package com.javarush.test.level12.lesson02.task01;  
  
/\* Я не корова, Я - кит.  
Переопределить метод getName в классе Whale(Кит), чтобы программа выдавала:  
Я не корова, Я - кит.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Cow cow = new Whale();  
 System.*out*.println(cow.getName());  
 }  
 public static class Cow  
 {  
 public String getName()  
 {  
 return "Я - корова";  
 }  
 }  
 public static class Whale extends Cow  
 {  
   
 }  
}

package com.javarush.test.level12.lesson02.task02;  
  
/\* Ничего не выводится на экран  
Переопределить метод getName в классе Whale(Кит), чтобы программа ничего не выдавала на экран.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Cow cow = new Whale();  
  
 System.*out*.println(cow.getName());  
 }  
  
 public static class Cow  
 {  
 public String getName()  
 {  
 return "Я - корова";  
 }  
 }  
  
 public static class Whale extends Cow  
 {  
 }  
}

package com.javarush.test.level12.lesson02.task03;  
  
/\* Кот от кота, а собака от собаки  
Переопределить метод getChild в классах Cat(кот) и Dog(собака),  
чтобы кот порождал кота, а собака – собаку.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Pet pet1 = new Cat();  
 Pet cat = pet1.getChild();  
  
 Pet pet2 = new Dog();  
 Pet dog = pet2.getChild();  
 }  
  
 public static class Pet  
 {  
 public Pet getChild()  
 {  
 return new Pet();  
 }  
 }  
  
 public static class Cat extends Pet  
 {  
 }  
  
 public static class Dog extends Pet  
 {  
 }  
}

package com.javarush.test.level12.lesson02.task04;  
  
/\* Или «Кошка», или «Собака», или «Птица», или «Лампа»  
Написать метод, который определяет, объект какого класса ему передали, и выводит на экран одну из надписей: Кошка, Собака, Птица, Лампа.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 *printObjectType*(new Cat());  
 *printObjectType*(new Bird());  
 *printObjectType*(new Lamp());  
 *printObjectType*(new Cat());  
 *printObjectType*(new Dog());  
 }  
  
 public static void printObjectType(Object o)  
 {  
 //Напишите тут ваше решение  
 }  
  
 public static class Cat  
 {  
 }  
  
 public static class Dog  
 {  
 }  
  
 public static class Bird  
 {  
 }  
  
 public static class Lamp  
 {  
 }  
}

package com.javarush.test.level12.lesson02.task05;  
  
/\* Или «Корова», или «Кит», или «Собака», или «Неизвестное животное»  
Написать метод, который определяет, объект какого класса ему передали, и возвращает результат – одно значение из: «Корова», «Кит», «Собака», «Неизвестное животное».  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 System.*out*.println(*getObjectType*(new Cow()));  
 System.*out*.println(*getObjectType*(new Dog()));  
 System.*out*.println(*getObjectType*(new Whale()));  
 System.*out*.println(*getObjectType*(new Pig()));  
 }  
  
 public static String getObjectType(Object o)  
 {  
 //Напишите тут ваше решение  
  
 return "Неизвестное животное";  
 }  
  
 public static class Cow  
 {  
 }  
  
 public static class Dog  
 {  
 }  
  
 public static class Whale  
 {  
 }  
  
 public static class Pig  
 {  
 }  
}

package com.javarush.test.level12.lesson04.task01;  
  
/\* print(int) и print(String)  
Написать два метода: print(int) и print(String).  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 //Напишите тут ваши методы  
}

package com.javarush.test.level12.lesson04.task02;  
  
/\* print(int) и print(Integer)  
Написать два метода: print(int) и print(Integer).  
Написать такой код в методе main, чтобы вызвались они оба.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 //Напишите тут ваши методы  
}

package com.javarush.test.level12.lesson04.task03;  
  
/\* Пять методов print с разными параметрами  
Написать пять методов print с разными параметрами.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 //Напишите тут ваши методы  
}

package com.javarush.test.level12.lesson04.task04;  
  
/\* Три метода возвращают минимальное из двух переданных в него чисел  
Написать public static методы: int min(int, int), long min(long, long), double min(double, double).  
Каждый метод должен возвращать минимальное из двух переданных в него чисел.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 //Напишите тут ваши методы  
}

package com.javarush.test.level12.lesson04.task05;  
  
/\* Три метода возвращают максимальное из двух переданных в него чисел  
Написать public static методы: int max(int, int), long max (long, long), double max (double, double).  
Каждый метод должен возвращать максимальное из двух переданных в него чисел.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 //Напишите тут ваши методы  
}

package com.javarush.test.level12.lesson06.task01;  
  
/\* Абстрактный класс Pet  
Сделать класс Pet абстрактным.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public static class Pet  
 {  
 public String getName()  
 {  
 return "Я - котенок";  
 }  
 }  
  
}

package com.javarush.test.level12.lesson06.task02;  
  
/\* Исправь код. Первая задача  
Исправь код, чтобы программа компилировалась.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
  
 public static class Pet  
 {  
 public String getName()  
 {  
 return "Я - котенок";  
 }  
  
 public abstract Pet getChild();  
 }  
  
}

package com.javarush.test.level12.lesson06.task03;  
  
/\* Исправь код. Вторая задача  
Исправь код, чтобы программа компилировалась.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public static class Pet  
 {  
 public String getName()  
 {  
 return "Я - котенок";  
 }  
  
 public Pet getChild();  
 }  
  
}

package com.javarush.test.level12.lesson06.task04;  
  
/\* Класс Cow от Animal  
Унаследуй класс Cow от Animal.  
Реализуй все недостающие методы в классе Cow.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public static abstract class Animal  
 {  
 public abstract String getName();  
 }  
  
 public static class Cow  
 {  
 }  
  
}

package com.javarush.test.level12.lesson06.task05;  
  
/\* Классы Cat и Dog от Pet  
Унаследуй классы Cat и Dog от Pet.  
Реализуй недостающие методы. Классы Cat и Dog не должны быть абстрактными.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public static abstract class Pet  
 {  
 public abstract String getName();  
 public abstract Pet getChild();  
 }  
  
 public static class Cat   
 {  
 }  
  
 public static class Dog   
 {  
   
 }  
  
}

package com.javarush.test.level12.lesson09.task01;  
  
/\* Интерфейс Fly  
Напиши свой public интерфейс Fly(летать). Добавь в него два метода.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
  
}

package com.javarush.test.level12.lesson09.task02;  
  
/\* Интерфейсы Fly, Run, Swim  
Напиши public интерфейсы Fly(летать), Run(бежать/ездить), Swim(плавать).  
Добавить в каждый интерфейс по одному методу.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
//add interfaces here - добавь интерфейсы тут  
  
}

package com.javarush.test.level12.lesson09.task03;  
  
/\* Fly, Move, Eat для классов Dog, Car, Duck, Airplane  
Есть public интерфейсы Fly(летать), Move(передвигаться), Eat(есть).  
Добавь эти интерфейсы классам Dog(собака), Car(автомобиль), Duck(утка), Airplane(самолет).  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public interface Fly  
 {  
 public void fly();  
 }  
  
 public interface Move  
 {  
 public void move();  
 }  
  
 public interface Eat  
 {  
 public void eat();  
 }  
  
 public class Dog  
 {  
 }  
  
 public class Duck  
 {  
 }  
  
 public class Car  
 {  
 }  
  
 public class Airplane  
 {  
 }  
}

package com.javarush.test.level12.lesson09.task04;  
  
/\* Fly, Run, Swim для классов Human, Duck, Penguin, Airplane  
Есть public интерфейсы Fly(летать), Run(бежать/ездить), Swim(плавать).  
Добавь эти интерфейсы классам Human(человек), Duck(утка), Penguin(пингвин), Airplane(самолет).  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public interface Fly  
 {  
 public void fly();  
 }  
  
 public interface Run  
 {  
 public void run();  
 }  
  
 public interface Swim  
 {  
 public void swim();  
 }  
  
  
 public class Human  
 {  
  
 }  
  
 public class Duck  
 {  
  
 }  
  
 public class Penguin  
 {  
  
 }  
  
 public class Airplane  
 {  
  
 }  
}

package com.javarush.test.level12.lesson09.task05;  
  
/\* Класс Human и интерфейсы Run, Swim  
Напиши public класс Human(человек) и public интерфейсы Run(бежать/ездить), Swim(плавать).  
Добавь в каждый интерфейс по одному методу.  
Добавь эти интерфейсы классу Human, но не реализуй методы.  
Объяви класс Human абстрактным.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 //add public interfaces and public class here - добавь public интерфейсы и public класс тут  
}

package com.javarush.test.level12.lesson12.home01;  
  
/\* Метод getName в классе Cat  
Переопредели метод getName в классе Cat так, чтобы программа выдавала на экран надпись  
«Я - кот».  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Pet pet = new Cat();  
  
 System.*out*.println(pet.getName());  
 }  
  
 public static class Pet  
 {  
 public String getName()  
 {  
 return "Я - пушистик";  
 }  
 }  
  
 public static class Cat extends Pet  
 {  
  
 }  
}

package com.javarush.test.level12.lesson12.home02;  
  
/\* Метод setName в классе Cat  
Переопредели метод setName в классе Cat так, чтобы программа выдавала на экран надпись  
«Я - кот».  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Pet pet = new Cat();  
 pet.setName("Я - пушистик");  
  
 System.*out*.println(pet.getName());  
 }  
  
 public static class Pet  
 {  
 protected String name;  
  
 public Pet()  
 {  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 }  
  
 public static class Cat extends Pet  
 {  
  
 }  
}

package com.javarush.test.level12.lesson12.home03;  
  
/\* Добавь один метод в класс Cat  
Добавь один метод в класс Cat так, чтобы программа ничего не выводила на экран.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Pet pet = new Cat();  
 pet.setName("Я - пушистик");  
  
 System.*out*.println(pet.getName());  
 }  
  
 public static class Pet  
 {  
 protected String name;  
  
 public Pet()  
 {  
 }  
  
 public final String getName()  
 {  
 return name;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 }  
  
 public static class Cat extends Pet  
 {  
  
 }  
}

package com.javarush.test.level12.lesson12.home04;  
  
/\* Что это? «Кот», «Тигр», «Лев», «Бык», «хз»  
Напиши метод, который определяет, какой объект передали в него.  
Программа должна выводить на экран одну из надписей:  
«Кот», «Тигр», «Лев», «Бык», «хз».  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 System.*out*.println(*getObjectType*(new Cat()));  
 System.*out*.println(*getObjectType*(new Tiger()));  
 System.*out*.println(*getObjectType*(new Lion()));  
 System.*out*.println(*getObjectType*(new Bull()));  
 System.*out*.println(*getObjectType*(new Pig()));  
 }  
  
 public static String getObjectType(Object o)  
 {  
 //напишите тут ваш код  
  
 return "хз";  
 }  
  
 public static class Cat  
 {  
 }  
  
 public static class Tiger  
 {  
 }  
  
 public static class Lion  
 {  
 }  
  
 public static class Bull  
 {  
 }  
  
 public static class Pig  
 {  
 }  
}

package com.javarush.test.level12.lesson12.home05;  
  
/\* Что это? «Кот», «Тигр», «Лев», «Бык», «Корова», «Животное»  
Напиши метод, который определяет, какой объект передали в него.  
Программа должна выводить на экран одну из надписей:  
«Кот», «Тигр», «Лев», «Бык», «Корова», «Животное».  
Замечание: постарайся определять тип животного как можно более точно.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 System.*out*.println(getObjectType(new Cat()));  
 System.*out*.println(getObjectType(new Tiger()));  
 System.*out*.println(getObjectType(new Lion()));  
 System.*out*.println(getObjectType(new Bull()));  
 System.*out*.println(getObjectType(new Cow()));  
 System.*out*.println(getObjectType(new Animal()));  
 }  
  
 public static String getObjectType(Object o)  
 {  
 //напишите тут ваш код  
  
 return "Животное";  
 }  
  
 public static class Cat extends Animal //<--Классы наследуются!!  
 {  
 }  
  
 public static class Tiger extends Cat  
 {  
 }  
  
 public static class Lion extends Cat  
 {  
 }  
  
 public static class Bull extends Animal  
 {  
 }  
  
 public static class Cow extends Animal  
 {  
 }  
  
 public static class Animal  
 {  
 }  
}

package com.javarush.test.level12.lesson12.home06;  
  
/\* Fly, Run, Climb для классов Cat, Dog, Tiger, Duck  
1. Внутри класса Solution создай интерфейс public interface Fly(летать) с методом void fly().  
2. Внутри класса Solution создай интерфейс public interface Climb(лазить по деревьям) с методом void climb().  
3. Внутри класса Solution создай интерфейс public interface Run(бегать) с методом void run().  
4. Подумай логически, какие именно интерфейсы нужно добавить для каждого класса.  
5. Добавь интерфейсы классам Cat(кот), Dog(собака), Tiger(тигр), Duck(Утка).  
\*/  
  
public class Solution {  
  
 public class Cat {  
 }  
  
 public class Dog {  
 }  
  
 public class Tiger extends Cat {  
 }  
  
 public class Duck {  
 }  
}

package com.javarush.test.level12.lesson12.home07;  
  
/\* Fly, Run, Swim для классов Duck, Penguin, Toad  
Есть интерфейсы Fly(летать), Swim(плавать), Run(бегать).  
Добавь эти интерфейсы классам Duck(утка), Penguin(пингвин), Toad(жаба)  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
  
 }  
  
 public interface Fly  
 {  
 public void fly();  
 }  
  
 public interface Run  
 {  
 public void run();  
 }  
  
 public interface Swim  
 {  
 public void swim();  
 }  
  
 public class Duck  
 {  
  
 }  
  
 public class Penguin  
 {  
  
 }  
  
 public class Toad  
 {  
  
 }  
}

package com.javarush.test.level12.lesson12.home08;  
  
/\* Интерфейсы к классу Human  
Добавь как можно больше интерфейсов к классу Human, но чтобы он не стал абстрактным классом.  
Добавлять методы в класс Human запрещается.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Human human = new Human();  
 System.*out*.println(human);  
 }  
  
 public static interface Worker  
 {  
 public void workLazy();  
 }  
  
 public static interface Businessman  
 {  
 public void workHard();  
 }  
  
 public static interface Secretary  
 {  
 public void workLazy();  
 }  
  
 public static interface Miner  
 {  
 public void workVeryHard();  
 }  
  
 public static class Human  
 {  
  
 public void workHard()  
 {  
 }  
  
 public void workLazy()  
 {  
 }  
 }  
}

package com.javarush.test.level12.lesson12.home09;  
  
/\* Родитель класса CTO  
Добавь такой класс-родитель к классу CTO(технический директор), чтобы класс перестал быть абстрактным.  
Добавлять/реализовывать методы в классе CTO запрещается.  
\*/  
  
public class Solution  
{  
  
 public static void main(String[] args)  
 {  
 CTO cto = new CTO();  
 System.*out*.println(cto);  
 }  
  
 public static interface Businessman  
 {  
 public void workHard();  
 }  
  
 public static class CTO implements Businessman  
 {  
  
 }  
  
  
}

package com.javarush.test.level12.lesson12.home10;  
  
/\* Метод, который выводит на экран число 10  
Добавь еще один метод, чтобы программа выводила на экран число 10.  
Подсказка: используй перегрузку методов.  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Integer i = 5;  
 int x = *transformValue*(i);  
  
 System.*out*.println(x);  
 }  
  
 public static int transformValue(int i)  
 {  
 return i\*i;  
 }  
}

package com.javarush.test.level12.lesson12.bonus01;  
  
/\* Нужно исправить программу, чтобы компилировалась и работала  
Расставить правильно ключевые слова abstract, чтобы программа компилировалась (там где надо и не надо).  
\*/  
  
public class Solution  
{  
  
 public static void main(String[] args)  
 {  
 Horse horse = new Pegas();  
 horse.run();  
 }  
  
 public static interface Fly  
 {  
 public abstract void fly();  
 }  
  
 public static abstract class Horse  
 {  
 public void run()  
 {  
  
 }  
 }  
  
 public static class Pegas extends Horse implements Fly  
 {  
 public abstract void fly()  
 {  
  
 }  
 }  
  
 public static class SwimPegas extends Pegas  
 {  
 public void swim();  
 }  
  
}

package com.javarush.test.level12.lesson12.bonus02;  
  
/\* Нужно добавить в программу новую функциональность  
Сделать класс Pegas(пегас) на основе класса Horse(лошадь) и интерфейса Fly(летать).  
\*/  
  
public class Solution  
{  
 public static void main(String[] args)  
 {  
 Pegas horse = new Pegas();  
 }  
  
 public static interface Fly  
 {  
 public void fly();  
 }  
  
 public static class Horse  
 {  
 public void run()  
 {  
  
 }  
 }  
  
 public static class Pegas   
 {  
   
 }  
}

package com.javarush.test.level12.lesson12.bonus03;  
  
/\* Задача по алгоритмам  
Написать метод, который возвращает минимальное число в массиве и его позицию (индекс).  
\*/  
  
public class Solution  
{  
 public static void main(String[] args) throws Exception  
 {  
 int[] data = new int[]{1, 2, 3, 5, -2, -8, 0, 77, 5, 5};  
  
 Pair<Integer, Integer> result = *getMinimumAndIndex*(data);  
  
 System.*out*.println("Minimum is " + result.x);  
 System.*out*.println("Index of minimum element is " + result.y);  
 }  
  
 public static Pair<Integer, Integer> getMinimumAndIndex(int[] array)  
 {  
 if (array == null || array.length == 0)  
 {  
 return new Pair<Integer, Integer>(null, null);  
 }  
 return new Pair<Integer, Integer>(0, 0);  
 }  
  
  
 public static class Pair<X, Y>  
 {  
 public X x;  
 public Y y;  
  
 public Pair(X x, Y y)  
 {  
 this.x = x;  
 this.y = y;  
 }  
 }  
}