TOBB ETÜ, BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL 113/012 Computer Programming I

HOMEWORK 6 (40 Points)

Dec 2, 2020

1 [30 POINTS] BUS TICKET RESERVATION SYSTEM

In this task, you are going to simulate an online bus ticket reservation system. You will write the following classes: *BusCompany*, *Trip*, *Date*, *Ticket*, and *TripReservation*.
The trip class represents a trip and holds various information such as source and destination locations, date of the trip, etc. BusCompany class represents Bus companies which have a name and lots of designated trips. Date class is a wrapper class that will help you represent and compare dates. Combining these, there is the TripReservation class that can display the trips the user wants to see. For instance, it can show all the trips between given dates, or it can show the trips for which the bus has 2+1 seats in a row, or it can show the cheapest trip, etc. Also, it should be able to create and cancel tickets and control if there is an empty seat on a given trip. Tickets hold information about the customer and have a unique (distinct) PNR number that can be used to cancel the ticket. You can find the detailed descriptions below.

Trip Class

- The Trip Class has the following instance variables: String companyName, source, destination, depertureTime, arrivalTime and seatType, int ticketPrice, numberOfSeats, and numberOfTicketsSold, and Date date. companyName refers to the name of the company that owns the bus. source variable refers to the name of the departure city and destination refers to the arrival city. departureTime and arrivalTime refer to the time of departure and arrival respectively and takes values between "00:00" to "23:59". seatType refers the to seat arrangement inside the bus. It can be either "2+2" or "2+1". date refers to the date of the trip in "dd/mm/yyyy" format. numberOfTickets refers to the number of seats on the bus and numberOfTicketsSold refers to how many of them are sold. ticketPrice refers to the price of a ticket and it can take integer values.

- The Trip class should have the constructor that takes companyName, source, destination, ticketPrice, date, departureTime, arrivalTime, seatType, numberOfSeats as parameters in that order. It should set numberOfTicketsSold variable to 0.
- You should use the following toString method given below. You can copy-paste it.

```
public String toString() {
        return String.format("%s tarihinde %s saatinde %s sehrinden
         %s sehrine %s firmasinin %s araci bulunmaktadir.
          Fiyat: %d TL, bos koltuk sayisi:%d",
        date.toString(), departureTime, source, destination, companyName, seatType,
         ticketPrice, (numberOfSeats - numberOfTicketsSold));
}
```

Date Class

- The Date Class has the following instance variables: int day, month, and year.
- The Date class should have two constructors given below:
  public Date(String date), the parameter date has the form "DD/MM/YYYY" e.g., "01/10/2020"
  public Date(int day, int month, int year)
- The Date class should have the following toString method:
  public String toString() { return String.format("%d/%d/%d", day, month, year); }
- I suggest you write a method that compares to date object and decides which one is earlier, later, or equal.

BusCompany Class

- The BusCompany class has the following instance variables: String name and ArrayList<Trip> designatedTrips
- The BusCompany class should have the constructor given below:
  public BusCompany(String name, ArrayList<Trip> designatedTrips)

TripReservation Class The TripReservation class should be able to list all trips or the cheapest trip that suits the user's request. Users may want to filter trips based on the seat type or may want to list all trips between the given two dates. Also, it should be able to create (purchase) and cancel tickets for a given trip if there are empty seats.

- The TripReservation class has the following instance variables:
  ArrayList<BusCompany> registeredBusCompanies, which holds the bus companies in the system.
  ArrayList<Ticket> soldTickets, which holds the tickets that are purchased (it will be needed when canceling tickets).
  ArrayList<Trip> designatedTrips which holds all trips that registered companies have.

- The TripReservation class should have the following constructor:
  public TripReservation(ArrayList<BusCompany> registeredBusCompanies): This constructor should setup designatedTrips list from the registeredBusCompanies list and initialize soldTickets list.

- The TripReservation class should have the following methods.

  public void printAllRegisteredCompanies(): Prints the name of each registered company in a seperate line.

  public ArrayList<Trip> findAllTrips(String source, String destination, Date date1, Date date2, String seatType): This is the most general search method. This method should find and return all trips from source to destination between the dates date1 and date2, inclusive. If the seatType parameter is "2+2" then it should return all trips that suit the parameters before. If it is "2+1" then it should only return the trips whose seatType is "2+1". You can use this method to create more basic methods with fewer parameters given below.

  public ArrayList<Trip> findAllTrips(String source, String destination, Date date)

  public ArrayList<Trip> findAllTrips(String source, String destination, Date date, String seatType)

  public ArrayList<Trip> findAllTrips(String source, String destination, Date date1, Date date2)

public Trip findCheapestTrip(String source, String destination, Date date, Date date2, String seatType): This method is similar to the most general findAllTrips method but only returns the trip whose price is the lowest. Similarly, you can use this method to create more basic methods with fewer parameters given below.

public Trip findCheapestTrip(String source, String destination, Date date, String seatType)

public Trip findCheapestTrip(String source, String destination, Date date, Date date2)

public Trip findCheapestTrip(String source, String destination, Date date)

public Ticket sellTicket(Trip trip, String customerName, String customerSurname, String gender): This method takes customer information and a trip as a parameter and creates a Ticket for that customer for that Trip. **Each ticket has a unique pnrNo which will be used while canceling a ticket.** You should store this ticket in the soldTickets ArrayList since you may need to cancel this ticket later on. Also, you should update the trip's numberOfTicketsSold variable. **If there are no more empty seats on that trip, this method should return null.**

public boolean cancelTicket(String customerSurname, String pnrNo): This method takes the surname of the customer and the unique PNR number of the ticket and checks whether that ticket exists. If the ticket exists, then it should cancel the ticket, update the numberOfTicketsSold variable corresponding to the trip on that ticket and return true. Otherwise, it should return false.

Ticket Class

- The Ticket class has the following instance variables: String customerName, customerSurname, gender, pnrNo, and Trip trip. As I mentioned earlier, **pnrNo must be unique**. You must make sure that the same pnrNo cannot be used for two tickets. You can deal with this in this class or inside the TripReservation class.
- The class should have the following constructor: public Ticket(String customerName, String customerSurname, String gender, Trip trip, String pnrNo): Depending on how you handle unique PNR no's, you may omit the String pnrNo parameter.
- The Ticket class should have the toString method given below.

```
public String toString() {
  return String.format(
  "Ad soyad:%s %s Yolculuk tarihi:%s Saat:%s Kalkis:%s Varis:%s
  Firma:%s Koltuk Tipi:%s PNR:%s",customerName, customerSurname,
    trip.getDate().toString(), trip.getDepartureTime(), trip.getSource(),
     trip.getDestination(), trip.getCompanyName(), trip.getSeatType(), pnrNo);
}
```

You can and should add additional methods or variables to any class you want. We shared a driver class named TripReservationDemo that tests the given methods. Put the TripReservationDemo.java file along with your .java files into the folder Q1. So, the Q1 folder should have at least the following files: ***Trip.java, TripReservation.java, TripReservationDemo.java, Ticket.java, Date.java, and BusCompany.java***.
You can find the expected run of the program below. The order of the companies and buses can be different, but it must contain all buses listed. So do not worry if the order is not the same. Texts that do not fit in a single line are automatically divided into two lines, e.g., "1/12/2020 tarihinde 23:30 saatinde Mersin sehrinden Ankara sehrine Koksallar firmasinin 2+2 araci bulunmaktadir. Fiyat: 65 TL, bos koltuk sayisi:48" is actually one line. You can find the .txt version of the output and the TripReservationDemo.java under the resource section.

EXAMPLE 1
Kayitli sirketler:
Luks Mersin
Mersin Vif
Koksallar
Metro

1/12/2020 tarihinde Mersinden Ankara'ya giden araclar:
1/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Ankara sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 60 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 09:00 saatinde Mersin sehrinden Ankara sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 60 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 13:00 saatinde Mersin sehrinden Ankara sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 60 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 17:00 saatinde Mersin sehrinden Ankara sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 55 TL, bos koltuk sayisi:54
1/12/2020 tarihinde 13:00 saatinde Mersin sehrinden Ankara sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 70 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 23:30 saatinde Mersin sehrinden Ankara sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 70 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Ankara sehrine Koksallar firmasinin 2+1 araci bulunmaktadir. Fiyat: 70 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 16:00 saatinde Mersin sehrinden Ankara sehrine Koksallar firmasinin 2+1 araci bulunmaktadir. Fiyat: 70 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 23:30 saatinde Mersin sehrinden Ankara sehrine Koksallar firmasinin 2+2 araci bulunmaktadir. Fiyat: 65 TL, bos koltuk sayisi:48

En uygun fiyatli arac
1/12/2020 tarihinde 17:00 saatinde Mersin sehrinden Ankara sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 55 TL, bos koltuk sayisi:54

1/12/2020 tarihinde Mersinden Istanbul'a giden araclar:

1/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 100 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
1/12/2020 tarihinde 20:30 saatinde Mersin sehrinden Istanbul sehrine Koksallar firmasinin 2+2 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:36
1/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1

En uygun fiyatli arac
1/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 100 TL, bos koltuk sayisi:48

3/12/2020 ve 5/12/2020 arasi tarihlerde Mersinden Istanbul'a giden araclar:
3/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
4/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
5/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
3/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 100 TL, bos koltuk sayisi:48
4/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 100 TL, bos koltuk sayisi:48
5/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 100 TL, bos koltuk sayisi:48
3/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
4/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
5/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
3/12/2020 tarihinde 20:30 saatinde Mersin sehrinden Istanbul sehrine Koksallar firmasinin 2+2 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:36
4/12/2020 tarihinde 20:30 saatinde Mersin sehrinden Istanbul sehrine Koksallar firmasinin 2+2 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:36
5/12/2020 tarihinde 20:30 saatinde Mersin sehrinden Istanbul sehrine Koksallar firmasinin 2+2 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:36
3/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1
4/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1

5/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1

En uygun fiyatli arac
3/12/2020 tarihinde 09:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+2 araci bulunmaktadir. Fiyat: 100 TL, bos koltuk sayisi:48

3/12/2020 ve 5/12/2020 arasi tarihlerde Mersinden Istanbul'a giden 2+1 araclar:
3/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
4/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
5/12/2020 tarihinde 00:30 saatinde Mersin sehrinden Istanbul sehrine Luks Mersin firmasinin 2+1 araci bulunmaktadir. Fiyat: 120 TL, bos koltuk sayisi:48
3/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
4/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
5/12/2020 tarihinde 18:00 saatinde Mersin sehrinden Istanbul sehrine Mersin Vif firmasinin 2+1 araci bulunmaktadir. Fiyat: 125 TL, bos koltuk sayisi:48
3/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1
4/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1
5/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1

En uygun fiyatli 2+1 arac
3/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1

Bu bileti alalim
Bilgileriniz asagidaki gibidir.
Ad soyad:Utku Umur ACIKALIN Yolculuk tarihi:3/12/2020 Saat:21:00 Kalkis:Mersin Varis:Istanbul Firma:Metro Koltuk Tipi:2+1 PNR:0

Bilet satin alindiktan sonraki durum
3/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:0

Aynı otobusten bilet almaya calisalim
Otobuste yer olmadigi icin bilet alinamadi

Aldigimiz bileti iptal edelim
Bilet basariyla iptal edildi.
Iptal ettiğimiz bileti bir daha iptal etmeye calisalim
Belirtmis oldugunuz bilet sistemde bulunamadi.

Bilet iptal edildikten sonraki durum
3/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:1

Tekrar alalim
Bilgileriniz asagidaki gibidir.
Ad soyad:Utku Umur ACIKALIN Yolculuk tarihi:3/12/2020 Saat:21:00 Kalkis:Mersin Varis:Istanbul Firma:Metro Koltuk Tipi:2+1 PNR:1

Bilet satin alindiktan sonraki durum
3/12/2020 tarihinde 21:00 saatinde Mersin sehrinden Istanbul sehrine Metro firmasinin 2+1 araci bulunmaktadir. Fiyat: 115 TL, bos koltuk sayisi:0

2 [10 POINTS] FRACTIONAL COMPLEX NUMBERS

In this task, you are going to write a Java class named ***ComplexNumber*** which will support various operations on complex numbers such as addition, subtraction, etc.
Fractional Complex numbers consists of a *real part* and *imaginary part* which are both fractional numbers. So your class should have them as instance variables. You should use the FractionalNumber class we shared as a solution for Homework #5. You can find the methods that you need to implement below.

- *ComplexNumber* class should have two constructors. One of them should take a String as a parameter. The other one should take two FractionalNumbers (realPart and imaginaryPart) as a parameter. You can assume that the string parameter has the form "(+-)FractionalNumber(+-)FractionalNumber". The first + is optional, but the second is not. For example, the parameter can be one of the following: "$-4/3-5/4i$", "$4/3-5/4i$", "$+4/3-5/4i$", "$-4/3+5/4i$", "$+4/3+5/4i$".

- public String toString(): This method should return string representation of the ComplexNumber which is formatted as "*realPart*(+-)*imaginaryPart*". For example if the *realPart* is $4/3$ and *imaginaryPart* is $5/4$, then method should return "$4/3+5/4i$". It should omit the first +. If the *realPart* is $-4/3$ and *imaginaryPart* is -5/4, then method should return "$-4/3-5/4i$".

- public boolean equals(ComplexNumber): This method should return true if the ComplexNumber given in the parameter is equivalent to this one.

- public static ComplexNumber simplify(ComplexNumber cn): This method should return the simplified version of the given ComplexNumber. For example, if the complex number is $4/8+2/4i$, then this method should return ComplexNumber $1/2+1/2i$. Basically, it needs to simplify both *realPart* and *imaginaryPart*. You can use the *FractionalNumber.simplify()* method since both *realPart* and *imaginaryPart* are FractionalNumber.

- public static ComplexNumber add(ComplexNumber cn1, ComplexNumber cn2): This method should return the simplified version of the summation of the two given ComplexNumber. For example, if the first complex number is $4/8+4/8i$ and the second fractional number is $2/8+2/8i$ then this method should return the ComplexNumber $3/4+3/4i$.

- public static ComplexNumber subtract(ComplexNumber cn1, ComplexNumber cn2): This method should return the simplified version of the difference between the two given ComplexNumber. For example, if the first complex number is $4/8+3/6i$ and the second complex number is $2/8-3/6i$ then this method should return the ComplexNumber $1/4+0/1i$.

- public static ComplexNumber multiply(ComplexNumber cn1, ComplexNumber cn2): This method should return the simplified version of the product of the two given ComplexNumber. For example, if the first complex number is $1/1+1/1i$ and the second complex number is $2/2-4/4i$ then this method should return the ComplexNumber $0/1+2/1i$.

- public static ComplexNumber divide(ComplexNumber cn1, ComplexNumber cn2): This method should return the simplified version of the division of the two given ComplexNumber. If the second complex number equals to 0 then this method should return null. For example, if the first complex number is $1/1+1/1i$ and the second complex number is $2/1-1/i$ then this method should return the ComplexNumber $1/5+3/5i$. But if the first fractional number is $4/8+4/8i$ and the second fractional number is $0/1+0/1i$ then this method should return null.

We shared a demo class named ***ComplexNumberDemo***. Put this file to Q2 folder with your ***ComplexNumber.java*** and the ***FractionalNumber.java***. You can use this class to check your implementation is correct. If you could not implement some methods before submitting your homework, write methods with the same signature and make them return the default value of the return type. For example, if you could not implement the subtract method, then write the following.

```
public static ComplexNumber subtract(FractionalNumber cn1, ComplexNumber cn2){
        return null;
}
```

Otherwise, your code will not compile. Example outputs are given below. Green texts represent the input.

Example 1
Please enter the first complex number in string representation
Please enter the first complex number in string representation
1/1+1/1i
Please enter the second complex number in string representation
1/1-1/1i
Please enter the Third complex number in string representation
1/1+5/1i
First complex number is 1/1+1/1i and its simplified version is 1/1+1/1i
Second complex number is 1/1-1/1i and simplified version is 1/1-1/1i
Third complex number is 1/1+5/1i and simplified version is 1/1+5/1i

First and second complex numbers are not equal
First and third complex numbers are not equal
Second and third complex numbers are not equal

1/1+1/1i + 1/1-1/1i = 2/1+0/1i
1/1+1/1i + 1/1+5/1i = 2/1+6/1i
1/1-1/1i + 1/1+5/1i = 2/1+4/1i

1/1+1/1i - 1/1-1/1i = 0/1+2/1i
1/1+1/1i - 1/1+5/1i = 0/1-4/1i
1/1-1/1i - 1/1+5/1i = 0/1-6/1i

1/1+1/1i * 1/1-1/1i = 2/1+0/1i
1/1+1/1i * 1/1+5/1i = -4/1+6/1i
1/1+5/1i * 1/1-1/1i = 6/1+4/1i

1/1+1/1i / 1/1-1/1i = 0/1+1/1i
1/1+1/1i / 1/1+5/1i = 3/13-2/13i
1/1+5/1i / 1/1-1/1i = -2/1+3/1i

Example 2
Please enter the first complex number in string representation
7/4-3/2i
Please enter the second complex number in string representation
3/14+1/45i
Please enter the Third complex number in string representation
1/2-0/1i
First complex number is 7/4-3/2i and its simplified version is 7/4-3/2i
Second complex number is 3/14+1/45i and simplified version is 3/14+1/45i
Third complex number is 1/2+0/1i and simplified version is 1/2+0/1i

First and second complex numbers are not equal
First and third complex numbers are not equal
Second and third complex numbers are not equal

7/4-3/2i + 3/14+1/45i = 55/28-133/90i
7/4-3/2i + 1/2+0/1i = 9/4-3/2i
3/14+1/45i + 1/2+0/1i = 5/7+1/45i

7/4-3/2i - 3/14+1/45i = 43/28-137/90i
7/4-3/2i - 1/2+0/1i = 5/4-3/2i
3/14+1/45i - 1/2+0/1i = -2/7+1/45i

7/4-3/2i * 3/14+1/45i = 49/120-89/315i
7/4-3/2i * 1/2+0/1i = 7/8-3/4i
1/2+0/1i * 3/14+1/45i = 3/28+1/90i

7/4-3/2i / 3/14+1/45i = 271215/36842-143010/18421i
7/4-3/2i / 1/2+0/1i = 7/2-3/1i
1/2+0/1i / 3/14+1/45i = 42525/18421-4410/18421i