Практика день 2 стр.87

Вариант 10

Условие

Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы setТип(), getТип(), toString(). Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль. В каждом классе, обладающем информацией, должно быть объявлено несколько конструкторов.

Train: Пункт назначения, Номер поезда, Время отправления, Число мест (общих, купе, плацкарт, люкс). Создать массив объектов. Вывести:

a) список поездов, следующих до заданного пункта назначения;

b) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;

c) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места

Код программы

public class Main {  
 public static void main(String[] args) {  
 List<Train> trainsList = Trains.*newRandomTrainList*(10);  
  
 System.*out*.println("Original list:");  
 trainsList.forEach(System.*out*::println);  
  
 List<Train> trainsWithGivenDestination = Trains.*findTrainsWithSameDestination*(trainsList, "Moscow");  
 System.*out*.println("Trains for Moscow:");  
 trainsWithGivenDestination.forEach(System.*out*::println);  
  
 List<Train> trainsWithGivenDestinationAndTime = Trains  
 .*findTrainsWithSameDestinationAndAfterTime*(trainsList, "Paris", "09:00:00");  
 System.*out*.println("Trains for Paris and 09:00:");  
 trainsWithGivenDestinationAndTime.forEach(System.*out*::println);  
  
 List<Train> trainsWithGivenDestinationAndSeats = Trains  
 .*findTrainsWithSameDestinationAndHaveCommonSeats*(trainsList, "Ekaterinburg");  
 System.*out*.println("Trains for Ekaterinburg and seats:");  
 trainsWithGivenDestinationAndSeats.forEach(System.*out*::println);  
 }  
}

// Даниелян Гор Артурович ИС18-2К стр 87 Задания к главе 3 Вариант А задание 10(по номеру журнала)  
public class Train {  
 private String destination;  
 private String number;  
 private String timeOfDeparture;  
  
 private EnumMap<CarType, Integer> seatCountHolder;  
  
 public enum CarType {  
 *COMMON*, *COMPARTMENT*, *RESERVED\_SEAT*, *LUXURY*;  
 }  
  
 public Train(String destination, String number, String timeOfDeparture,  
 final int commonSeatsCount, final int compartmentSeatsCount,  
 final int reservedSeatCount, final int luxurySeatCount ) {  
 this.destination = destination;  
 this.number = number;  
 this.timeOfDeparture = timeOfDeparture;  
  
 this.seatCountHolder = new EnumMap<CarType, Integer>(CarType.class);  
  
 seatCountHolder.put(CarType.*COMMON*, commonSeatsCount);  
 seatCountHolder.put(CarType.*COMPARTMENT*, compartmentSeatsCount);  
 seatCountHolder.put(CarType.*RESERVED\_SEAT*, reservedSeatCount);  
 seatCountHolder.put(CarType.*LUXURY*, luxurySeatCount);  
 }  
  
 public String getDestination() {  
 return destination;  
 }  
  
 public void setDestination(String destination) {  
 this.destination = destination;  
 }  
  
 public String getNumber() {  
 return number;  
 }  
  
 public void setNumber(String number) {  
 this.number = number;  
 }  
  
 public String getTimeOfDeparture() {  
 return timeOfDeparture;  
 }  
  
 public void setTimeOfDeparture(String timeOfDeparture) {  
 this.timeOfDeparture = timeOfDeparture;  
 }  
  
 public int getSeatCountByCarType(final CarType type) {  
 return seatCountHolder.get(type);  
 }  
  
 public void setSeatCountByCarType(final CarType type, final int newCount) {  
 seatCountHolder.put(type, newCount);  
 }  
  
 public int getTotalSeatCount() {  
 int count = 0;  
  
 for (CarType type : CarType.*values*())  
 count = count + seatCountHolder.get(type);  
  
 return count;  
 }  
  
 @Override  
 public String toString() {  
 return "Train :\n" +  
 " destination : '" + destination + "\'\n" +  
 " number : '" + number + "\'\n" +  
 " time of departure : '" + timeOfDeparture + "\'\n" +  
 " seats : " + getTotalSeatCount() + "\n" +  
 " common seats : " + seatCountHolder.get(CarType.*COMMON*) + "\n" +  
 " compartment seats : " + seatCountHolder.get(CarType.*COMPARTMENT*) + "\n" +  
 " reserved seats : " + seatCountHolder.get(CarType.*RESERVED\_SEAT*) + "\n" +  
 " luxury seats : " + seatCountHolder.get(CarType.*LUXURY*);  
 }  
}

public class Trains {  
 private static final String[] *CITIES* = {"Moscow", "St.Petersburg", "Ekaterinburg", "Paris", "London"};  
  
 private static final ThreadLocalRandom *RNG* = ThreadLocalRandom.*current*();  
  
  
 public static List<Train> findTrainsWithSameDestination(final Collection<Train> trains, final String destination) {  
 List<Train> result = new ArrayList<>(trains);  
 return result.stream()  
 .filter(train -> Objects.*equals*(train.getDestination(), destination))  
 .collect(Collectors.*toList*());  
 }  
  
 public static List<Train> findTrainsWithSameDestinationAndAfterTime(final Collection<Train> trains,  
 final String destination,  
 final String givenTime) {  
 List<Train> result = new ArrayList<>(trains);  
 return result.stream()  
 .filter(train -> Objects.*equals*(train.getDestination(), destination))  
 .filter(train -> DatatypeConverter.*parseTime*(train.getTimeOfDeparture()).after(  
 DatatypeConverter.*parseTime*(givenTime)))  
 .collect(Collectors.*toList*());  
 }  
  
 public static List<Train> findTrainsWithSameDestinationAndHaveCommonSeats(final Collection<Train> trains,  
 final String destination) {  
 List<Train> result = new ArrayList<>(trains);  
 return result.stream()  
 .filter(train -> Objects.*equals*(train.getDestination(), destination))  
 .filter(train -> train.getSeatCountByCarType(Train.CarType.*COMMON*) > 0)  
 .collect(Collectors.*toList*());  
 }  
  
 public static List<Train> newRandomTrainList(final int trainCount) {  
 List<Train> trains = new ArrayList<>(trainCount);  
  
 for (int i = 0; i < trainCount; i++)  
 trains.add(*newRandomTrain*());  
  
 return trains;  
 }  
  
 public static Train newRandomTrain() {  
 final String city = *randomCity*();  
 final String number = *randomNumber*();  
 final String time = *randomTime*();  
  
 Train train = new Train(city, number, time, 0, 0, 0, 0);  
 for (Train.CarType type : Train.CarType.*values*())  
 train.setSeatCountByCarType(type, *RNG*.nextInt(21));  
 return train;  
 }  
  
 private static String randomCity() {  
 return *CITIES*[*RNG*.nextInt(*CITIES*.length)];  
 }  
  
 private static String randomTime() {  
 return String.*format*("%02d:%02d:%02d", *RNG*.nextInt(13), *RNG*.nextInt(60), *RNG*.nextInt(60));  
 }  
  
 private static String randomNumber() {  
 return String.*format*("%03d", *RNG*.nextInt(1000));  
 }  
}

Выполнение

Original list:

Train :

destination : 'St.Petersburg'

number : '569'

time of departure : '03:28:26'

seats : 29

common seats : 7

compartment seats : 12

reserved seats : 4

luxury seats : 6

Train :

destination : 'Paris'

number : '143'

time of departure : '07:28:32'

seats : 48

common seats : 4

compartment seats : 19

reserved seats : 12

luxury seats : 13

Train :

destination : 'Moscow'

number : '771'

time of departure : '05:13:29'

seats : 23

common seats : 8

compartment seats : 3

reserved seats : 4

luxury seats : 8

Train :

destination : 'Paris'

number : '851'

time of departure : '08:30:55'

seats : 16

common seats : 6

compartment seats : 7

reserved seats : 2

luxury seats : 1

Train :

destination : 'Ekaterinburg'

number : '837'

time of departure : '12:17:47'

seats : 26

common seats : 15

compartment seats : 0

reserved seats : 6

luxury seats : 5

Train :

destination : 'St.Petersburg'

number : '550'

time of departure : '04:18:44'

seats : 29

common seats : 14

compartment seats : 7

reserved seats : 3

luxury seats : 5

Train :

destination : 'London'

number : '344'

time of departure : '03:27:25'

seats : 47

common seats : 18

compartment seats : 0

reserved seats : 14

luxury seats : 15

Train :

destination : 'Moscow'

number : '255'

time of departure : '10:43:01'

seats : 57

common seats : 18

compartment seats : 2

reserved seats : 18

luxury seats : 19

Train :

destination : 'Moscow'

number : '429'

time of departure : '08:08:09'

seats : 42

common seats : 1

compartment seats : 16

reserved seats : 14

luxury seats : 11

Train :

destination : 'London'

number : '086'

time of departure : '07:23:58'

seats : 26

common seats : 13

compartment seats : 2

reserved seats : 5

luxury seats : 6

Trains for Moscow:

Train :

destination : 'Moscow'

number : '771'

time of departure : '05:13:29'

seats : 23

common seats : 8

compartment seats : 3

reserved seats : 4

luxury seats : 8

Train :

destination : 'Moscow'

number : '255'

time of departure : '10:43:01'

seats : 57

common seats : 18

compartment seats : 2

reserved seats : 18

luxury seats : 19

Train :

destination : 'Moscow'

number : '429'

time of departure : '08:08:09'

seats : 42

common seats : 1

compartment seats : 16

reserved seats : 14

luxury seats : 11

Trains for Paris and 09:00:

Trains for Ekaterinburg and seats:

Train :

destination : 'Ekaterinburg'

number : '837'

time of departure : '12:17:47'

seats : 26

common seats : 15

compartment seats : 0

reserved seats : 6

luxury seats : 5