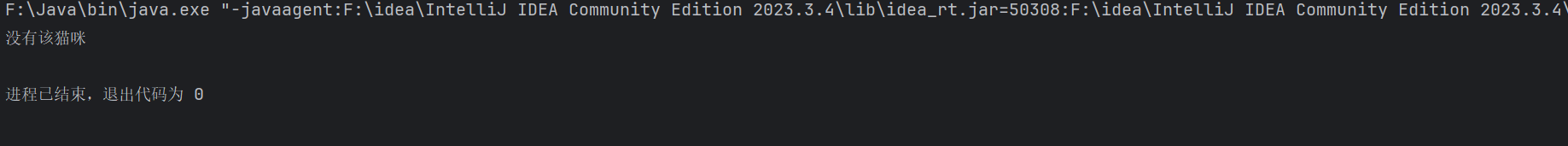
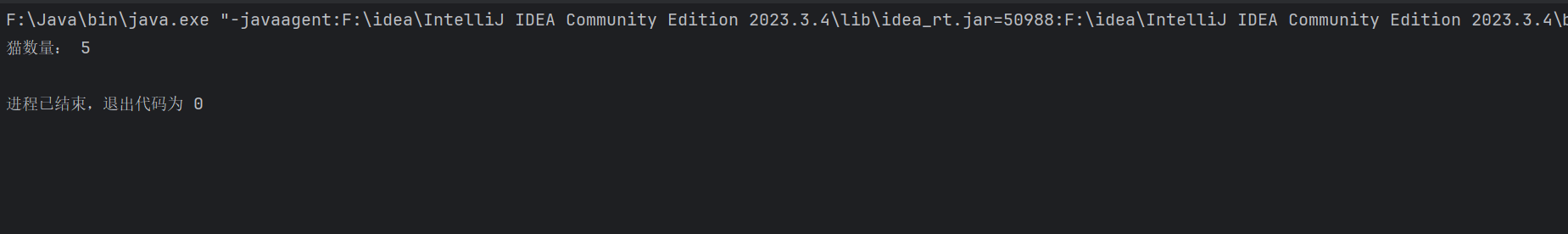
**补充作业**

1.使用ArrayList集合存储5个Cat类对象，并查找是否包含目标对象，如果存在则删除，不存在则输出“没有该猫咪”

1. import java.util.ArrayList;
2. class Cat {
3. private String name;
4. public Cat(String name) {
5. this.name = name;
6. }
7. public String getName() {
8. return name;
9. }
10. public void setName(String name) {
11. this.name = name;
12. }
13. }
14. public class Main {
15. public static void main(String[] args) {
16. ArrayList<Cat> cats = new ArrayList<>();
17. cats.add(new Cat("Tom"));
18. cats.add(new Cat("Jerry"));
19. cats.add(new Cat("Garfield"));
20. cats.add(new Cat("Tony"));
21. cats.add(new Cat("Max"));
22. Cat targetCat = new Cat("Garfield");
23. if (cats.contains(targetCat)) {
24. cats.remove(targetCat);
25. System.out.println("已删除目标猫咪：" + targetCat.getName());
26. } else {
27. System.out.println("没有该猫咪");
28. }
29. }
30. }

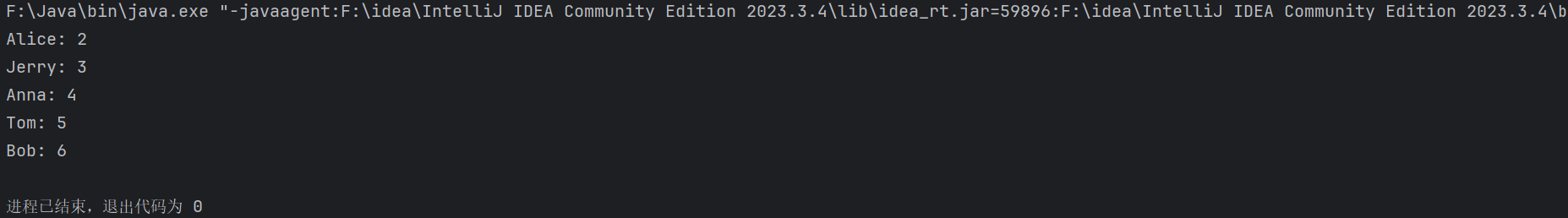


1. 使用HashSet集合存储5个Cat类对象，给出Cat类的equals和hashCode函数，实现相同猫的判断。
2. import java.util.HashSet;
3. import java.util.Objects;
4. class Cat {
5. private String name;
6. private int age;
7. public Cat(String name, int age) {
8. this.name = name;
9. this.age = age;
10. }
11. @Override
12. public boolean equals(Object o) {
13. if (this == o) return true;
14. if (o == null || getClass() != o.getClass()) return false;
15. Cat cat = (Cat) o;
16. return age == cat.age && Objects.equals(name, cat.name);
17. }
18. @Override
19. public int hashCode() {
20. return Objects.hash(name, age);
21. }
22. }
23. public class Main {
24. public static void main(String[] args) {
25. HashSet<Cat> cats = new HashSet<>();
26. cats.add(new Cat("Tom", 3));
27. cats.add(new Cat("Jerry", 2));
28. cats.add(new Cat("Anna", 1));
29. cats.add(new Cat("Bob", 4));
30. cats.add(new Cat("Alice", 5));
31. System.out.println("猫数量： " + cats.size());
32. }
33. }



使用TreeSet集合存储5个Cat类对象，给出Cat类的compareTo函数，实现按照体重排序。

1. import java.util.TreeSet;
2. class Cat implements Comparable<Cat> {
3. String name;
4. int weight;
5. Cat(String name, int weight) {
6. this.name = name;
7. this.weight = weight;
8. }
9. @Override
10. public int compareTo(Cat o) {
11. return Integer.compare(this.weight, o.weight);
12. }
13. }
14. public class Main {
15. public static void main(String[] args) {
16. TreeSet<Cat> cats = new TreeSet<>();
17. cats.add(new Cat("Tom", 5));
18. cats.add(new Cat("Jerry", 3));
19. cats.add(new Cat("Anna", 4));
20. cats.add(new Cat("Bob", 6));
21. cats.add(new Cat("Alice", 2));
22. for (Cat cat : cats) {
23. System.out.println(cat.name + ": " + cat.weight);
24. }
25. }
26. }



使用HashMap存储5个Cat类对象及每只猫咪对应的品种，实现遍历输出，查找修改品种信息等操作。

1. import java.util.HashMap;
2. import java.util.Map;
3. class Cat {
4. private String name;
5. private int age;
6. private String breed;
7. public Cat(String name, int age, String breed) {
8. this.name = name;
9. this.age = age;
10. this.breed = breed;
11. }
12. public String getName() {
13. return name;
14. }
15. public void setName(String name) {
16. this.name = name;
17. }
18. public int getAge() {
19. return age;
20. }
21. public void setAge(int age) {
22. this.age = age;
23. }
24. public String getBreed() {
25. return breed;
26. }
27. public void setBreed(String breed) {
28. this.breed = breed;
29. }
30. @Override
31. public String toString() {
32. return "Cat{" +
33. "name='" + name + '\'' +
34. ", age=" + age +
35. ", breed='" + breed + '\'' +
36. '}';
37. }
38. }
39. public class Main {
40. public static void main(String[] args) {
41. *// 创建一个HashMap来存储猫咪的名称和对应的Cat对象*
42. Map<String, Cat> cats = new HashMap<>();
43. *// 添加猫咪信息到HashMap中*
44. cats.put("Tom", new Cat("Tom", 3, "Persian"));
45. cats.put("Jerry", new Cat("Jerry", 2, "Siamese"));
46. cats.put("Anna", new Cat("Anna", 1, "Maine Coon"));
47. cats.put("Bob", new Cat("Bob", 4, "Ragdoll"));
48. cats.put("Alice", new Cat("Alice", 5, "Bengal"));
49. *// 打印所有猫咪的信息*
50. System.out.println("所有猫咪的信息：");
51. for (Cat cat : cats.values()) {
52. System.out.println(cat);
53. }
54. *// 查找特定猫咪的品种并打印*
55. System.out.println("查找Tom的品种： " + cats.get("Tom").getBreed());
56. *// 修改特定猫咪的品种并打印*
57. System.out.println("修改Bob的品种为Sphynx");
58. cats.get("Bob").setBreed("Sphynx");
59. System.out.println("Bob的新品种： " + cats.get("Bob").getBreed());
60. }
61. }

