Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет непрерывного и дистанционного обучения

Кафедра экономической информатики

Отчет

По лабораторной работе № 2

По курсу «Основы объектно-ориентированного программирования»

|  |  |  |
| --- | --- | --- |
| Выполнил |  | Студент  Шухов Владимир Григорьевич  Зачетная книжка № 6939018 |
|  |  |  |
| Проверил |  | Унучек Татьяна Михайловна |
|  |  |  |

**Минск-2018**

# Цель

Изучить особенности наследования и скрытия данных, создания абстрактных классов и интерфейсов в Java.

# Задание

Создать интерфейс Animal (животное), содержащий методы: расчет количества корма (принимаемые параметры вес и возраст животного), вывод информации о животном. Создать два класса Dog (собака) и Cat (кошка), реализующих этот интерфейс. Создать группу объектов классов Dog и Cat. Создать третий класс SearchInfo, позволяющий выводить нформацию о животных, употребивших наибольшее количество корма, значение которого введено с клавиатуры.

# Листинг кода

package IPR\_2;  
  
*/\*\*  
 \* Created by shukhovvg on 26.12.2017.  
 \*/*public interface Animal {  
  
 void countFood(int weight, int age);  
  
 void printAnimalInfo();  
  
 int getFood();  
}

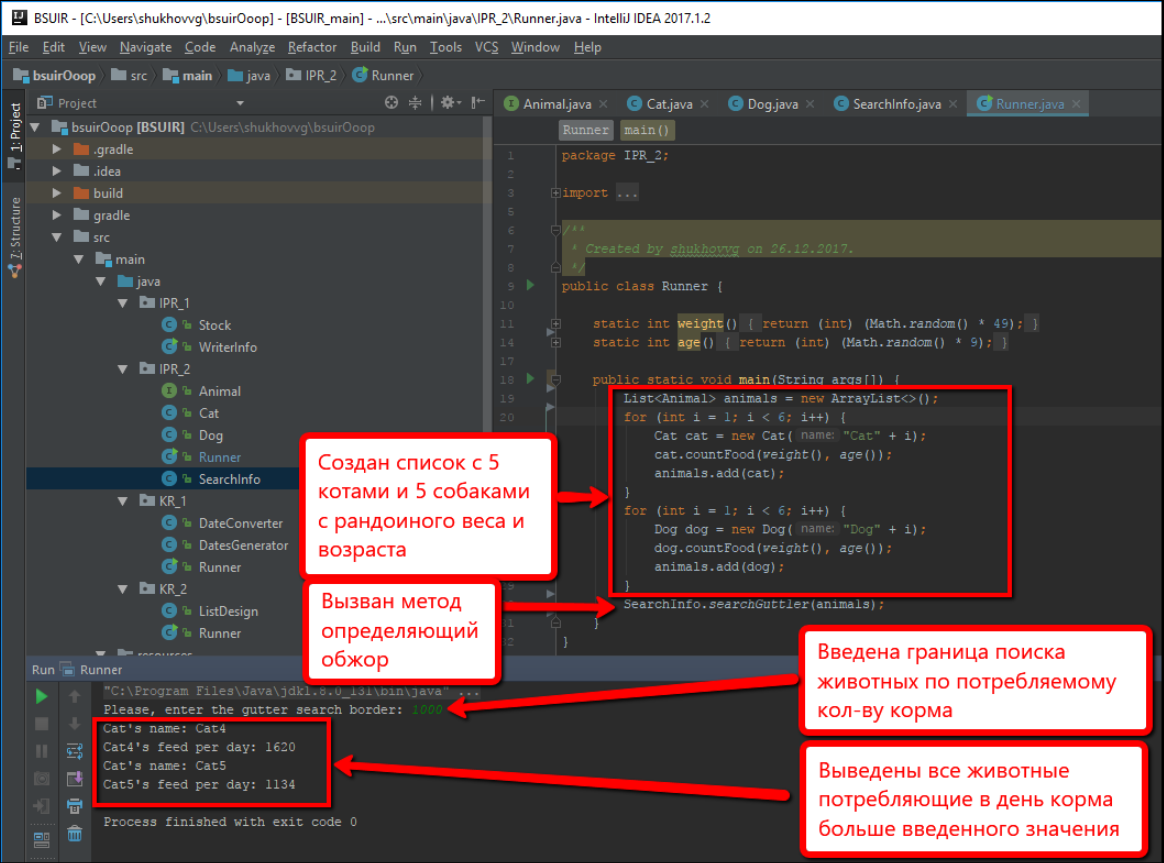
package IPR\_2;  
  
*/\*\*  
 \* Created by shukhovvg on 26.12.2017.  
 \*/*public class Cat implements Animal {  
  
 private String name;  
 private int food;  
  
 public Cat(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void countFood(int weight, int age) {  
  
 if (age <= 1) {  
 food = weight \* 10;  
 } else {  
 food = weight \* 54;  
 }  
 }  
  
 @Override  
 public void printAnimalInfo() {  
 System.*out*.println("Cat's name: " + name);  
 System.*out*.println(name + "'s feed per day: " + food);  
 }  
  
 @Override  
 public int getFood() {  
 return food;  
 }  
}

package IPR\_2;  
  
*/\*\*  
 \* Created by shukhovvg on 26.12.2017.  
 \*/*public class Dog implements Animal {  
  
 private String name;  
 private int food;  
  
 public Dog(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void countFood(int weight, int age) {  
  
 if (age <= 1) {  
 food = weight \* 15;  
 } else {  
 food = weight \* 10;  
 }  
 }  
  
 @Override  
 public void printAnimalInfo() {  
 System.*out*.println("Dog's name: " + name);  
 System.*out*.println(name + "'s feed per day: " + food);  
 }  
  
 @Override  
 public int getFood() {  
 return food;  
 }  
}

package IPR\_2;  
  
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.util.List;  
  
*/\*\*  
 \* Created by shukhovvg on 26.12.2017.  
 \*/*public class SearchInfo {  
  
 public static void searchGuttler(List<Animal> animals) {  
  
 int border = 0;  
  
 try {  
 System.*out*.print("Please, enter the gutter search border: ");  
 BufferedReader reader = new BufferedReader(new InputStreamReader(System.*in*));  
 String temp = reader.readLine();  
 border = Integer.*parseInt*(temp);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 for (int i = 0; i < animals.size(); i++) {  
 if (animals.get(i).getFood() >= border) {  
 animals.get(i).printAnimalInfo();  
 }  
 }  
 }  
}

import java.util.ArrayList;  
import java.util.List;  
  
*/\*\*  
 \* Created by shukhovvg on 26.12.2017.  
 \*/*public class Runner {  
  
 static int weight() {  
 return (int) (Math.*random*() \* 49);  
 }  
  
 static int age() {  
 return (int) (Math.*random*() \* 9);  
 }  
  
 public static void main(String args[]) {  
  
 List<Animal> animals = new ArrayList<>();  
  
 for (int i = 1; i < 6; i++) {  
 Cat cat = new Cat("Cat" + i);  
 cat.countFood(*weight*(), *age*());  
 animals.add(cat);  
 }  
  
 for (int i = 1; i < 6; i++) {  
 Dog dog = new Dog("Dog" + i);  
 dog.countFood(*weight*(), *age*());  
 animals.add(dog);  
 }  
  
 SearchInfo.*searchGuttler*(animals);  
  
 }  
}

# Интерфейс работы программы



# Выводы

Освоены особенности наследования и скрытия данных, создания абстрактных классов и интерфейсов в Java.