

הנחיות הגשה לפרוייקט המתגלגל

1. העבודה על התרגילים היא בזוגות, ויש לכתוב את שמות המגישים בראש ה- main.
2. את התרגילים יש להגיש במודל בכניסה המתאימה שתפורסם להגשת התרגיל.
3. יש להגיש קובץ זיפ עם תיקיית הפרוייקט שנכתב ונבדק ב- Visual Studio.
4. לפני הגשת התרגיל יש לבדוק שהקלקה על קובץ הפרוייקט שאתם מגישים מצליחה לפתוח את הפרוייקט באופן תקין ב- Visual Studio ונוצר פרויקט חוקי שמתקמפל ורץ כיאות וללא צורך בצעדים נוספים (כדאי לבדוק ממיקום חדש ולא מהספרייה שבה פיתחתם ואפילו יותר טוב – על מחשב אחר מזה שבו פיתחתם).
5. בפתרון כל חלק יש להשתמש אך ורק בנושאים שנלמדו בכיתה ומוגדרים בראש החלק, **אין להשתמש בנושאים מתקדמים.**

הנחיות כלליות לכתיבת המערכת

1. שדות שהם **מחרוזות** יוקצו דינאמית ויוקצו גודל מדויק. בעת הקליטה ניתן להניח גודל מקסימלי אך הגודל המדויק הוא שישמר במחלקה.
2. הגדלה של **מערכים** (שהם לא מחרוזות) תהיה כל פעם פי 2. לכן יש צורך לתחזק משתנה לגודל הלוגי ולגודל הפיזי.
3. צריך "מחלקה מנהלת" שהיא המחלקה שמחזיקה את כל המידע ורק מולה ה- main יעבוד.
4. לא יהיה קלט במחלקות וגם לא במחלקה המנהלת. ה- main הוא זה שקולט את המידע ושולח ערכים כפרמטרים לפונקציות. באופן זה גם ניתן להכניס למערכת ערכים התחלתיים ב- main במקום להקלידם בכל הרצה מחדש.
5. בדיקות תקינות:
 - a. אין צורך להמציא הגבלות על הקלטים, אלא להתייחס רק להגבלות הנתונות בגוף השאלה. יחד עם זאת, להקפיד שמתודות ה- set תמיד יחזירו bool כתשתית לבדיקת תקינות עתידית.
 - b. ניתן להניח שהקלט המתקבל הוא מהטיפוס המבוקש, כלומר אם הקלט הצפוי הוא מספר, ניתן להניח שאכן הוקלד מספר. אך אם יש בטקסט הגבלות למשל לטווח ערכים אפשרי, אז יש לוודא שהמספר שהוקלד בטווח המתאים.
 - c. כל עוד לא נלמד הפרק של חריגות (exceptions), ניתן להניח שפרמטרים המועברים לקונסטרקטור תמיד תקינים, ואין צורך לטפל בקלט שגוי בקונסטרקטור. הסיבה לכך היא שבפרק של exception נראה דרך אלגנטית לטפל בערכים שגויים גם בקונסטרקטור.
6. אין צורך לבדוק הצלחה של הקצאות דינאמיות, ניתן להניח שכולן מצליחות.

התרגיל – ניהול בית חולים

תרגילים 1 עד 4 עוסקים בפרויקט מתמשך המפתח מערכת לניהול בית חולים.

שימו לב: ניתן להניח שהקלט תקין מבחינת הטיפוס שהוקלד (למשל, לא הוקלדה מחרוזת בקליטה המצפה לקבל מספר), אך לא ניתן להניח תקינות ערכים, ובכך עליכם לתמוך ברמת הקוד.

תיאור המערכת

בתרגיל זה ננהל מערכת בית חולים. בבית החולים יש מחלקות (למשל כירורגיה, פנימית, ילדים, לב וכו') כאשר לכל מחלקה יש אנשי צוות המשובצים אליה. איש צוות הוא רופא או אחות.

הנתונים שיש לשמור לרופא הם שמו, מספר עובד מזהה ותחום התמחות, ועבור אחות יש לשמור את שמה, מספר העובד המזהה שלה ומספר שנות הניסיון שלה. גם רופא וגם אחות צריכים לשמור את המחלקה בה הם עובדים.

כמו-כן בבית החולים ישנו מכון מחקר שיש לו שם ועובדים בו חוקרים. עבור כל חוקר, יש לשמור את שמו ואת מספר העובד שלו אוסף המאמרים שכתב. עבור כל מאמר יש לשמור את תאריך הפרסום שלו, שם המגזין בו הוא פורסם ושם המאמר.

חלק מהחוקרים הם גם רופאים.

לבית החולים מגיעים למחלקות השונות מטופלים. כאשר מטופל מגיע, נפתח עבורו כרטיס ביקור הכולל את המידע על המטופל (שם, ת.ז. שנת לידה ומין), תאריך ההגעה, מטרת הביקור ומחלקה אליה הוא משוייך. עבור כל ביקור יש לשמור מי איש בצוות הראשי המטפל במטופל.

תרגיל 1 – החומר המותר הינו עד מצגת קונסטרקטורים כולל

בחלק זה יש לכתוב תפריט אינטראקטיבי החוזר על עצמו, המאפשר את הפעולות הבאות:

1. הוספת מחלקה לבית החולים
2. הוספת אחות לבית החולים ושיבוצה למחלקה מסוימת
3. הוספת רופא לבית החולים ושיבוצו למחלקה מסוימת

4. הוספת ביקור: ראשית יש לשאול האם המטופל כבר ביקר בבית החולים בעבר ואם כן לבחור אותו מרשימת המטופלים הקיימים במערכת, ואם לא להזין עבורו נתונים. לאחר מכן לשאול לאיזו מחלקה הוא הגיע ואת הפרטים הדרושים להזנת הביקור
 5. הוספת חוקר למכון המחקר
 6. הוספת מאמר לחוקר
 7. הצגת כל המטופלים המשויכים למחלקה מסוימת
 8. הצגת כל אנשי הצוות הרפואי שבמערכת
 9. הצגת כל החוקרים שבמערכת
 10. חיפוש מטופל לפי ת.ז.: הצגת שמו והמחלקה בה הוא נמצא
 11. יציאה
- בנוסף, לפני התפריט, תכניסו ערכים hard-coded כדי שעם ההרצה המערכת לא תהייה ריקה, וניתן יהיה לעשות בדיקות בקלות.

תרגיל 2 – עד פרק 6 כולל (העמסת אופרטורים)

בחלק זה יש לעדכן את הקוד עם התוספות התחביריות והכלים החדשים שנלמדו בשפה, וכן לתמוך בהרחבת הסיפור:

1. הוספת תמיכה במספר סידורי ייחודי במערכת לכל הזמנה.

כמו כן, יש לתמוך באופרטורים הבאים:

1. $+=$ אשר מוסיף רופא למערכת, וגרסה המוסיפה אחות
2. יש לממש אופרטור $<<$ לפחות ל-3 מחלקות, לבחירתכם
3. הוסיפו אפשרות בתפריט לבדיקת כל אחד מהאופרטורים
4. יש לתמוך באופרטורים הבאים:
 - a. $+=$ אשר מוסיף איש צוות למחלקה מסוימת
 - b. $>$ אשר משווה בין שני חוקרים על-פי מספר המאמרים שלהם
 - c. יש לממש אופרטור $<<$ לפחות ל-3 מחלקות, לבחירתכם
 - d. הוסיפו אפשרות בתפריט לבדיקת כל אחד מהאופרטורים

תרגיל 3 – עד פולימורפיזם כולל

בחלק זה יש לעדכן את הקוד עם התוספות התחביריות והכלים החדשים שנלמדו בשפה (ובפרט הורשה ופולימורפיזם), וכן להוסיף למערכת תמיכה בדברים הבאים:

חלק מהרופאים הם רופאים מנתחים, ויש בנוסף לשמור עבורם את מספר הניתוחים שביצעו בתחום ההתמחות שלהם.

כאשר מטופל מגיע לבית החולים לביקור, הביקור יכול להיות או לצורך בדיקות או לצורך ניתוח, ובמקרה זה, יש לשמור גם את מספר חדר הניתוח בו יתנהל הניתוח והאם המטופל בצום.

לכן עדכנו את התפריט:

1. הרחבת האפשרות של הוספת רופא כך שהוא יהיה רופא מנתח, וכן יש לקלוט את הנתונים הרלוונטיים לכל סוג.
2. הרחבת האפשרות של הוספת ביקור כך שכל ביקור יהיה למטרת בדיקה או למטרת ניתוח, וכן יש לקלוט את הנתונים הרלוונטיים לכל סוג.
3. הצגת כל המנתחים
4. עדכון אופציית חיפוש המטופל, כך שהיא בנוסף תציג גם האם מטרת הביקור היא בדיקה או ניתוח

תרגיל 4 – STL

עדכנו את הקוד כך שיהיה שימוש במחלקה string במקום char* וכן שימוש ב-vector במקום במערכים.

שימו לב שיש הרבה קוד שיהיה עליכם למחוק.