Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Кафедра безпеки інформаційних систем і технологій

Лабораторна робота №3

*з навчальної дисципліни*

«Спеціалізовані мови програмування та проектування електронних елементів і систем»

Виконав:

Студент групи КБ-41

Кравченко Є.М.

Перевірив:

Олійников Р.В.

Харків – 2020 р.

1. При помощи статического метода реализовать пузырьковую сортировку массива целых чисел (длина массива задается при вызове метода сортировки).

**static void** sort(**int** n) {  
 **int**[] arr = **new int**[n];  
 **for** (**int** i = 0; i < n; i++) {  
 arr[i] = (**int**) (Math.*random*() \* 100);  
 }  
 System.***out***.println(Arrays.*toString*(arr));  
 **for** (**int** i = 0; i < arr.**length**; i++) {  
 **for** (**int** j = 0; j < arr.**length**; j++) {  
 **if** (arr[i] < arr[j]) {  
 **int** tmp = arr[i];  
 arr[i] = arr[j];  
 arr[j] = tmp;  
 }  
 }  
 }  
 System.***out***.println(Arrays.*toString*(arr));  
}

1. Задана “неровная матрица” (или ступенчатый массив - двумерный массив, строки в котором имеют различную длину: int [][] arr; arr = new int[ 5 ][]; arr[ 0 ] = new int[ 3 ]; arr[ 1 ] = new int[ 7 ]; и т.д.). Матрица содержит целые значения. Отсортировать каждую строку (использовать Arrays.sort() ), распечатать матрицу на экране. Повторно распечатать матрицу, но выводить на экран только отрицательные значения (использовать цикл с break).

**int**[][] arr = **new int**[10][];  
**for** (**int** i = 0; i < arr.**length**; i++) {  
 arr[i] = **new int**[(**int**) (Math.*random*() \* 10 + 1)];  
}  
System.***out***.println(**"Array:"**);  
**for** (**int** i = 0; i < arr.**length**; i++) {  
 **for** (**int** j = 0; j < arr[i].**length**; j++) {  
 arr[i][j] = (**int**) (Math.*random*() \* 200 - 100);  
 System.***out***.print(arr[i][j] + **" "**);  
 }  
 System.***out***.println();  
}  
**for** (**int**[] ints : arr) {  
 Arrays.*sort*(ints);  
}  
System.***out***.println(**"Sorted array:"**);  
**for** (**int**[] ints : arr) {  
 **for** (**int** anInt : ints) {  
 System.***out***.print(anInt + **" "**);  
 }  
 System.***out***.println();  
}  
System.***out***.println(**"Negative array"**);  
**for** (**int**[] ints : arr) {  
 **for** (**int** anInt : ints) {  
 **if** (anInt > 0) {  
 **continue**;  
 }  
 System.***out***.print(anInt + **" "**);  
 }  
 System.***out***.println();  
}

1. Создать собственный класс с полями доступа public и private, проверить их доступность из другого класса текущего пакета.

**public class** NewClass {  
 **private int privateField**;  
 **public int publicField**;  
}

1. Сделать класс с перегруженными конструкторами (по умолчанию, с заданными параметрами, создание копии другого объекта).

**public class** NewClass {  
 **private int privateField**;  
 **public int publicField**;  
 **public** NewClass() {  
  
 }  
  
 **public** NewClass(**int** privateField, **int** publicField) {  
 **this**.**privateField** = privateField;  
 **this**.**publicField** = publicField;  
 }  
  
 **public** NewClass(NewClass newClass) {  
 **this**.**publicField** = newClass.**publicField**;  
 **this**.**privateField** = newClass.**privateField**;  
 }  
  
}

1. Сделать абстрактный класс “автомобиль” и его наследников (“легковой”, “грузовой” и т.п.), продемонстрировать полиморфные методы (вывод названия автомобиля, перемещение за единицу времени, вывод потребленного топлива, перевезенного груза и пассажиров, загрязнение окружающей среды; предусмотреть возможности для электромобилей).

**public abstract class** Car {  
 **public abstract void** printSpeed();  
 **public abstract void** printName();  
 **public abstract void** consumedFuel();  
 **public abstract void** printPassengerCount();  
}

**public class** BigCar **extends** Car {  
 @Override  
 **public void** printSpeed() {  
 System.***out***.println(60);  
 }  
 @Override  
 **public void** printName() {  
 System.***out***.println(**"Track"**);  
 }  
 @Override  
 **public void** consumedFuel() {  
 System.***out***.println(20);  
 }  
 @Override  
 **public void** printPassengerCount() {  
 System.***out***.println(3);  
 }  
}

**public class** LiteCar **extends** Car {  
 @Override  
 **public void** printSpeed() {  
 System.***out***.println(100);  
 }  
 @Override  
 **public void** printName() {  
 System.***out***.println(**"Name"**);  
 }  
 @Override  
 **public void** consumedFuel() {  
 System.***out***.println(12);  
 }  
 @Override  
 **public void** printPassengerCount() {  
 System.***out***.println(4);  
 }  
}