

תכנות מתקדם

תרגיל בית מספר 1

תכנות מונחה עצמים

הנחיות:

- א. יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים – המסמך מופיע באתר הקורס – יש לקרואו בעיון!!
- ב. הקפד/י על קריאות התכנית ועל עימוד (Indentation).
- ג. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- ד. בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התכנית.
- ה. השתמש/י בשמות משמעותיים עבור המשתנים.
- ו. יש לתעד את התכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים ובנוסף, **דוגמת הרצה לכל תכנית בסוף הקובץ!**
- ז. הגשה יחידנית - אין להגיש בזוגות.

הערה חשובה: לכל תרגיל בית מוגדר שבוע אחד בלבד להגשה.
תיבות ההגשה הפתוחות לא מהוות היתר להגשה באיחור.

שאלה מס' 1:

- הגדר/י מחלקה עבור המספרים הרציונליים (Rational).
תזכורת: ערך המכנה לא יהיה 0 או שלילי
המחלקה תכלול את השדות הבאים:
- מונה (numerator)
 - מכנה (denominator)
- וכן את המתודות הבאות:
- בנאים:
 - בנאי ctor עם ערכי ברירת מחדל, ערך 0 עבור המונה וערך 1 עבור המכנה. בנוסף, יש לבצע הדפסה במימוש הבנאי:
print: in constructor
 - בנאי העתקה copy-ctor. בנוסף, יש לבצע הדפסה במימוש בנאי העתקה:
print: in copy constructor
 - עבור כל שדה:
 - מתודת הצבה (set). במכנה אין להציב 0, לכן במידה והפרמטר שמתקבל הוא 0 המתודה (setDenominator), תציב 1. שימו לב, כמו שמוגדר למעלה, אסור ערך שלילי במכנה יש לאתחל את השדות בהתאם, כלומר, במידה והמונה חיובי והמכנה שלילי יש לשנות כך שהמונה יהיה שלילי והמכנה חיובי, במידה וערך המונה והמכנה הינם שליליים הרי השבר חיובי.
 - מתודה המחזירה את ערכו (get).
 - מתודה להדפסת השבר הרציונאלי. ההדפסה תהיה בפורמט: מונה/מכנה (לדוגמא 1/2, 3/4, 54/56 וכו') עפ"י הערכים המקוריים לא לאחר צמצום.
 - מתודה בוליאנית equal המשווה בין שני מספרים רציונאליים ובודקת האם הם שווים. שימו לב: הכוונה של שווה היא שיש להם בדיוק אותו מונה ומכנה (כלומר 1/2 ו- 3/6 הם לא שווים)
 - מתודה makeEquals המקבלת כפרמטר אובייקט מסוג Rational ומעדכנת אותו להיות שווה (מונה ומכנה) לאובייקט המזמן את הפונקציה.
שים לב:
המתודה היא void.
הפרמטר המתקבל הינו cbr ולכן אין קריאה ל copy-ctor

- מתודת עזר לצמצום השבר

- חתימת המתודה: `void reduce()`;
- על המתודה לצמצם את האובייקט המזמן את המתודה כלומר, עבור (זימון שאינו ב-main):

```
Rational rat(2,4);  
rat.reduce ();
```

המונה של rat הוא 1 והמכנה של rat הוא 2.

לאחר הצמצום, אם המונה הוא 0, אז המתודה תהפוך את המכנה להיות 1.

- מתודה `add` המקבלת כפרמטר אובייקט מסוג `Rational` ומחברת בין שני מספרים רציונליים על המתודה להחזיר את סכום שני המספרים (בתור אובייקט שלישי מסוג `Rational`). על הפונקציה לצמצם את הסכום. כלומר עבור

```
Rational r1, r2;  
r1.setNumerator(1);  
r1.setDenominator(6);  
r2.setNumerator(1);  
r2.setDenominator(3);
```

```
Rational r3;  
r3 = r1.add(r2);
```

r1 הינו $1/6$

r2 הינו $1/3$

r3 הינו $1/2$

שיס/י לב:

הפרמטר המתקבל הינו `cbv` ולכן מתקיימת קריאה ל `copy-ctor` בעת זימון המתודה

- מתודה `addOne` הבונה אובייקט חדש מסוג `Rational` שערכו גדול ב-1 מהאובייקט המזמן את המתודה. כלומר עבור

```
Rational r1(2,3), r2;  
r2 = r1.addOne();
```

r2 הינו $5/3$

השתמש/י בתוכנית הראשית (`main`) הבאה כדי לבדוק את נכונות המחלקה שכתבת. התוכנית בודקת, את מתודות `ctor` ו-`copy-ctor`, `set` ו-`get`, `add` ו-`addOne`, `equals` ו-`makeEquals`.

יש להשתמש ב-main המצורף עבור הרצת התוכנית:

```
#include "Rational.h"  
#include <iostream>  
using namespace std;
```

```
int main()  
{
```

```
    int num1, num2, num3;  
    char junk;
```

```
    cout << "part 1. ctor/set/get" << endl;  
    cout << "enter a rational number:" << endl;
```

```
cin >> num1 >> junk >> num2;
Rational rat1;
rat1.setNumerator(num1);
rat1.setDenominator(num2);
cout << "numerator: " << rat1.getNumerator() << endl;
cout << "denominator: " << rat1.getDenominator() << endl;
cout << endl;

cout << "enter a rational number: " << endl;
cin >> num1 >> junk >> num2;
Rational rat2(num1,num2);
rat2.print();
cout << endl << endl;

cout << "part 2. copy-ctor" << endl;
Rational rat3(rat2);
rat3.print();
cout << endl << endl;

cout << "part 3. equals" << endl;
if (rat1.equal(rat2))
    cout << "The two numbers are equal" << endl;
else
    cout << "The two numbers are different" << endl;
cout << endl;

cout << "part 4. makeEquals" << endl;
rat1.makeEquals(rat2);
if (rat1.equal(rat2))
    cout << "The two numbers are equal" << endl;
else
    cout << "The two numbers are different" << endl;
cout << endl;

cout << "part 5. addOne" << endl;
Rational ans1;
ans1 = rat1.addOne();
rat1.print();
cout << " + 1 = ";
ans1.print();
cout << endl<<endl;

cout << "part 6. add" << endl;
Rational ans2;
ans2 = rat1.add(rat3);
rat1.print();
cout << " + ";
rat3.print();
cout << " = ";
ans2.print();
cout << endl<<endl;
return 0;
```

```
}
```

דוגמה להרצת התוכנית :

```
part 1. ctor/set/get
enter a rational number:
2/12
print: in constructor
numerator: 2
denominator: 12

enter a rational number:
4/12
print: in constructor
4/12

part 2. copy-ctor
print: in copy-constructor
4/12

part 3. equals
print: in copy-constructor
The two numbers are different

part 4. makeEquals
print: in copy-constructor
The two numbers are equal

part 5. addOne
print: in constructor
print: in constructor
2/12 + 1 = 14/12

part 6. add
print: in constructor
print: in copy-constructor
print: in constructor
2/12 + 4/12 = 1/2
```

שאלה מס' 2:

הגדרי מחלקה Employee אשר תייצג נתוני עובד בעמותת "לתת מהלב" לצורך חישוב משכורתו. המחלקה תכלול את השדות הבאים:

- מספר עובד (int) בדיוק 5 ספרות
- שם – עד 20 תווים (מערך סטטי) char name[21];
- שכר לשעה (float)
- מספר השעות שעבד (int).
- הסכום שהצליח להכניס לעמותה (float).

וכן את המתודות הבאות:

- constructor. יש לבצע הדפסה במימוש הבנאי:
print: in constructor
- עבור כל שדה:
 - מתודת הצבה (set).
 - מתודת המחזירה את ערכו (get).
- מתודה לחישוב המשכורת. חישוב המשכורת יבוצע בצורה הבאה:
מספר שעות * שכר לשעה + אחוזים מהסכום שהצליח להכניס לעמותה
האחוזים יחושבו באופן הבא:

סכום ההכנסה :	אחוזים עבור המשכורת :
עד 1000 שקלים (כולל)	10%
מ-1000 שקלים עד 2000 שקלים (כולל)	15%
מ-2000 שקלים עד 4000 שקלים (כולל)	20%
מ-4000 שקלים עד 5000 שקלים (כולל)	30%
מ-5000 שקלים ומעלה	40%

לדוגמא: עובד שהכניס לעמותה 4500 שקלים, חלק האחוזים במשכורתו יחושב כך:
 $1000 \cdot 0.1 + 1000 \cdot 0.15 + 2000 \cdot 0.2 + 500 \cdot 0.3 = 800$ כלומר, בנוסף למשכורתו הבסיסית של מספר השעות בתעריף לשעה ירוויח עובד זה עוד 800 שקלים

כתובי תוכנית ראשית אשר תקלוט פרטי עובדים עד להקשת מס' עובד 0. עבור כל עובד יקלטו הנתונים הבאים, לפי הסדר: מספר עובד, שם, שכר לשעה, שעות עבודה, הסכום שהצליח להכניס לעמותה. (ניתן להניח שקיים לפחות עובד אחד).

על התוכנית להדפיס:

- את מספר העובד ושמו של העובד שהכניס הכי פחות כסף לעמותה. בנוסף יודפס סכום הכסף שהכניס לעמותה.
- את מספר העובד ושמו של העובד שקיבל את המשכורת הגבוהה ביותר. בנוסף תודפס משכורתו של העובד.

שים לב:

- במקרה של קלט לא תקין, התוכנית תדפיס ERROR ותמשיך לקלוט את נתוני העובד-הנוכחי
- רמז: יש לכתוב את הקוד כך שהתוכנית תקלוט תחילה את כל הנתונים של העובד ורק אחר כך תבדוק את תקינותם.
- להגבלות הרלוונטיות בכל שדה.
- בתוכנית הראשית אין להקצות מערך של עובדים.

דוגמאות להרצת התוכנית:

```
print: in constructor
print: in constructor
print: in constructor
enter details, to end enter 0:
12345 moshe 50 40 2000
13579 rivka 120 55 3450
97531 sara 35 100 5632
0
minimum collected: 12345 moshe 2000
highest salary: 13579 rivka 7140
```

```
print: in constructor
print: in constructor
print: in constructor
enter details, to end enter 0:
11111 doron 35 120 6000
22222 tal 50 55 1400
44444 levi 45 -4 100
ERROR
33333 naomi 30 120 800
0
minimum collected: 33333 naomi 800
highest salary: 11111 doron 5550
```

בהצלחה רבה!!