

**The course “Application development with Java”**

Practical assignment 3

Student name: Roberts

Student surname: Dubovskis

St. code: 62085

**RIGA**

**2022**

Contents

[Exercise 1 3](#_Toc99360641)

[Exercise 2 3](#_Toc99360642)

[Exercise 3 4](#_Toc99360643)

[Exercise 4 4](#_Toc99360644)

[Exercise 5 5](#_Toc99360645)

[Conclusion 9](#_Toc99360646)

# Exercise 1



Img.1 (result)

package com.company.lamda;  
  
import java.util.List;  
  
public class ListProcessingWithStreams {  
  
 public static void main(String[] args) {  
   
 List<String> list = List.*of*("a1", "c3", "a2", "a3","b3", "b2", "c1", "c2","b1");  
 System.*out*.printf("First element %s", list.get(0));

}  
}

# Exercise 2



Img.2 (result)

package com.company.lamda;

import java.util.List;

public class ListProcessingWithStreams {

public static void main(String[] args) {

}  
}

# Exercise 3



Img.3 (result)

package com.company.lamda;

import java.util.List;

public class ListProcessingWithStreams {

public static void main(String[] args) {

}  
}

# Exercise 4



Img.4 (result)

package com.company.lamda;

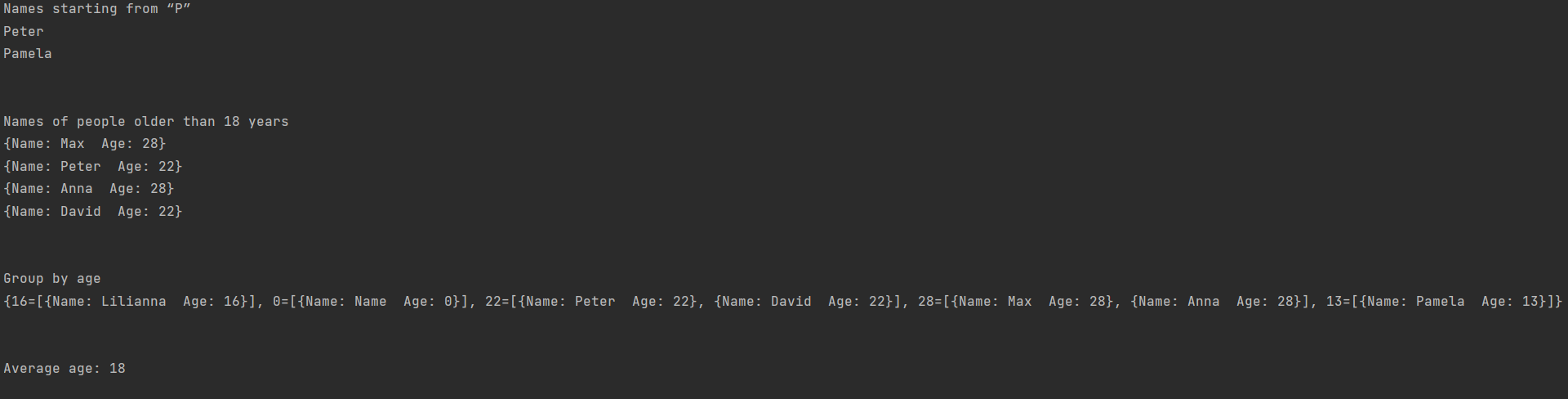
import java.util.List;

public class ListProcessingWithStreams {

public static void main(String[] args) {

}  
}

# Exercise 5



Img.5 (result)

**Person.java**

package com.company.lamda;

public class Person {

private String name;

private int age;

public Person() {

}//default constructor

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return this.name;

}

public int getAge() {

return this.age;

}

@Override

public String toString() {

return "{" +

"Name: " + name +

"Age: " + age +

"}";

}

}

**PersonEvaluator.java**

package com.company.lamda;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import java.util.stream.Collectors;

public class PersonEvaluator {

private final List<Person> list = new ArrayList<>();

public PersonEvaluator(List<Person> a) {

this.list.addAll(a);

}

public void findAndPrint(int a) {

System.out.println("Names of people older than 18 years");

list.stream()

.filter(l -> l.getAge() > a)

.collect(Collectors.toList())

.forEach(System.out::println);

}

public void findAndPrint() {

System.out.println("Names starting from “P”");

list.stream()

.map(Person::getName)

.filter(l -> l.charAt(0) == 'P')

.collect(Collectors.toList())

.forEach(System.out::println);

}

public void groupAndPrint() {

System.out.println("Group by age");

Map<Integer, List<Person>> a = list.stream()

.collect(Collectors.groupingBy(Person::getAge));

System.out.println(a);

}

public void getAverageAge() {

var buf = list.stream()

.mapToInt(Person::getAge)

.sum();

System.out.println("Average age: " + buf / list.size());

}

}

**PersonProcessingWithStreams.java**

package com.company.lamda;

import java.util.ArrayList;

import java.util.List;

public class PersonProcessingWithStreams {

public static void main(String[] args) {

List<Person> list = new ArrayList<>();

list.add(new Person("Name", 0));

list.add(new Person("Max", 28));

list.add(new Person("Peter", 22));

list.add(new Person("Anna", 28));

list.add(new Person("Lilianna", 16));

list.add(new Person("Pamela", 13));

list.add(new Person("David", 22));

PersonEvaluator personEvaluator = new PersonEvaluator(list);

personEvaluator.findAndPrint();

System.out.println("\n");

personEvaluator.findAndPrint(18);

System.out.println("\n");

personEvaluator.groupAndPrint();

System.out.println("\n");

personEvaluator.getAverageAge();

}

}

# Conclusion

All tasks have been accomplished.