

234124 - מבוא לתכנות מערכות

תרגיל בית מספר 3 (C++)

סמסטר אביב תשע"ט

אחראים : ישראל גוטר

נוי נאור

תאריך הגשה : יום ראשון 30.6.2019 23:59

אופן הגשה : בזוגות או ביחידים. פרטים הרלוונטיים להגשה רשומים בסוף תרגיל זה.

משקל התרגיל : 12% מהציון הסופי (תקף).

הערה : מענה לשאלות ו/או הבהרות בנושאים ענייניים הקשורים לתרגיל יינתן בפורום ב – Moodle. בנוסף ניתנות סדנאות וניתן להיעזר בהן לצורך בירור עניינים כלליים הקשורים בחומר. הודעה על קיום הסדנאות פורסמה באתר הקורס. (נא שימו לב שהסדנאות אינן מיועדות לתת שירותי debugging לקוד שאתם כותבים).

חלק יבש

בחלק זה 2 שאלות. יש להגיש את הפתרון לחלק זה מודפס (hard copy) לתא של הקורס ובנוסף יש לצרף לקובץ זיף של ההגשה האלקטרונית קובץ Word dry.docx שמכיל את כל מה שהגשתם ידנית מודפס על נייר. בהגשה על נייר הקפידו לרשום בכתב ברור וקריא שם מלא, מס זהות מדויק ב-9 ספרות וכתובת מייל מדויקת (של 2 המגישים, אם מוגש על ידי 2 מגישים).

1. תכנות גנרי

1. א. כתבו קוד גנרי אשר בהינתן Container כלשהו ובהינתן תנאי (פרדיקט)

המופעל על 2 איברים מטיפוס איברי ה-Container, הקוד סופר כמה זוגות

איברים הנמצאים ב – Container מקיימים את התנאי. הסדר בין איברי הזוג

אינו משנה ואין לספור פעמיים את איברי אותו הזוג כשהם מחליפים סדר

ביניהם. בנוסף אין לספור זוג שבו אותו איבר מופיע פעמיים, אבל אם יש שני

איברים נפרדים שיש להם אותו ערך – אזי זה כן נספר.

הערות : ה – Container יינתן באמצעות ערכי קיצון של איטרטור. מותר

להניח שה-Container תומך באיטרטור שמספק את האופרטורים הבאים :

++ שמאלי, *, <, == בנוסף יש גם תמיכה ב-ctr מתאים וב-dtr.

מעבר לכך אין תמיכה באופרטורים נוספים.

1. ב. ממשו את הפונקציה

```
bool isSorted(vector<int> v)
```

הפונקציה בודקת האם v מונוטוני עולה ממש משמאל לימין (ממין מהקטן לגדול משמאל לימין). אם כן – מוחזר `true`. במקרה אחר מוחזר `false`.
דרישה : המימוש חייב להיעשות באמצעות שימוש בקוד הגנרי שכתבתם ב-1.א.
הערה : אין צורך להידרש כאן לנושא סיבוכיות/יעילות.

2. ירשה ופולימורפיזם
נתון הקוד הבא:

```
class Base
{
    void method() {std::cout << "from Base" << std::endl;}
public:
    virtual ~Base() {method();}
    void baseMethod() { method();}
};

class A: public Base
{
    void method() {std::cout << "from A" << std::endl;}
public:
    ~A () { method();}
};

int main (void)
{
    Base* base = new A();
    base -> baseMethod();
    delete base;
    return 0;
}
```

- א. רשמו מה ידפיס הקוד הנ"ל ונמקו מדוע.
ב. כיצד ניתן לשנות את השורה הראשונה בפלט על ידי שינוי המחלקה Base בלבד? נמקו את תשובתכם.

ג. בתגובה לביצוע שורת הקוד

```
Base* base = new A();
```

ברצוננו לקבל את הפלט הבא :

from Base

from A

הראו כיצד ניתן לקבל זאת בלי להוסיף ו/או לשנות פקודות פלט (אופרטור <<).

חלק רטוב

בחלק זה של התרגיל אנו נמשיך ונעסוק בנושא הארוויזיון.

חלק א

נתון הקובץ `test1.cpp`. התוכנית הכתובה בו עושה שימוש בישויות הבאות :

מחלקה Participant

מחלקה זו מייצגת משתתף פוטנציאלי בתחרות הארוויזיון. משתתף כזה מאופיין על ידי :
שם מדינה, שם שיר, שם זמר (להקה), משך זמן השיר (בשניות).
כל עוד עדיין לא בוצעה הרשמה פורמלית של המשתתף הפוטנציאלי לתחרות, ניתן לעדכן בו את
הערכים המאפיינים אותו (למעט שם מדינה).

מחלקה Voter

מחלקה זו מייצגת מצביע בתחרות הארוויזיון.
כל מצביע משתייך למדינה מסוימת. למצביע אסור להצביע עבור המדינה של עצמו. המדינות
הרלוונטיות של המצביעים הן גם המדינות שמשתתפות בתחרות הארוויזיון. מדינה שאינה
משתתפת בתחרות הארוויזיון אינה רשאית להצביע.
קיימים 2 סוגים של מצביעים :
Regular – זהו מצביע רגיל אשר בכל הצבעה שלו הוא מצביע עבור מדינה אחת בלבד. כל הצבעה
שווה נקודה אחת. מצביע רגיל מורשה להצביע יותר מפעם אחת (עד לחסם מקסימלי שמוגדר
כפרמטר במערכת הארוויזיון).
Judge – זהו מומחה מקצועי אשר מצביע לפי קריטריונים מקצועיים. הוא רשאי להצביע פעם
אחת בלבד ובהצבעה שלו הוא מציין לפי סדר עד 10 מדינות (בלי המדינה שלו, כמובן) אשר
מקבלות לפי הסדר את הניקוד 1 2 3 4 5 6 7 8 10 12.

מבנה Vote

זהו, למעשה, `struct` עזר אשר משתמשים בו בפעולת הצבעה. המבנה מפרט מי מצביע ועבור מי
הוא מצביע. למעט `ct'r` ו- `dt'r` אין צורך להגדיר בו שום מתודות נוספות.

מחלקה MainControl

מחלקה זו מייצגת את מנהלת הארוויזיון. היא זו שמנהלת ומרכזת את רישום המתחרים ובדיקת
חוקיותם. בנוסף היא מנהלת את כל תהליך ההצבעה, בדיקת החוקיות וריכוז התוצאות.

הרשמת משתתפים נעשית באמצעות הפעולה

`+=`

פעולת הצבעה מסומנת גם היא כפעולת

`+=`

מנהלת הארוויזיון יכולה להימצא באחד מתוך 3 שלבים :

Registration – בשלב זה מבוצע רישום של המשתתפים השונים (או ביטול רישום של משתתפים קיימים שהתחרטו).

Contest – בשלב זה מבוצעים השירים על במת הארוויזיון. שלב זה לא בא לידי ביטוי בתוכנית שלנו כאן.

Voting – זה השלב שבו מבוצעת ההצבעה ומוצגות התוצאות.

המשימות בחלק א של התרגיל :

1. כאמור מצורף הקובץ `test1.cpp`. התוכנית הכתובה בו מדגימה באופן מלא את השימוש האפשרי במחלקות ו- `struct` לעיל. בנוסף מופיע תיעוד מלא ומפורט על השימוש בכל הפעולות. הקובץ `test1_out.txt` מכיל את הפלט הטקסטואלי המתקבל בהרצת התוכנית.
קראו בעיון רב את הקובץ `test1.cpp` תוך שאתם נעזרים גם בקובץ הפלט `test1_out.txt`. וודאו שהנכם מבינים את השימוש המדויק שנעשה בישויות השונות כמוזכר לעיל.
2. נתון הקובץ `eurovision.h`
עליכם להשלים בקובץ זה את כל ההגדרות הנחוצות על מנת שתהיה תאימות מלאה עם התוכנית המופיעה ב- `text1.cpp`
בחלק זה של התרגיל אסור להשתמש בתבניות וגם לא בירושה. אסור להשתמש ב- `STL`.
הקפידו לתמוך ולהתרכז אך ורק בפעולות כפי שבאות לידי ביטוי ב- `test1.cpp`
אין צורך להוסיף פעולות ממשק נוספות. מותר להוסיף פעולות עזר למימוש פנימי, במידת הצורך.
מצבי שגיאה, ערכי פרמטרים לא חוקיים וכו' – בחלק זה של התרגיל איננו מחזירים כלל סטטוס של שגיאה ואיננו זורקים כלל מצבים חריגים `exceptions`. פעולה שאינה יכולה להתבצע עקב מצב חריג כזה או אחר פשוט לא תתבצע ללא כל חיווי אחר, והתוכנית תמשיך לפקודות הבאות. אנו מניחים כאן שלא יהיה מצב חריג קריטי שלא יאפשר המשך ביצוע תקין של התוכנית.
3. כתבו בעצמכם את הקובץ `eurovision.cpp` אשר מכיל את כל המימושים הנדרשים מההגדרות המופיעות ב- `eurovision.h`
4. וודאו שהקוד שכתבתם מתקמפל ב- `CSL3` עם כל הדגלים הנדרשים ללא שום שגיאה ו/או אזהרה. הריצו את התוכנית `test1.cpp` ביחד עם הקוד שכתבתם וודאו שהפלט הטקסטואלי שמתקבל זהה לפלט הדוגמה שקיבלתם. הקפידו גם על זהות של פורמט – רווחים, ירידת שורה וכו'.

חלק ב :

בחלק זה של התרגיל נרחיב את חלק א.

נגדיר `template` ונשתמש בו. בנוסף, בהתאם לצורך, מותר להשתמש ב- `STL`.

ב.1. הגדירו פונקציה כללית (`template`) ששמה `get` אשר מוצאת את האיבר ה- `i` – י הגדול ביותר ב- `container` כלשהו. ה- `container` ניתן באמצעות ערכי קצה של איטרטור לתחילת ה- `container` ולסופו (מקום אחד אחרי האיבר האחרון). הסדר לפיו ייקבע מיהו האיבר ה- `i` – י הגדול ביותר אינו בהכרח סדר ההתקדמות של האיטרטור על פני איברי ה- `container`. ניתן להניח שה- `container` אינו מכיל איברים שווים. אסור לשנות את ה- `container`. הפונקציה מחזירה כתוצאה איטרטור לאיבר ה- `i` – י הגדול ביותר. (`i` שווה 1 משמעו האיבר המקסימלי. `i` שווה 2 משמעו האיבר שמבחינת גודלו בא אחרי האיבר המקסימלי וכו'). אם מסיבה כלשהי לא קיים איבר כזה (`i` לא חוקי/ גדול מדי וכו') אזי יוחזר כתוצאה ערך קצה של האיטרטור למקום אחד אחרי האיבר האחרון ב- `container`.

הערה : אין דרישה מיוחדת לגבי סיבוכיות / יעילות.

ב.2. נתונים 3 הטסטים הבאים, בהמשך לחלק א :

TEST("1.7.1")

```
MainControl::Iterator i;
```

```
for (i = eurovision.begin(); i<eurovision.end(); ++i)
```

```
    cout << *i << endl;
```

```
// we get here the display of all the participants that are registered in the
```

```
// eurovision. They are displayed sorted by state name, from small to big.
```

TEST("1.7.2")

```
for (i = eurovision.begin(); !(i==eurovision.end()); ++i)
```

```
    cout << *i << endl;
```

```
// the same as 1.7.1
```

TEST("1.7.3")

```
// --i; // compilation error. operator -- does not exist for i.
```

הוסיפו ב- `MainControl` class את כל התוספות הדרושות על מנת לתמוך בטסטים אלה.

3.ב בהמשך לחלק א, בסוף תהליך ההצבעה, ברצוננו לתמוך בפעולות הבאות :

TEST("1.17.1")

```
cout << eurovision(1, Regular) << endl;

/* We get here displayed the name of the state that won 1st place
according to Regular votes */

/* if 2 states have the same num of points, then first comes the state
whose name is bigger. i.e. if Israel and UK got together the max num
points, then place 1 is for UK and place 2 is for Israel. */

cout << eurovision(2, Regular) << endl;

/* the state that won 2nd place according to Regular votes */

cout << eurovision(3, Regular) << endl;

/* the state that won 3rd place according to Regular votes */

cout << eurovision(4, Regular) << endl;

/* the state that won 4th place according to Regular votes */

cout << eurovision(5, Regular) << endl;

/* the state that won 5th place according to Regular votes */

/* if no such winner, then "" is displayed */
```

TEST("1.17.2")

```
/* the same as TEST 1.17.1 but this time according to Judge votes */

cout << eurovision(1, Judge) << endl;
cout << eurovision(2, Judge) << endl;
cout << eurovision(3, Judge) << endl;
cout << eurovision(4, Judge) << endl;
cout << eurovision(5, Judge) << endl;
```

TEST("1.17.3")

/* the same as TEST 1.17.2 but this time according to All votes, which means points==(Regular points)+(Judge points) */

```
cout << eurovision(1, All) << endl;  
cout << eurovision(2, All) << endl;  
cout << eurovision(3, All) << endl;  
cout << eurovision(4, All) << endl;  
cout << eurovision(5, All) << endl;
```

הוסיפו את מה שנדרש ב – MainControl על מנת לתמוך בטסטים אלה.

מימוש הפעולה `eurovision(<i>, <VoterType>)`

חייב להיעשות תוך שימוש בפונקציה הכללית `get` שכתבתם בסעיף ב.1.

הערה : ניתן להניח שטסטים אלה מבוצעים בסוף תהליך ההצבעה. אין צורך לבדוק זאת.

אם מסיבה כזו או אחרת לא ניתן לקבל כתוצאה שם של מדינה אזי, כאמור, מקבלים "".

הקבצים המצורפים `test2.cpp` `test2_out.txt` מדגימים את הטסטים לעיל. וודאו שאתם מקבלים פלט הזהה לזה המופיע ב - `test2_out.txt`.

חלק ג :

בחלק זה של התרגיל נעסוק במקומות הישיבה השונים המוקצים לקהל הרב שבא לאולם הארוויזיון לצפות בתחרות.

לכל מקומות הישיבה (ללא יוצא מן הכלל) מוגדרים מס' שורה ומס' כסא, וכן מחיר בסיס.

מחיר בסיס הוא מחיר אשר משמש כבסיס לחישוב המכיר האמיתי של מקום הישיבה. יחד עם זאת, ישנם מקומות ישיבה שעבורם המחיר מחושב ללא התחשבות במחיר הבסיס, למרות שעקרונית יש להם מחיר בסיס.

לכל מקומות הישיבה (מכל הסוגים) מוגדרת הפעולות `location()` ו - `price()` אשר מספקות מידע על המיקום ועל המחיר.

הפעולה `location()` מספקת בכל מקרה פרטים על השורה והכסא. כמו כן, היא יכולה לספק פרטים נוספים בהתאם לסוג מקום הישיבה, ובכל מקרה היא גם מציינת את הסוג הספציפי של מקום הישיבה.

דוגמאות :

Green Room-> line: 2, chair: 10

Rear-> area: E, line: 60, chair: 18

הפעולה `price()` אמורה לספק מידע על מחיר מקום הישיבה, אבל בניגוד לפעולה `location()`

לא ניתן לתאר מראש בוודאות משהו שייעשה בפעולה זו בכל מקרה, כיון שיתכנו גם מקומות ישיבה שלגביהם המחיר לא יהיה כלל רלוונטי, ובמקרה זה הפעולה `price()` לא תנהג כרגיל אלא תאפשר לקבל חיווי מיוחד אודות כך.

ישנם מקומות ישיבה רבים וניתן לסווגם לפי סוגים שונים.

`MainHallSeat` זהו מקום שנמצא באולם הראשי. מחיר של מקום כזה כולל, בכל מקרה, את מחיר הבסיס ועליו ישנה תוספת של 100 ₪.

`RegularSeat` הוא מקום ישיבה שנמצא באולם הראשי (ולכן הוא, למעשה, סוג של `MainHallSeat`). בנוסף לשורה וכסא אשר מוגדרים לכל מקום ישיבה, הוא מגדיר גם אזור שמצוין ע"י תו א"ב גדול בתחום 'A'..'Z'.

`FrontRegularSeat`

`MiddleRegularSeat`

`RearRegularSeat`

הם 3 סוגי מקומות ישיבה ספציפיים מסוג `RegularSeat`.

`FrontRegularSeat` ממוקם בשורות קדמיות ולכן המחיר שלו כולל תוספת של 500 ₪.

`MiddleRegularSeat` ממוקם במרכז האולם והמחיר שלו כולל תוספת של 250 ₪.

`RearRegularSeat` ממוקם בחלק האחורי של האולם ואינו כולל שום תוספת במחיר.

לא כל מקומות הישיבה הממוקמים באולם הראשי מוגדרים בתוך `RegularSeat`.

ישנם מקומות באולם הראשי אשר מוגדרים `SpecialSeat`. מקומות כאלה אינם מצוינים ע"י

תו אזור מפני שמפאת ייחודם מוקצה להם מראש אזור אחד מיוחד ושמור.

`GoldenCircleSeat` הוא דוגמה למקום כזה. עקב ייחודו במיקום מאד מיוחד ליד הבמה, מחירו כולל תוספת נוספת מיוחדת של 1000 ₪. מקומות מסוג זה הם היקרים ביותר שמוצעים למכירה.

`DisablePodiumSeat` הוא מקום נוסף מסוג `SpecialSeat`. הוא מיועד לקהל בעל מוגבלויות

ולכן המחיר שלו הוא קבוע מראש – 200 ₪ בדיוק – ואינו תלוי כלל במחיר בסיס או תוספות אחרות כלשהן.

לא כל מקומות הישיבה ממוקמים באולם הראשי.

`GreenRoomSeat` הוא דוגמה למקום כזה. זהו מקום מיוחד אשר ממוקם ב- `Green Room`

ביחד עם המשלחות המתחרות. מפאת ייחודו של מקום כזה, לא מוגדר עבורו מחיר כלל והוא אינו מיועד למכירה לקהל הרחב. מקום כזה יינתן לאח"מים בלבד לפי הנחיות מחלקת יחסי ציבור.

פעולה `price()` אשר מופעלת בהקשר למקום כזה אינה מספקת מחיר (שלא קיים) אלא מאפשרת

להבין שמחיר אינו רלוונטי לגבי מקום כזה.

המשימות בחלק זה של התרגיל :

מצורף הקובץ `test_part_c.cpp`
קובץ זה מכיל דוגמה לתוכנית שמשתמשת במקומות ישיבה שונים כפי שפורטו לעיל.
הקובץ `test_part_c_out.txt` מכיל את הפלט המתקבל מהרצת התוכנית `test_part_c.cpp`.

- א. נתון הקובץ `seat.h` השלימו בו את הגדרות ה-classים השונים בהתאם למתואר לעיל ובהתאם למודגם בקבצים `test_part_c.cpp` ו-`test_part_c_out.txt`
הקפידו על הגדרה נכונה ומלאה של מערכת ירושה.
בנוסף הקפידו על כל הדיוקים הנדרשים בהקשר של `const`.
ב. כתבו בעצמכם את הקובץ `seat.cpp` אשר כולל את כל המימושים הנדרשים בתור השלמות להגדרות המופיעות בקובץ `seat.h`.
ג. קמפלו את התוכנית `test_part_c.cpp` ביחד עם הקוד שכתבתם וודאו שלא מתקבלות שגיאות ו/או אזהרות כלשהן.
הריצו את התוכנית וודאו שהפלט המתקבל זהה לפלט שמופיע בקובץ `test_part_c_out.txt`
ד. וודאו שכל הטסטים שמופיעים בהערה בקובץ `test_part_c.cpp` גם הם מתקיימים ועובדים.

הידור

צריך להדר ולהריץ את התוכנית על שרת ה- CSL3 .
ליצירת `test1.exe` (חלקים א+ב) , יש לבצע את הפקודה

```
g++ -o test1.exe -std=c++11 -Wall -pedantic-errors -Werror -DNDEBUG test1.cpp  
eurovision.cpp
```

ליצירת `test_part_c.exe` (חלק ג) , יש לבצע את הפקודה

```
g++ -o test_part_c.exe -std=c++11 -Wall -pedantic-errors -Werror -DNDEBUG test_part_c.cpp  
seat.cpp
```

שימו לב :

הקפידו על הדגלים השונים המופיעים בפקודה.
הקפידו לבדוק את התוכנית בשרת CSL3 ב – shell של bash.
הקפידו לעבוד עם גירסת gcc שאינה קטנה מ – 4.7.1 גירסה כזו אמורה להיות מותקנת בשרת CSL3. תוכלו לבדוק מהי הגירסה ע"י הפקודה `gcc --version`.

דגשים

הקפידו על ניהול תקין של זיכרון בכל מה שקשור להקצאות ולשחרורים.
בדקו את התוכנית שלכם עם valgrind. תוכלו למצוא פרטים באתר הקורס.
הקפידו על קוד איכותי ואמין. יש לבדוק הצלחה/אי הצלחה של הקצאות זיכרון.
יש להשתמש ב- assert במקרים רלוונטיים.
הקפידו על תיעוד סביר, שמאפשר לקרוא את הקוד בלי הפעלת מאמץ לא סביר כדי להבינו.
יישמו באופן רלוונטי את הנחיות הקונבנציות השונות, בהתאם לקובץ הקונבנציות כפי שמפורסם באתר הקורס.

בדיקות

הקפידו לבדוק באופן איכותי את כל יחידות התוכנית שכתבתם.
בתרגיל זה אין צורך להגיש בדיקות יחידה, אך באחריותכם לבצע בדיקות כאלה באופן מקיף ומלא. בפרט, כאמור, הקפידו לוודא שהקוד שלכם נותן תוצאות זהות לדוגמה שפורסמה, גם מבחינה לוגית וגם בפורמט.

הגשה

הגשה יבשה :

הדפסה של קובץ dry.docx אשר מכיל תשובות לחלק היבש של התרגיל תוגש לתא של הקורס, בניין טאוב קומה 1. הקפידו לרשום בדף הראשון בכתב ברור וקריא שם מלא, מס זהות מדויק ב-9 ספרות וכתובת מייל מדויקת (של 2 המגישים, אם מוגש על ידי 2 מגישים).

הגשה רטובה :

הגשה רטובה תבוצע באופן אלקטרוני באמצעות אתר הקורס, לפי מסלול הניווט באתר כדלקמן :
Assignments -> HW3 -> Electronic submission
יש להגיש קובץ zip (לא RAR וגם לא כל דבר אחר) הכולל :
• הקבצים eurovision.h eurovision.cpp
seat.h seat.cpp
• קובץ Word dry.docx שמכיל את כל מה שהגשתם ידנית בתור פתרון לחלק היבש של התרגיל.
• קובץ student.txt ממולא בפרטים האישיים שלכם. נתון קובץ ריק.
נא הקפידו הקפדה יתרה על מילוי מדויק של הפרטים (אנגלית בלבד) ובעיקר יש להקפיד על ת"ז ב-9 ספרות בדיוק וכן על כתובת mail מדויקת לחלוטין. **אסור לשנות את הכותרות הקיימות ואסור לשנות את מבנה השורות וגם לא להוסיף שורות או למחוק שורות.**

הקפידו, בבקשה, שבקובץ ה-zip אין שום תת תיקיות ושכל הקבצים המוגשים בתוך קובץ ה-zip נמצאים כולם באותה רמה, ללא תת תיקיות.

בהצלחה !