

08.08.2005

תכנות מונחה עצמים בשפת C++
אוניברסיטת תל אביב
בית הספר למדעי המחשב

תרגיל בית מספר 4

תאריך הגשה: 15.8.2005

שאלה 1

נרצה לשכלל את `myset<T>` מהתרגילים הקודמים, כך שתתמוך ב"סכום" כל אבריה. סכום זה יהיה מטיפוס `T` ויקבל משמעות לפי משמעות האופרטור `" += "` של טיפוס `T` השונים. המחלקה החדשה תקרא `mysetSum<T>`. המתודה `const T sum()` תחזיר את הסכום הנוכחי של הקבוצה.

לדוגמא:

- עבור `myset<int>` תחזיר `sum` את הסכום המספרי של אברי הקבוצה
- עבור `myset<string>` תחזיר `sum` שרשור של כל המחרוזות הנמצאות בקבוצה

דונו ב-3 אלטרנטיבות למימוש המחלקה (יתרונות וחסרונות. לא יותר מ-5 שורות):

- (1) האצלה של `myset<T>`
- (2) `subtyping` (ירושה `public`)
- (3) ירושת מימוש (ירושה `private` או `protected`)

וממשו את המחלקה באחת מהגישות לעיל לבחירתכם, נמקו בחירתכם. הנימוק: "כי זה הכי קצר" לגיטימית רק אם באמת בחרתם בדרך הקצרה ביותר...

שאלה 2

בשאלה זו עליך לתת Design (Prototypes) בלבד.

מנהל המרינה בעיר הנמל הציורית גוצ'ק ביקש ממך לממש אפליקציה לניהול כלי השייט העוגנים במרינה.

כלי השייט נחלקים ל:

- 1) סירות מרוץ
- 2) סירות משוטים
- 3) יאכטות

עבור כל כלי שייט יש לשמור את הפרטים הבאים:

- א) שם כלי השייט
- ב) אורך
- ג) רוחב
- ד) משקל
- ה) שנת יצור
- ו) שם החברה שיצרה את כלי השייט
- ז) החומר ממנו בנוי כלי השייט – יכול להיות: עץ, ברזל, פיברגלס או בטון

עבור סירות מרוץ יש לשמור גם מהירות מקסימלית. עבור סירות משוטים יש לשמור גם את מספר המשוטים. עבור יאכטות יש לשמור גם את מחירן. יאכטות נחלקות לשני סוגים: יאכטות מנועיות ויאכטות מפרש. עבור יאכטות מנועיות יש לשמור גם את עוצמת המנוע (בכוחות סוס), ועבור יאכטות מפרש יש לשמור את מספר התרנים, גובה תורן מקסימלי ושטח המפרשים הכולל. כמו כן, קיימות במרינה מספר יאכטות משולבות (דהיינו יאכטות מנועיות עם מפרשים). לכל כלי השייט, ממשק המאפשר לעדכן את פרטי כלי השייט.

כל כלי שייט צריך לעגון ברציף.

במרינה קיימים 100 רציפי עגינה קטנים (ממוספרים מ-1 עד 100), ו-26 רציפים גדולים (מסומנים ב-A עד Z). בכל רציף יכול לעגון לכל היותר כלי שייט אחד. ברציף קטן יכול לעגון כלי שייט שאורכו עד 20 מטר ובגודל עד 100 מטר. בחלק מהרציפים (בגלל רוח חזקה) אסור לעגון יאכטות עם מפרשים (יש לאפשר למנהל המרינה לעדכן נתון זה עבור כל רציף).

מערכת הניהול מאפשרת:

- 1) להוסיף ולמחוק כלי שייט מהרשימה. כאשר מוסיפים כלי שייט יש להקצות לו רציף מתאים לעגינה (או להודיע אם אין מקום מתאים).
- 2) לעדכן ולהדפיס פרטים של כלי שייט קיים.
- 3) להחליף את מקום העגינה בין שתי יאכטות (או להודיע אם אי אפשר).
- 4) להדפיס את רשימת כל כלי השייט, ממוינים לפי אורכם, ועבור כל כלי שייט לציין גם את רציף העגינה.
- 5) להדפיס את רשימת כל רציפי העגינה המלאים, ממוינים לפי מספרם (או סימנם – עבור רציפים גדולים), ועבור כל רציף לציין את שם כלי השייט שעוגן בו.
- 6) להדפיס את רשימת כל רציפי העגינה הפנויים (שוב, בצורה ממוינת).

בין השאר, עליך לתאר את הממשק (פונקציות) דרכו מבצעים פעולות אלו.

כתוב את תיאור המחלקות המתאימות: תרשימי Design וכן Prototypes של המתודות הרלבנטיות (Header files). אין צורך לממש בקוד מפורש אף פונקציה.

פרט ונמק בחירותיך.

שאלה 3

עליך להגדיר את המחלקות הבאות על מנת לתמוך בתכנית ה- main שלמטה

1. Expression – מחלקה המייצגת ביטוי כלשהו.
2. Literal – מחלקה המתארת מספר בודד.
3. BinaryOp – מחלקה המתארת פעולה בינארית.
4. TrinaryOp – מחלקה המתארת פעולה טרינארית.
5. Sum – מחלקה המתארת פעולה של סכום.
6. Product - מחלקה המתארת פעולה של מכפלה.
7. Exponent - מחלקה המתארת פעולה של חזקה.
8. CondExp – מחלקה המתארת פעולה על שלושה פרמטרים a, b, c שהיא $a ? b : c$.

כל אחת מן המחלקות הללו תכיל את הפונקציה הוירטואלית: **eval()** float.

הערות:

- כל הפעולות לעיל פועלות על ביטויים (Expression)
- שערך של ביטוי מתבצע ע"י הפונקציה eval
- ביטויים מורכבים ישוערכו ע"י הפעלה רקורסיבית של eval על הארגומנטים

הצג תרשימי Design המתארים את עץ ההורשה (מלבנים וחיצים).

```
int main()
{
    Literal *l1 = new literal(1.0);
    Literal *l2 = new literal(2.0);
    Literal *l3 = new literal(3.0);
    Sum *sum = new Sum(l1, l2);
    Expression *e1 = new Exponent(l3, sum);

    Product* prod = new Product(l1, l2);
    Exponent* exp = new Exponent(l2, l3);
    Expression *e2= new CondExp(sum, prod, exp);

    cout << *e1 << " = " << e1->eval();
    cout << *e2 << " = " << e2->eval();

    delete l1;
    delete l2;
    delete l3;
    delete sum;
    delete prod;
    delete exp;
    delete e1;
    delete e2;
}
```