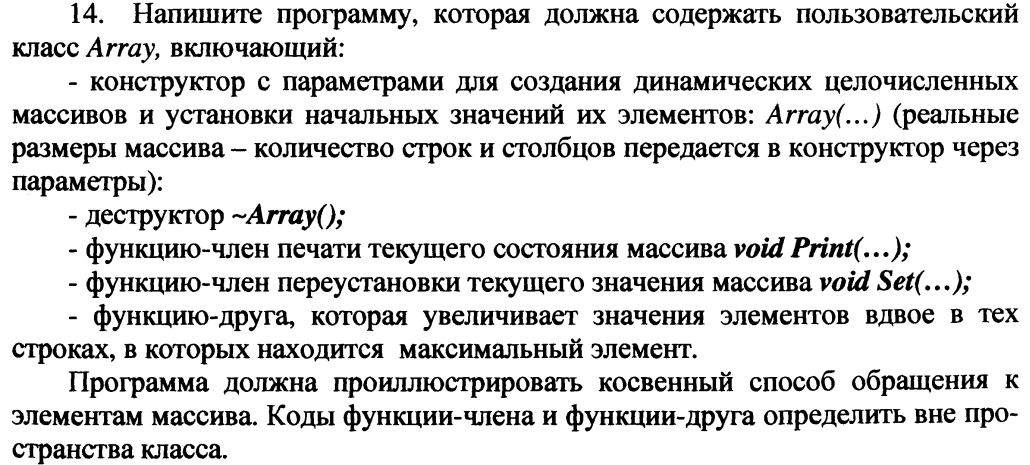
**Лабораторная работа #4**

**Ткачев Иван**

Вариант 14

Задание 4.1.

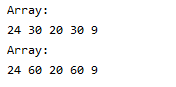


**Листинг**

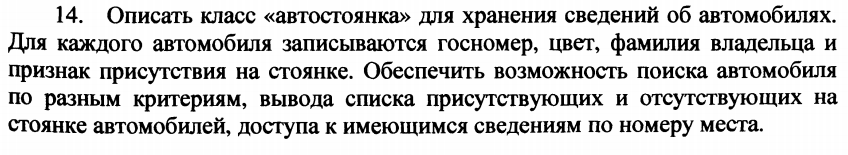
**import** java.util.Arrays;  
  
**public class** Array {  
 **private int size**;  
 **private int currPos**;  
 **private int**[] **mass**;  
  
 **public** Array() {  
 **this**.**size** = 10;  
 **mass** = **new int**[**size**];  
 **currPos** = 0;  
 }  
  
 **public** Array(**int** size){  
 **this**.**size** = size;  
 **mass** = **new int**[size];  
 **currPos** = 0;  
 }  
  
 **public** Array(**int** size, **int** defaultValue){  
 **this**.**size** = size;  
 **mass** = **new int**[size];  
 Arrays.*fill*(**mass**, defaultValue);  
 **currPos** = size-1;  
 }  
  
 **public int** getSize() {  
 **return currPos**;  
 }  
  
 **public int**[] getMass() {  
 **return mass**;  
 }  
  
 **public void** clear(){  
 **this**.**size** = 10;  
 **mass** = **new int**[**size**];  
 **currPos** = 0;  
 }  
  
 **public void** addValue(**int** value){  
 **if**(**currPos** == **size**) {  
 **size** = **size** \* 2;  
 System.***out***.println(**"UP size from "** + **currPos** + **" to "** + **size**);  
 **int**[] newMass = **new int**[**size**];  
 **for** (**int** i = 0; i < **currPos**; i++) {  
 newMass[i] = **mass**[i];  
 }  
 newMass[**currPos**] = value;  
 **mass** = newMass;  
 **currPos**++;  
 } **else** {  
 **mass**[**currPos**] = value;  
 **currPos**++;  
 }  
 }  
  
 **public void** removeValue(){  
 **if**(**currPos** != 0){  
 **mass**[**currPos**-1] = 0;  
 **currPos**--;  
 }  
 }  
  
 **public void** setValue(**int** index, **int** value){  
 **if**(index > **size** - 1){  
 System.***out***.println(**"Out of bounds!"**);  
 } **else** {  
 **mass**[index] = value;  
 }  
 }  
  
 **public int** getValue(**int** index){  
 **return mass**[index];  
 }  
  
 **public void** print(){  
 System.***out***.println(**"Array: "**);  
 **for** (**int** i = 0; i < **currPos**; i++) {  
 **if**(i % 10 == 0){  
 System.***out***.println();  
 }  
 System.***out***.print(**mass**[i] + **" "**);  
 }  
 System.***out***.println();  
 }  
}

**public static void** main(String[] args) {  
 Array mass = **new** Array();  
 **for** (**int** i = 0; i < 5; i++) {  
 mass.addValue((**int**) (Math.*random*() \* 30));  
 }  
  
 mass.setValue(1, 30);  
 mass.setValue(3, 30);  
 mass.print();  
 *updateValues*(mass);  
 mass.print();  
}  
  
**private static void** updateValues(Array mass){  
 **int** maxValue = 0;  
 Array maxValuePos = **new** Array();  
 **for** (**int** i = 0; i < mass.getSize(); i++) {  
 **if**(mass.getValue(i) > maxValue){  
 maxValue = mass.getValue(i);  
 maxValuePos.clear();  
 maxValuePos.addValue(i);  
 } **else if**(mass.getValue(i) == maxValue){  
 maxValuePos.addValue(i);  
 }  
 }  
  
 **for** (**int** i = 0; i < maxValuePos.getSize(); i++) {  
 **int** index = maxValuePos.getValue(i);  
 mass.setValue(index, mass.getValue(index) \* 2);  
 }  
}

**Результат**



Задание 4.2.

**Листинг**

**public class** Parking {  
  
 **private** List<Pair<Integer, Car>> **carsOnParking**;  
  
 **public** Parking() {  
 **carsOnParking** = **new** ArrayList<Pair<Integer, Car>>();  
 }  
 **public void** infoAboutPlace(**int** placeNumber){  
 **for** (Pair<Integer, Car> carPlace: **carsOnParking**) {  
 **if**(carPlace.getKey() == placeNumber){  
 System.***out***.print(carPlace.getValue().getLicensePlate() + **" is on this place: "** + placeNumber);  
 **return**;  
 }  
 }  
 System.***out***.print(**"There is no cars on this place: "** + placeNumber);  
 }  
  
}

**public class** Car {  
  
 **private** String **licensePlate**;  
 **private** String **color**;  
 **private** String **fullName**;  
 **private boolean isOnParking**;  
  
 **public** Car(String licensePlate, String color, String fullName, **boolean** isOnParking) {  
 **this**.**licensePlate** = licensePlate;  
 **this**.**color** = color;  
 **this**.**fullName** = fullName;  
 **this**.**isOnParking** = isOnParking;  
 }  
}  
  
**public class** Main {  
 **public static void** main(String[] args) {  
 Scanner in = **new** Scanner(System.***in***);  
 System.***out***.println(**"Input count of cars: "**);  
 **int** count = in.nextInt();  
 in.nextLine();  
  
 Parking parking = **new** Parking();  
  
 List<Car> cars = **new** ArrayList<Car>();  
 **for** (**int** i = 0; i < count; i++) {  
 System.***out***.println(**"Enter license plate number..."**);  
 String licensePlate = in.nextLine();  
 System.***out***.println(**"Enter color..."**);  
 String color = in.nextLine();  
 System.***out***.println(**"Enter full name..."**);  
 String fullName = in.nextLine();  
 System.***out***.println(**"Is car present on parking (Y/N)..."**);  
 String yesNo = in.nextLine();  
 **boolean** isOnParking = **false**;  
 **if**(!yesNo.equals(**"Y"**)){  
 isOnParking = **true**;  
 }  
 Car car = **new** Car(licensePlate, color, fullName, isOnParking);  
 cars.add(car);  
 **if**(isOnParking){  
 parking.getCarsOnParking().add(**new** Pair<>(i, car));  
 }  
 }  
  
 *showCarsOnParking*(parking);  
 *showCarsNotOnParking*(cars);  
 parking.infoAboutPlace(1);  
  
 **boolean** isEnd = **true**;  
 System.***out***.println();  
 **while**(isEnd){  
 System.***out***.println(**"Do you want to start filtering (Y/N)?"**);  
 **if**(in.nextLine().equals(**"Y"**)){  
 *findCarsByCriteria*(in, cars);  
 } **else** isEnd = **false**;  
 }  
 }  
  
 **private static** List<Car> findCarsByCriteria(Scanner in, List<Car> cars){  
 List<Pair<String, String>> criteria = **new** ArrayList<Pair<String, String>>();  
  
 System.***out***.println(**"Find cars by criteria (write 'skip' to avoid current criterion)"**);  
  
 System.***out***.println(**"License plate number..."**);  
 String licensePlate = in.nextLine();  
 **if**(!**"skip"**.equals(licensePlate)){  
 criteria.add(**new** Pair<>(**"licencePlate"**, licensePlate));  
 }  
 System.***out***.println(**"Color..."**);  
 String color = in.nextLine();  
 **if**(!**"skip"**.equals(color)){  
 criteria.add(**new** Pair<>(**"color"**, color));  
 }  
 System.***out***.println(**"Full name..."**);  
 String fullName = in.nextLine();  
 **if**(!**"skip"**.equals(fullName)){  
 criteria.add(**new** Pair<>(**"fullName"**, fullName));  
 }  
 System.***out***.println(**"Is car present on parking (Y/N)..."**);  
 String yesNo = in.nextLine();  
 **boolean** isOnParking = **false**;  
 **if**(!yesNo.equals(**"Y"**)){  
 isOnParking = **true**;  
 }  
 **if**(!**"skip"**.equals(yesNo)){  
 criteria.add(**new** Pair<>(**"isOnParking"**, String.*valueOf*(isOnParking)));  
 }  
  
 List<Car> result = **new** ArrayList<>();  
 **for** (Car car : cars) {  
 **boolean** check = **true**;  
 **for** (Pair<String, String> criterion: criteria) {  
 **switch** (criterion.getKey()){  
 **case "licencePlate"**:  
 **if**(!car.getLicensePlate().contains(criterion.getValue())){  
 check = **false**;  
 }  
 **break**;  
 **case "color"**:  
 **if**(!car.getColor().contains(criterion.getValue())){  
 check = **false**;  
 }  
 **break**;  
 **case "fullName"**:  
 **if**(!car.getFullName().contains(criterion.getValue())){  
 check = **false**;  
 }  
 **break**;  
 **case "isOnParking"**:  
 **if**(!String.*valueOf*(car.isOnParking()).contains(criterion.getValue())){  
 check = **false**;  
 }  
 **break**;  
 }  
 **if**(!check) **break**;  
 }  
 **if**(check){  
 result.add(car);  
 }  
 }

System.***out***.println(**"After filtering: "**);  
**for** (Car car: result) {  
 System.***out***.println(car.getLicensePlate() + **" "** + car.getColor() + **" "** + car.getFullName() + **" "** + car.isOnParking());  
}

**return** result;  
 }  
  
 **private static void** showCarsOnParking(Parking parking){  
 System.***out***.println(**"Cars with license plate number on parking now: "**);  
 **for** (Pair<Integer, Car> carPlace: parking.getCarsOnParking()) {  
 System.***out***.print(carPlace.getValue().getLicensePlate() + **" "**);  
 }  
 System.***out***.println();  
 }  
   
 **private static void** showCarsNotOnParking(List<Car> cars){  
 System.***out***.println(**"Cars with license plate number not on parking now: "**);  
 **for** (Car car: cars) {  
 **if**(!car.isOnParking()){  
 System.***out***.print(car.getLicensePlate() + **" "**);  
 }  
 }  
 System.***out***.println();  
 }  
}

**Результат**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| License Plate | 123fa | fa123 | f123a |
| Color | red | green | red |
| Full name | fio1 | fio2 | fio3 |
| Is in parking | Yes | No | Yes |

Requirements:

License Plate: 123

Color: red

Is in parking: Yes



|  |  |  |
| --- | --- | --- |
|  |  |  |

Вывод авто на парковке: 

Информация о парковочном месте #1: 

