Частное учреждение образования

«Колледж бизнеса и права»

Лабораторная работа №5

Учащийся Е.А. Чужавко.

группа: Т-992

2021

Текст задачи 1

Создайте HashMap, содержащий пары значений - имя игрушки и объект игрушки (класс Product). Перебрать и распечатать пары значений - entrySet(). Перебрать и распечатать набор из имен продуктов - keySet(). Перебрать и распечатать значения продуктов - values(). Для каждого перебора создать свой метод.

// Main и функции

package core;

import java.util.HashMap;

import java.util.ArrayList;

import Classes.Cat;

import Classes.Dog;

import Classes.Parrot;

import Classes.Pet;

import Classes.Product;

import Classes.Student;

public class class\_laba5 {

// My metods

public static void HashMapForEntrySet(HashMap<String, Product> x)

{

System.out.print("Вызвал в цикле для каждого entrySet()");

System.out.println("\n==1==\n" + x.entrySet());

}

public static void HashMapForKeySet(HashMap<String, Product> x)

{

System.out.print("Вызвал в цикле для каждого keySet()");

System.out.println("\n==2==\n" + x.keySet());

}

public static void printValueHashMap(HashMap<String, Product> x)

{

System.out.println("==3==\nprintValueHashMap");

System.out.println(x.values());

}

// for 2 task

public static void SortStudent(ArrayList<Student> listStudent)

{

// Сначала нахожу средний балл

int sredn = 0;

for (int i = 0; i < listStudent.size(); i++)

{

sredn = 0; // Зануляю для каждого студента

for (int j = 0; j < listStudent.get(i).marks.size(); j++) // listStudent.get(i).marks.size() Проверяю каждый список отметок у каждого студента

{

sredn += listStudent.get(i).marks.get(j);

}

// Посчитал

sredn /= listStudent.get(i).marks.size();

if (sredn < 3)

{

listStudent.remove(i);

}

else

{

// Значит больше или равно 3 // >=3

if (listStudent.get(i).getNumKursa() < 4) // Проверка зачем его отправлять на 5 курс если например всего 4

{

listStudent.get(i).setNumKursa(listStudent.get(i).getNumKursa()+1);

}

}

}

}

// task 2.2

public static void printStudentInList\_vibor\_pocursu(ArrayList<Student> listStudent, int numCourse)

{

for (int i = 0; i < listStudent.size(); i++)

{

if (listStudent.get(i).getNumKursa() == numCourse) // Если совпадает

{

System.out.println("Student name: " + listStudent.get(i).getName());

}

}

}

// for task 3

public static void pet\_\_HashMapForKeySet(HashMap<String, Pet> x)

{

System.out.println(x.keySet());

}

// main

public static void main(String[] args) {

//////////////////////////////////// 1

System.out.println("Task 1");

HashMap<String, Product> myHashMapProducts = new HashMap<String, Product>();

// Добавлю 3 продукта

Product p1 = new Product(1);

Product p2 = new Product(2);

Product p3 = new Product(3);

myHashMapProducts.put("orange", p1);

myHashMapProducts.put("banana", p2);

myHashMapProducts.put("tomato", p3);

// EntrySet, KeySet

HashMapForEntrySet(myHashMapProducts);

HashMapForKeySet(myHashMapProducts);

// print values

printValueHashMap(myHashMapProducts);

//////////////////////////////////// 2

System.out.println("\nTask 2");

// Список студентов

ArrayList<Student> listStudent = new ArrayList<Student>();

// Первый студент и его оценки

ArrayList<Integer> listMarksMatvey = new ArrayList<Integer>();

// {9, 8, 9, 10};

listMarksMatvey.add(9);

listMarksMatvey.add(8);

listMarksMatvey.add(9);

listMarksMatvey.add(10);

Student st1 = new Student("Matvey", 992, 3, listMarksMatvey);

// Второй студент и его оценки

ArrayList<Integer> listMarksNikita = new ArrayList<Integer>();

// {3, 2, 1, 2};

listMarksNikita.add(3);

listMarksNikita.add(2);

listMarksNikita.add(1);

listMarksNikita.add(2);

Student st2 = new Student("Nikita", 992, 3, listMarksNikita);

// Третий студент и его оценки

ArrayList<Integer> listMarksPeta = new ArrayList<Integer>();

// {10, 5, 8, 9};

listMarksPeta.add(10);

listMarksPeta.add(5);

listMarksPeta.add(8);

listMarksPeta.add(9);

Student st3 = new Student("Peta", 992, 3, listMarksPeta);

// Добавляю самих студентов

listStudent.add(st1);

listStudent.add(st2);

listStudent.add(st3);

// Тест функций

SortStudent(listStudent);

printStudentInList\_vibor\_pocursu(listStudent, 3); // Матвея не распечатает т.к он перевелся на next курс

//////////////////////////////////// 3

System.out.println("\nTask 3");

HashMap<String, Pet> myHashMapPets = new HashMap<String, Pet>();

Cat pet1 = new Cat("Masyanya");

Dog pet2 = new Dog("Sharik");

Parrot pet3 = new Parrot("Parrot\_1");

myHashMapPets.put(pet1.getName(), pet1);

myHashMapPets.put(pet2.getName(), pet2);

myHashMapPets.put(pet3.getName(), pet3);

pet\_\_HashMapForKeySet(myHashMapPets);

}

}

// product

package Classes;

public class Product<T> {

private T id;

public Product(T id)

{

this.setId(id);

}

public T getId() {

return id;

}

public void setId(T id) {

this.id = id;

}

}

// Student

package Classes;

import java.util.ArrayList;

public class Student {

private String name;

private int numGroup;

private int numKursa;

// Список оценок

public ArrayList<Integer> marks = new ArrayList<Integer>();

public Student(String name, int numGroup, int numKursa, ArrayList<Integer> marks)

{

this.setName(name);

this.setNumGroup(numGroup);

this.setNumKursa(numKursa);

this.marks = marks;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getNumKursa() {

return numKursa;

}

public void setNumKursa(int numKursa) {

this.numKursa = numKursa;

}

public int getNumGroup() {

return numGroup;

}

public void setNumGroup(int numGroup) {

this.numGroup = numGroup;

}

}

// pet

package Classes;

public class Pet {

private String name;

public Pet(String name)

{

this.setName(name);

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

// три одинаковые

package Classes;

public class Dog extends Pet{

public Dog(String name) {

super(name);

// TODO Auto-generated constructor stub

}

}

Текст задачи 2

Создать класс Student, содержащий следующие характеристики – имя, группа, курс, оценки по предметам. Создать коллекцию, содержащую объекты класса Student. Написать метод, который удаляет студентов со средним баллом =3, студент переводится на следующий курс. Напишите метод printStudents(List students, int course), который получает список студентов и номер курса. А также печатает на консоль имена тех студентов из списка, которые обучаются на данном курсе.

Текст задачи 3

Создайте класс Pet и его наследников Cat, Dog, Parrot. Создайте отображение из домашних животных, где в качестве ключа выступает имя животного, а в качестве значения класс Pet. Добавьте в отображение разных животных. Создайте метод выводящий на консоль все ключи отображения.

Результат работы программы показан на рисунке 1.

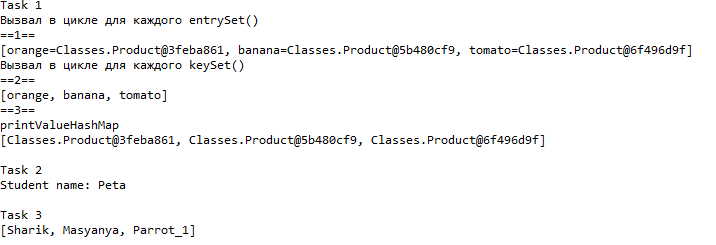


Рисунок 1