|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт комплексной безопасности и специального приборостроения**

**Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»**

**ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**По дисциплине «Кроссплатформенная среда исполнения программного обеспечения»**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы: | «\_\_» \_\_\_\_ 2020 г. | \_\_\_\_\_\_\_\_\_\_\_  (Подпись) | Турченков В.С. |

Москва 2020 г.

**Содержание**

1. **Задания на практическую работу JAVA №2**
2. **Скриншоты выполнения практического занятия 2**
3. **Скриншоты выполнения программы**
4. **Приложение А**

**Задания на практическую работу JAVA №2**

**Правила выполнения:**

1. Создайте новый пакет com.номергруппы.фамилия
2. Пример com.bsbo.00.00.ivanov или com.bsbo\_00\_00.ivanov
3. Внутри пакета для каждого задания создайте свой пакет
4. Пример com.bsbo.00.00.ivanov

           |

        |

        |\_ com.bsbo.00.00.ivanov.task1

1. Выполнение каждого задания должно вызываться из метода main в классе Main.Java в пакете верхнего уровня.

Заполните массив случайным числами и выведете максимальное, минимальное и среднее значение.

**Задание 1.**

Создать класс “кошачьи”. В классе “кошачьи” создать трёх наследников (напр. Лев, Тигр, Гепард и т.п.).

Создать объект из класса “кошачьи”. Вывести одного из наследников и полученный объект.

**Задание 2.**

Создать класс “кошка” (наследника “кошачьи”), который описывает имя кошки и кошку-маму. Создать два объекта: кошку-дочь и кошку-маму. Вывести их на экран.

Далее дополнить класс, добавив кота-папу. Создать 6 объектов: дедушку (папин папа), бабушку (мамина мама), папу, маму, сына, дочь.

**Задание 3.**

В классе “кошка” добавляем статическую переменную cats. Пусть при каждом создании кота (нового объекта) в переменную cats добавляется этот новый кот. Нам нужно создать 10 объектов “кошка”, а затем метод printCats должен выводить всех котов на экран.

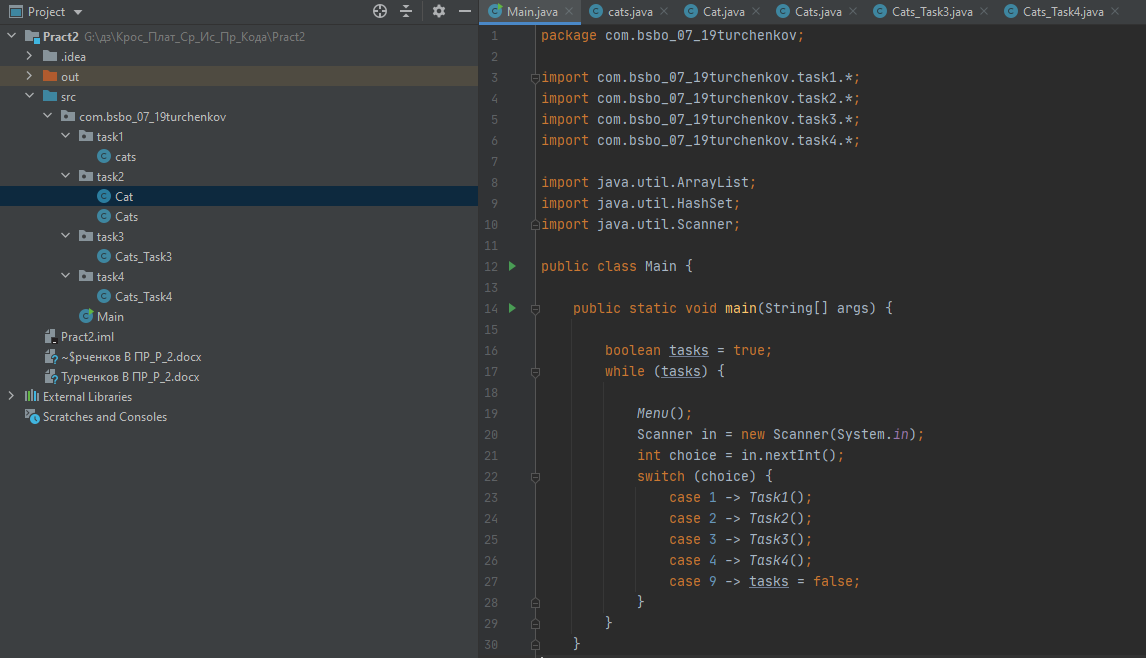
**Задание 4.**

В классе “кошка” реализуйте метод createCats, который должен создавать Set из котов и добавлять в него 3 котов (объекты класса). В методе main удалите одного кота из Set cats. И, наконец, реализуйте метод printCats, он должен вывести на экран всех котов, которые остались во множестве.

(можно использовать класс HashSet)

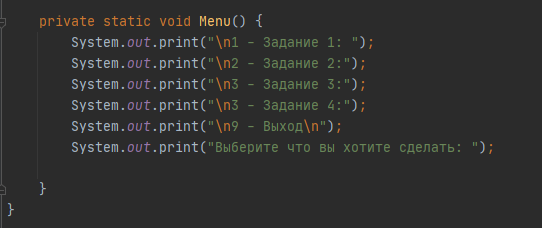
**Скриншоты выполнения практического занятия 2**

Основной код вызова всех классов с заданиями и методы этих классов представлены в каждом пакете. В main, подключаются эти пакеты и все классы в них. Все классы и пакеты представлены в колонке слева.

****

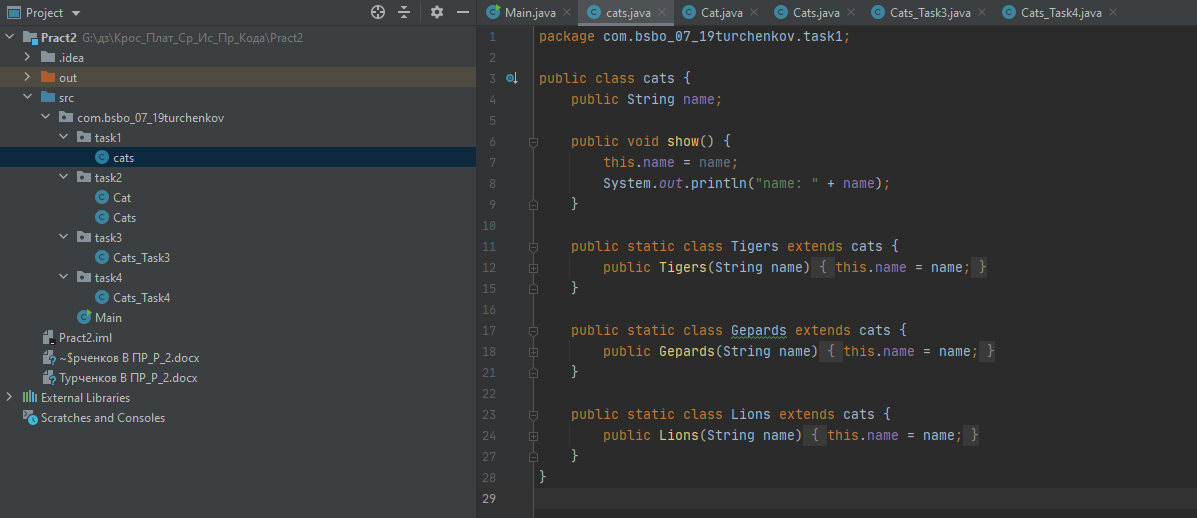
Листинг 1 Вызов классов и их методов

Вызов этих классов происходит с помощью оператора switch case, для удобства используется отдельный метод Menu().

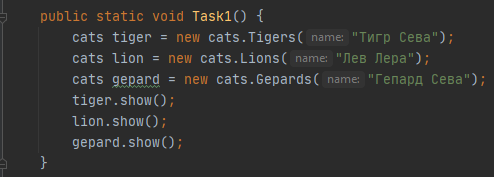


Листинг 2 Меню вызовов заданий

Ниже представлен код выполнения первого задания, сначала задается основной класс “cats”. От него наследуются последующие подклассы Tigers, Gepards, Lions, которые наследуют переменную name. Так же еще описан метод show, который выводит информацию о веденных животных.

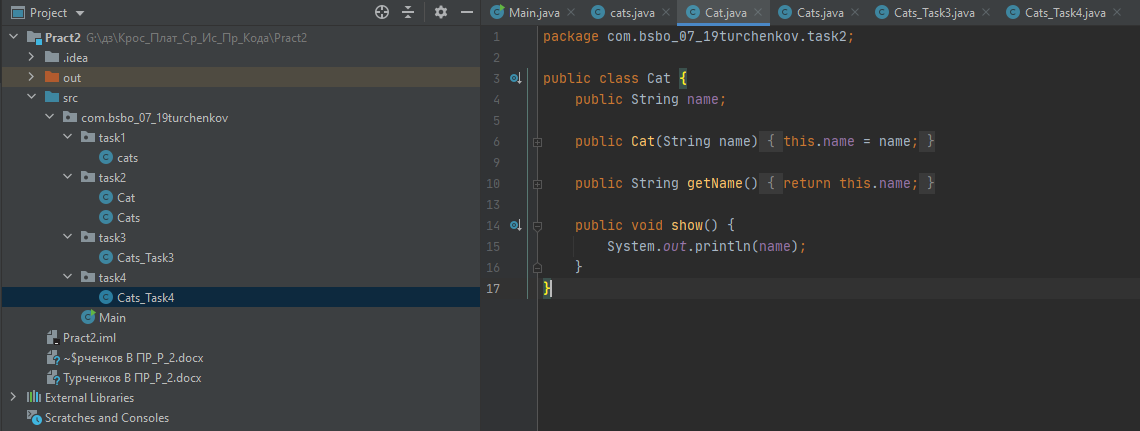


Листинг 3 Первое задание

Ниже представлен метод вызова задания 1 и создание объектов в классе “cats”. Затем происходит вызов этих наследников и полученные объекты.

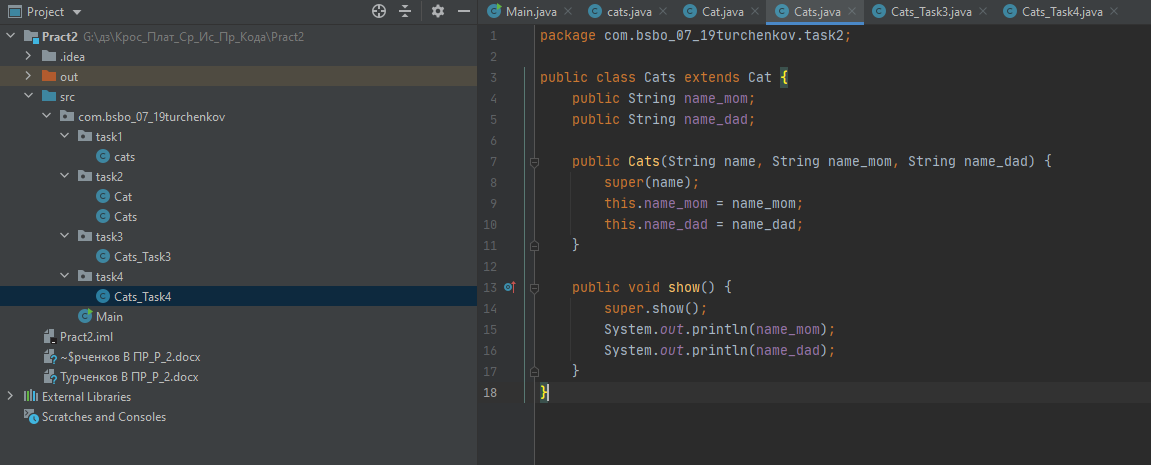
Листинг 4 Вызов задания 1

Во втором задании создается основной (родительский) класс Cat и метод show и getName, который возвращает имя



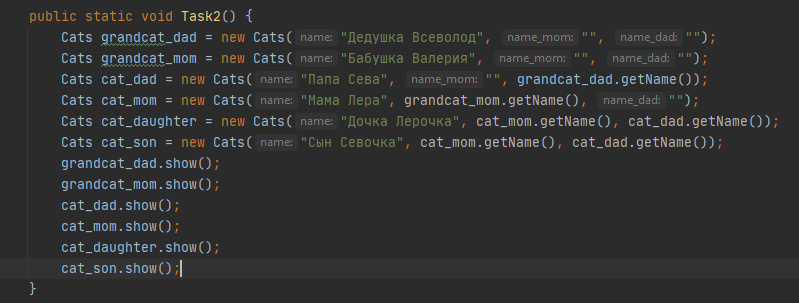
Листинг 5 Задание 2 Основной класс Cat

Далее создается подкласс Cats, который наследует основной класс Cat. В конструкторе указываются параметры, с помощью которых будут созданы коты дедушка (папин папа), бабушка (мамина мама), папа, мама, сын, дочь. Новый метод show, позволяет выводить информацию о котах вместе с его родителями.



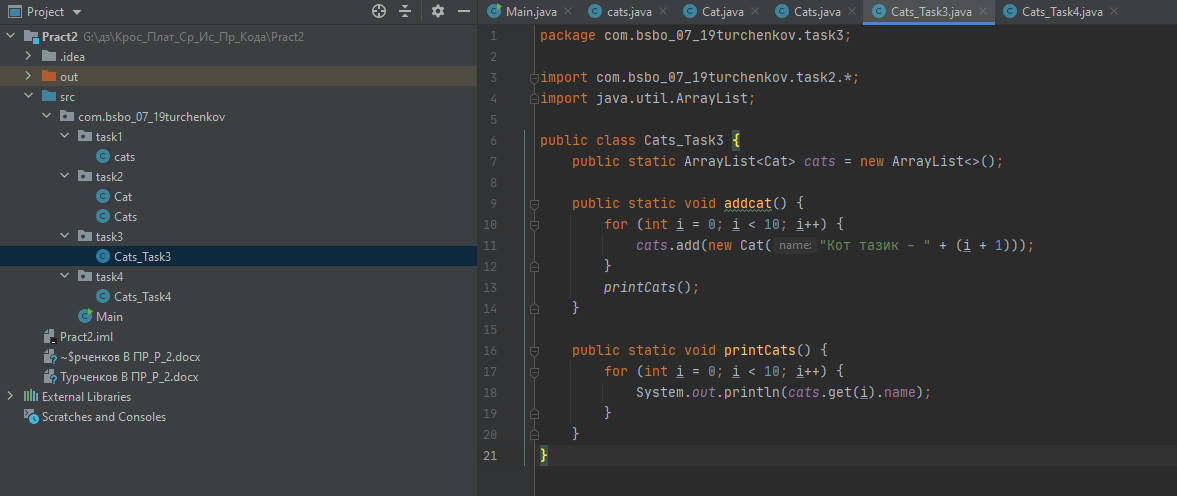
Листинг 6 Задание 2 Наследник Cat

После в main идет вызов задания 2. Создаются объекты класса Cats, заполняются имена и данные об их родителях. Согласно условию, у дедушки и у бабушки нет родителей. У папы есть отец (дедушка), но нет матери (бабушки). У мамы есть мать (бабушка), но нет отца (дедушки). У сына и дочери есть отец и есть мать.



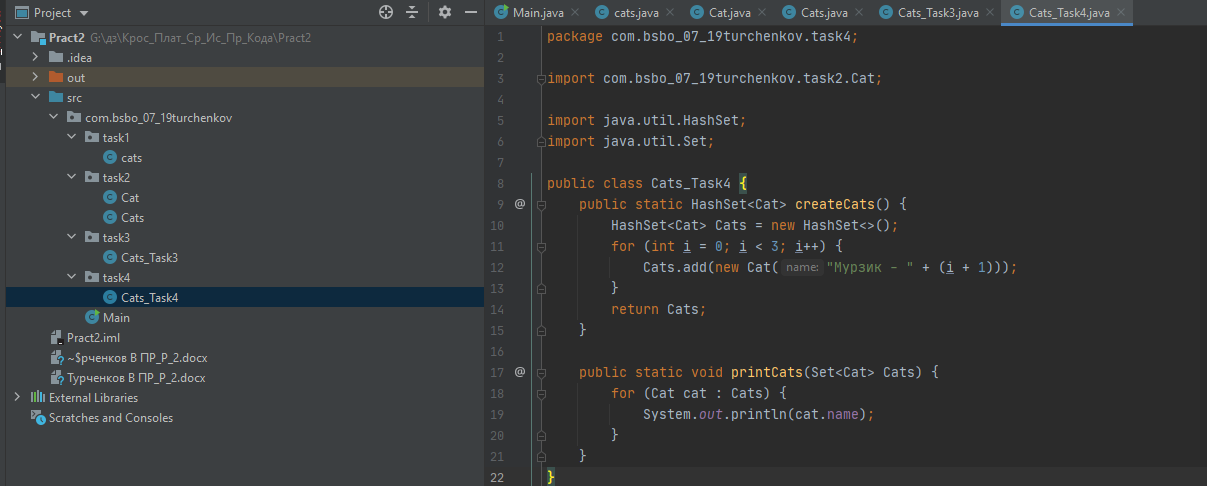
Листинг 7 Вызов задания 2

В задании 3 нужно добавить 10 одинаковых объектов. Реализовано это с помощью ArrayList и циклом for, который каждую итерацию добавляет и выводит на экран объекты с помощью метода printCats ниже.



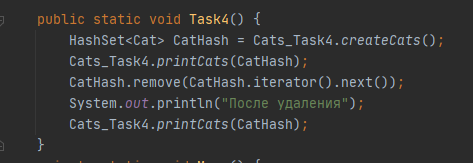
Листинг 8 Задание 3

В задании 4 нужно реализовать метод createCats. Он создает Set из котов и с помощью цикла добавляет три объекта.



Листинг 9 Задание 4

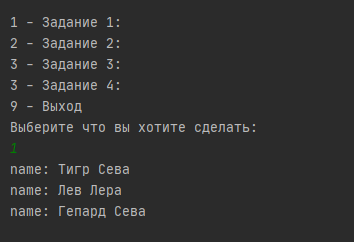
Затем в main создается экземпляр HashSet, выводим на экран с помощью метода printCats. И метод remove удаляет текущий элемент, который был получен последним вызовом next. После вся информация выводится на экран.



Листинг 10 Вызов задания 4

**Скриншоты выполнения программы**

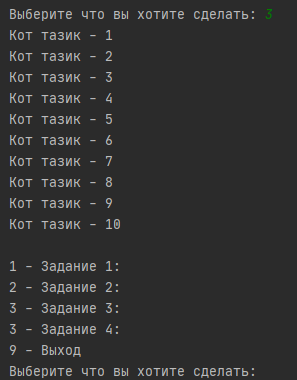
Далее приведены скриншоты работы программы:



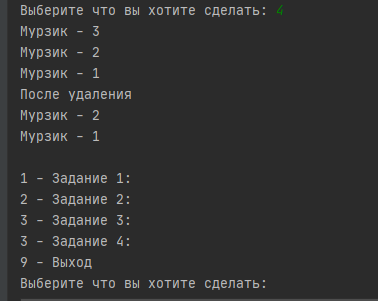
Листинг 11 Результат задания 1



Листинг 12 Результат задания 2



Листинг 13 Результат задания 3



Листинг 14 Результат задания 4

**Приложение А**

package com.bsbo\_07\_19turchenkov;  
  
import com.bsbo\_07\_19turchenkov.task1.\*;  
import com.bsbo\_07\_19turchenkov.task2.\*;  
import com.bsbo\_07\_19turchenkov.task3.\*;  
import com.bsbo\_07\_19turchenkov.task4.\*;  
  
import java.util.HashSet;  
import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 boolean tasks = true;  
 while (tasks) {  
  
 *Menu*();  
 Scanner in = new Scanner(System.*in*);  
 int choice = in.nextInt();  
 switch (choice) {  
 case 1 -> *Task1*();  
 case 2 -> *Task2*();  
 case 3 -> *Task3*();  
 case 4 -> *Task4*();  
 case 9 -> tasks = false;  
 }  
 }  
 }  
  
 public static void Task1() {  
 cats tiger = new cats.Tigers("Тигр Сева");  
 cats lion = new cats.Lions("Лев Лера");  
 cats gepard = new cats.Gepards("Гепард Сева");  
 tiger.show();  
 lion.show();  
 gepard.show();  
 }  
  
 public static void Task2() {  
 Cats grandcat\_dad = new Cats("Дедушка Всеволод", "", "");  
 Cats grandcat\_mom = new Cats("Бабушка Валерия", "", "");  
 Cats cat\_dad = new Cats("Папа Сева", "", grandcat\_dad.getName());  
 Cats cat\_mom = new Cats("Мама Лера", grandcat\_mom.getName(), "");  
 Cats cat\_daughter = new Cats("Дочка Лерочка", cat\_mom.getName(), cat\_dad.getName());  
 Cats cat\_son = new Cats("Сын Севочка", cat\_mom.getName(), cat\_dad.getName());  
 grandcat\_dad.show();  
 grandcat\_mom.show();  
 cat\_dad.show();  
 cat\_mom.show();  
 cat\_daughter.show();  
 cat\_son.show();  
 }  
  
 public static void Task3() {  
 Cats\_Task3.*addcat*();  
 }  
  
  
 public static void Task4() {  
 HashSet<Cat> CatHash = Cats\_Task4.*createCats*();  
 Cats\_Task4.*printCats*(CatHash);  
 CatHash.remove(CatHash.iterator().next());  
 System.*out*.println("После удаления");  
 Cats\_Task4.*printCats*(CatHash);  
 }  
 private static void Menu() {  
 System.*out*.print("\n1 - Задание 1: ");  
 System.*out*.print("\n2 - Задание 2:");  
 System.*out*.print("\n3 - Задание 3:");  
 System.*out*.print("\n3 - Задание 4:");  
 System.*out*.print("\n9 - Выход\n");  
 System.*out*.print("Выберите что вы хотите сделать: ");  
  
 }  
}

package com.bsbo\_07\_19turchenkov.task1;  
  
public class cats {  
 public String name;  
  
 public void show() {  
 this.name = name;  
 System.*out*.println("name: " + name);  
 }  
  
 public static class Tigers extends cats {  
 public Tigers(String name) {  
 this.name = name;  
 }  
 }  
  
 public static class Gepards extends cats {  
 public Gepards(String name) {  
 this.name = name;  
 }  
 }  
  
 public static class Lions extends cats {  
 public Lions(String name) {  
 this.name = name;  
 }  
 }  
}

package com.bsbo\_07\_19turchenkov.task2;  
  
public class Cat {  
 public String name;  
  
 public Cat(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return this.name;  
 }  
  
 public void show() {  
 System.*out*.println(name);  
 }  
}

package com.bsbo\_07\_19turchenkov.task2;  
  
public class Cats extends Cat {  
 public String name\_mom;  
 public String name\_dad;  
  
 public Cats(String name, String name\_mom, String name\_dad) {  
 super(name);  
 this.name\_mom = name\_mom;  
 this.name\_dad = name\_dad;  
 }  
  
 public void show() {  
 super.show();  
 System.*out*.println(name\_mom);  
 System.*out*.println(name\_dad);  
 }  
}

package com.bsbo\_07\_19turchenkov.task3;  
  
import com.bsbo\_07\_19turchenkov.task2.\*;  
import java.util.ArrayList;  
  
public class Cats\_Task3 {  
 public static ArrayList<Cat> *cats* = new ArrayList<>();  
  
 public static void addcat() {  
 for (int i = 0; i < 10; i++) {  
 *cats*.add(new Cat("Кот тазик - " + (i + 1)));  
 }  
 *printCats*();  
 }  
  
 public static void printCats() {  
 for (int i = 0; i < 10; i++) {  
 System.*out*.println(*cats*.get(i).name);  
 }  
 }  
}

package com.bsbo\_07\_19turchenkov.task4;  
  
import com.bsbo\_07\_19turchenkov.task2.Cat;  
  
import java.util.HashSet;  
import java.util.Set;  
  
public class Cats\_Task4 {  
 public static HashSet<Cat> createCats() {  
 HashSet<Cat> Cats = new HashSet<>();  
 for (int i = 0; i < 3; i++) {  
 Cats.add(new Cat("Мурзик - " + (i + 1)));  
 }  
 return Cats;  
 }  
  
 public static void printCats(Set<Cat> Cats) {  
 for (Cat cat : Cats) {  
 System.*out*.println(cat.name);  
 }  
 }  
}