метод collect() и Stream?
7. Для чего в Stream применяются методы forEach() и forEachOrdered()?
8. Для чего в Stream предназначены методы map() и mapToInt(), mapToDouble(), mapToLong()?
9. Какова цель метода filter() в Stream?
10. Для чего в Stream предназначен метод limit()?
11. Для чего в Stream предназначен метод sorted()?
12. Для чего в Stream предназначены методы flatMap(), flatMapToInt(), flatMapToDouble(), flatMapToLong()?
13. Какие конечные методы работы со Stream вы знаете?
14. Какие промежуточные методы работы со Stream вы знаете?

**Требования к оформлению отчета.**

Отчет по ЛР состоит из

**а)** Титульный лист: название ЛР, ФИО студента и номер группы, № варианта.

б)Текст задания; в) UML-диаграмму классов (создаете в Umlet):

г) Исходный текст программы;

д) Скрины выполнения;

е) Выводы.

**Критерии оценивания.**

За решение каждой задачи вы можете получить до 2 баллов: задание выполнено полностью и правильно -2 балла; имеются незначительные ошибки -1 балл; приложение не работоспособно – 0 баллов;

Ответ на контрольный вопрос должен продемонстрировать понимание механизмов Java, за ответ вы можете оценку: ответ полный и правильный -2 балл; ответ не полный – 1 балл, ответ не удовлетворительный – 0 баллов.

Итого, максимальная оценка - 6 баллов

**Внимание!** Полученная оценка автматически снижается на 2% за каждую полную неделю задержки сдачи отчета по работе, но не более, чес на 40%

**Библиографический список**

1. Программирование на языке Java. Конспект лекций : учебно-методическое пособие / А. В. Гаврилов, С. В. Клименков, Ю. А. Королёва [и др.]. — Санкт-Петербург : НИУ ИТМО, 2019. — С. 5-17, 26-35. — URL: https://e.lanbook.com/book/136549 (дата обращения: 21.08.2023). — Режим доступа: для авториз. пользователей.
2. Пономарчук, Ю. В. Программирование на языке Java : учебное пособие / Ю. В. Пономарчук, И. В. Кузнецов. — Хабаровск : ДВГУПС, 2021. — 103 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/259451 (дата обращения: 21.08.2023). — Режим доступа: для авториз. пользователей.

**Электронные ресурсы**

1. Руководство по языку программирования Java [Электронный ресурс]. – URL: <https://metanit.com/java/tutorial//> – Свободный доступ.
2. Курсы, статьи по Java. – URL: <https://javarush.com/> – Свободный доступ.
3. Учебное пособие по Java [Электронный ресурс]. – URL: <https://www.bestprog.net/ru/sitemap_ru/java/> – Свободный доступ.