

המחלקה להנדסת מערכות תקשורת

אוני' בן גוריון בנגב

הנחיות הגשה:

1. ההגשה מתבצעת בזוגות, ואין צורך להגיש פעמיים עבור שני בני הזוג(במידה ובוצעה הגשה כפולה, נא ציינו בהערה "לא לבדוק" על אחת מהן).
2. יש להגיש את כל קבצי העבודה במערכת ה VPL ולוודא שאין שגיאות קומפילציה במערכת.
3. כל מחלקה תכלול קובץ header וקובץ cpp (מלבד מחלקות Node ו LinkedList).
4. הקפידו על הוראות ברורות וקריאות למשתמש לפני כל פעולת קלט.
5. יש להקפיד של שמות המחלקות והשיטות כפי שמצוין בעבודה (הקפידו על אותיות גדולות וקטנות).
6. הקפידו על תכנות נכון, זהו חלק מן הציון:
 - a. שמות משתנים בעלי משמעות.
 - b. שימוש חוזר בקוד.
 - c. הקפדה על הזחות.
 - d. שימוש נרחב בהערות באנגלית בלבד. הקפידו לתעד כל שיטה וכל קטע קוד משמעותי.
7. בראש כל קובץ יש להוסיף בהערה את הטקסט הבא:

```
*/Assignment C++: 1
Author: Israel Israeli, ID: 01234567
/*
```

8. עליכם להחליף את הפרטים בפרטים שלכם.
9. כל שאלה יש להפנות לפורום המתאים לשאלות במודל.
10. הארכות יינתנו רק במקרי מילואים, אבל ומחלה חריפה ובצירוף אישורים מתאימים.

תרגיל מס' 3

נושא התרגיל: פולימורפיזם, RTTI, exceptions

הערה חשובה: רשימת המתודות היא חלקית בלבד. היא מהווה את המינימום הנדרש ממכם. ייתכן ואף סביר להניח שתצטרכו להוסיף מתודות נוספות עבור המחלקות המתוארות. לכל המחלקות חובה להוסיף Constructor ו- Destructor דיפולטיבים (אפילו אם הוא ריק).

**** שימו לב - בעבודה זו עליכם לוודא כי ערכי הקלט נמצאים בטווח המתאים. הטיפול יעשה באמצעות exceptions (אין צורך לבדוק טיפוסים), במידה והוזן קלט לא תקין יש להדפיס הודעה מתאימה ולאפשר למשתמש להזין את הנתונים בשנית.**

1. Templated LinkedList<T>

במשימה זו תממשו מחלקת תבנית של רשימה מקושרת. רשימה מקושרת היא מבנה נתונים שבו כל איבר הוא אובייקט נפרד (Node) אשר מכיל שדה של מידע ושדה של מצביע המצביע לאיבר הבא ברשימה.

Node:

Private members:

T data - The data stored in the node.

Node<T>* Next - A pointer to the next node in the linked list.

Public members:

Node(T data) - Constructor that takes a single T object as input and initializes the data field with this value. The next field should be initialized to nullptr.

LinkedList:

Private members:

Node<T>* head - A pointer to the first node in the linked list.

Public members:

LinkedList() - A default constructor that creates an empty linked list.

insert(const T&) - This method should create a new Node object with the given data and insert it at the front of the linked list (at the top of the list).

T deleteNode () - This method should delete the first Node in the list (the last inserted Node) and return the data member of this Node. In case the list is empty, the function throws an exception.

T deleteNode (const T& data) - This method should delete the first occurrence of a Node with the given data from the linked list and return the data member of this Node. If the Node is not found, the method should throw an exception.

search(const T&) - This method should search for the first occurrence of a Node with the given data in the linked list. If the Node is found, the method should return true. Otherwise, it should return false.

const T& getTop() const - Return the last inserted element (data) in the list. If the list is empty, the method should throw an exception.

operator<< - insertion operator should print the Nodes of the linked list.

2. Shape:

במשימה זו עליכם ליצור היררכיה של מחלקות אשר מייצגות צורות שונות.

Shape

- שדות:
 - string color – צבע הצורה, הצבעים האפשריים הם blue, red, green (ניתן לקרוא על ספריית string בלינק הבא: <https://cplusplus.com/reference/string/string>).
 - מתודות וירטואליות טהורות:
 - getPerimeter – מחזירה את ההיקף double.
 - getArea – מחזירה את השטח double.
 - בנאים:
 - בנאי המקבל את כל הפרמטרים ומאתחל בהתאם.
- ** במחלקה זו ניתן להגדיר מתודות נוספות **מלבד מתודת draw** שמוגדרת במחלקות ב Square ו OrthogonalTriangle.

Square

מחלקה זו יורשת מ Shape.

- שדות:
 - double – אורך צלע הריבוע.
- מתודות:
 - draw – מציגה למסך ריבוע של כוכביות בהתאם לאורך הצלע (עבור מספר ממשי נעגל כלפי מטה).
 - ממשו את המתודות הווירטואליות הטהורות שהוגדרו ב Shape.
- בנאים:
 - Square(string, double) – מקבל צבע ואורך צלע.

Circle

מחלקה זו יורשת מ Shape.

- שדות:
 - double – רדיוס.
- מתודות:
 - ממשו את המתודות הווירטואליות הטהורות שהוגדרו ב Shape.
- בנאים:
 - Circle(string, double) – מקבל צבע ואורך ורדיוס.

OrthogonalTriangle

מחלקה זו יורשת מ Shape.

המחלקה מתארת משולש ישר זווית ושווה-שוקיים.

- שדות:
 - double – אורך השוקיים.
- מתודות:
 - draw – מציגה למסך משולש ישר זווית של כוכביות בהתאם לאורכי השוקיים (עבר מספר ממשי נעגל כלפי מטה).
 - ממשו את המתודות הווירטואליות הטהורות שהוגדרו ב Shape.
- בנאים:
 - OrthogonalTriangle(string, double) – מקבל צבע ואורך הניצבים.

מחלקת תפריט – Menu

מחלקה זו תנהל את המערכת. על מחלקה זו לבצע פעולות קלט/פלט מהמשתמש. יש להציג את התפריט בלולאה עד שהמשתמש יבחר לסיים אותה.

שיטות:

mainMenu() – שיטה זו תדפיס את התפריט הבא:

המשתמש מקיש	שם הפעולה	הערות
1	הוספת צורה לרשימה	
2	הסרת צורה אחרונה שנכנסה לרשימה	
3	הדפסת פרטי הצורה האחרונה שנכנסה לרשימה	
4	יציאה מהתוכנית	יש לשים לב שמשחררים את כל ההקצאות הדינמיות שנעשו ב main

מתודה זו תיצור בתחילת הפעלתה אובייקט מסוג LinkedList של מצביעים ל Shape (LinkedList<Shape*>) ותציג את התפריט למשתמש.

התפריט צריך להראות כך:

```
=====
<1>Add element to the list
<2>Remove the last shape from list
<3>Print the last shape
<4>Exit
=====
```

1) הוספת צורה לרשימה. בבחירה זו המשתמש יתבקש לבחור את סוג הצורה ולהזין את הנתונים שלה. דוגמה להוספת צורה:

```
Choose 1 for Square, 2 for Circle, 3 for
Triangle: 1
Enter square's color: red
Enter square's side length: 5
```

2) הסרת הצורה האחרונה שהוכנסה.

3)הדפסת פרטי הצורה לפי אינדקס. דוגמה להדפסת פרטי צורה:

```
Square details:  
color=red, side length=5  
area= 25, perimeter=20  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

****שימו לב כי יש להפעיל את המתודה draw עבור משולש וריבוע.**

****שימו לב כי מתודת draw תוגדר אך ורק במחלקות Square ו OrthogonalTriangle (RTTI).**

4)יציאה מהתוכנית.

הערות:

1. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום נכניס את הטיפוס המתאים. אנחנו לא מתחייבים שהוא יהיה בטווח מסוים – אלא אם כן נאמר אחרת בשאלה עצמה.
2. אחרי כל הדפסה יש לבצע ירידת שורה.

יש להקפיד על תכנות נכון:

- a. כל הערכים שהם קבועים (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: `define`, `const`, או `enum` בהתאם לצורך.
 - b. יש לרשום הערות בשפע, ובאנגלית בלבד (לכל מחלקה מה התפקיד שלה, התוכנית מה היא מבצעת, כל פעולה לא טריוויאלית להסביר, וכל 2~3 שורות קוד – הערה, כל מתודה מה היא עושה).
 - c. יש להקפיד על כימוס נכון – כל השדות ומתודות השירות ב-`private`, הממשק ב-`public`, חלוקה לקבצים.
 - d. **יש להקפיד על הזחות, כיתוב נכון וקריא, ושמות משמעותיים.**
 - e. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על `reuse` בקוד. נקודות ירדו על קוד כפול !
 - f. לפני בקשת קלט, יש להדפיס למשתמש הוראה (איזה קלט מבוקש)
 - g. להזכירם: הנכם נדרשים לתכנת בשפת `C++` אי-לכך, כל שימוש בפונקציות וספריות של שפת `C` אסורה.
3. עבור הצלחת הבדיקות עליכם לוודא שהדפסות זהות לקובץ ההרצה שקיבלתם. (לא יורדו נקודות על שוני ברווחים ושורות).
4. בהצלחה!