חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"א

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

היבטים שונים של ייצוג מבני נתונים בעזרת מערך של תורים / מערך של מחסניות

תוך שימת דגש על גנריות מבני הנתונים

שאלה 1 – תחנת הרכבת קרית שלום – פתרון לגרסת ג'אווה

כתיבה ועריכה:

שרה וגנר

דורית ליקרמן

אהובה שפרלינג

שאלה 1 – פתרון בסביבת Java:

המחלקה נוסע – Passenger:

```
public class Passenger
{
	private String name;
	private String destination;
	private int type;	/* 0-citizens, 1-soldiers boys, 2 - sodiers girls */

	public Passenger(String name, String destination, int type)
	{
		this.name = name;
		this.destination = destination;
		this.type = type;
	}

	public String getDestination()
	{
		return this.destination;
	}

	public String getName()
	{
		return this.name;
	}

	public int getType()
	{
		return this.type;
	}

	public void setDestination(String destination)
	{
		this.destination = destination;
	}

	public String toString()
	{

		return "Passneger "+this.name+" of type "+this.type+"goes to
	"+this.destination;
	}

}
```

המחלקה אוטובוס – Bus:

```
public class Bus
{
	private int number;
	private String destination;
	private int freePlaces;

	public Bus(int num, String dest, int places)
	{
		this.number=num;
		this.destination=dest;
		this.freePlaces=places;
	}

	public String getDestination()
	{
		return this.destination;
	}

	public int getFreePlaces()
	{
		return this.freePlaces;
	}

	public int getNumber()
	{
		return this.number;
	}

	public void setDestination(String destination)
	{
		this.destination = destination;
	}

	public void setFreePlaces(int freePlaces)
	{
		this.freePlaces = freePlaces;
	}

	public void setNumber(int number)
	{
		this.number = number;
	}

	public String toString()
	{

		return "Bus no: "+this.number+" to "+this.destination+"has
		"+this.freePlaces+" free places";
	}
}
```

המחלקה FixedPriorityQueue:

```
import unit4.collectionsLib.Queue;

public class FixedPriorityQueue<T>
{
	private Queue<T> [] fpq;


	public FixedPriorityQueue (int n)
	{
		this.fpq = new Queue [n];
		for (int i=0;i<n;i++)
			this.fpq[i]=new Queue<T>();
	}
	public Queue<T>[]getFpq()
	{
		return this.fpq;
	}
	public boolean isEmpty()
	{
		int n=this.fpq.length-1;
		while(n>=0)
		{
			if (!this.fpq[n].isEmpty())
				return false;
			n--;
		}
		return true;
	}


	public T removeFixedQueue ()
	{
		int n=this.fpq.length-1;
		while(this.fpq[n].isEmpty())
			n--;
		if (n<0)
			return null;
		else
			return this.fpq[n].remove();

	}

	public void insertFixedQueue(T item, int pr)
	{
		this.fpq[pr].insert(item);
	}
}
```

המחלקה BusStop:

```
public class BusStop
{
	private FixedPriorityQueue<Passenger> station;

	public BusStop(int n)
	{
		this.station=new FixedPriorityQueue<Passenger>(n);
	}

	public void printPassengerGetIntoBus(Bus b)
	{
		FixedPriorityQueue<Passenger> temp=new
	FixedPriorityQueue<Passenger>(this.station.getFpq().length);
		Passenger person;
		int p=this.station.getFpq().length;
		int free=b.getFreePlaces();
		while((free>0) && (!this.station.isEmpty()))
		{
			person=this.station.removeFixedQueue();
			if (person.getDestination().equals(b.getDestination()))
			{
				System.out.println(person.getName());
				free--;
			}
			else
				temp.insertFixedQueue(person, person.getType());
		}
		while(!this.station.isEmpty())
		{
			person=this.station.removeFixedQueue();
			temp.insertFixedQueue(person,person.getType());
		}
		while(!temp.isEmpty())
		{
			person=temp.removeFixedQueue();
			this.station.insertFixedQueue(person,person.getType());
		}
	}

	public FixedPriorityQueue<Passenger> getBs()
	{
		return this.bs;
	}
}
```

פעולה ראשית:

```
public class Main
{

	public static void main(String[] args)
	{
		BusStop bs=new BusStop(3);
		Bus b1=new Bus(115,"Haifa",4);
		Passenger p1=new Passenger("dorit","Haifa",2);
		Passenger p2=new Passenger("dor","Tel-Aviv",2);
		Passenger p3=new Passenger("doron","Haifa",2);
		Passenger p4=new Passenger("doriti","Haifa",1);
		bs.getBs().insertFixedQueue(p1, 2);
		bs.getBs().insertFixedQueue(p2, 2);
		bs.getBs().insertFixedQueue(p3, 2);
		bs.getBs().insertFixedQueue(p4, 1);
		bs.printPassengerGetIntoBus(b1);
	}
}
```