

תרגיל בית מספר 3

נושא: סימולציה של כלי-שיט באמצעות MVC

תאריך הגשה: יום ה', 23:59, 28/06/2018

הגשה אפשרית בזוגות

בהצלחה רבה!

תיאור התרגיל

תרגיל זה עוסק בסימולציה תלוית-זמן של כלי-שיט, והיא מיועדת לעשות שימוש בפרדיגמת **Model-View-Controller**. אתם נדרשים לתת פתרון מונחה-עצמים בשפת C++ תחת אילוצים לשימוש בתבניות-תכן (*design patterns*). ראשית תינתן הצגה כללית של מרכיבי עולם הסימולציה ותיאור אחריות המחלקות השונות:

מרחב השיט והניווט

מרחב השיט מוגדר באמצעות מרחב אוקלידי דו-מימדי; יחידת המידה הינה מייל ימי (*Nautical Mile: nm*), וההנחה היא כי המרחב אינו סופי (אין אכיפה של תנאי שפה, מעבר למגבלות האחסון של הטיפוס *double*). צפון מוגדר להיות הכיוון החיובי של ציר y (0 מעלות), מזרח הוא הכיוון החיובי של ציר x (90 מעלות), דרום הינו הכיוון השלילי של ציר y (180 מעלות), ומערב הכיוון השלילי של ציר x (270 מעלות). מיקום או יעד של כלי שיט יוגדר באמצעות צמד קואורדינטות, אך כיוון שיט יכול להיות מוגדר באמצעות זווית.

נמל (port)

נמל הינו אובייקט סימולציה בעל מיקום קבוע ומאגרי דלק (ללא מגבלת אחסון), והוא נושא באחריות לפרוק, להעמס, ולתדלק כלי-שיט. לנמל קיימת יכולת עצמית של ייצור דלק, ואין לו מגבלה במלאי המכולות. קיומו נקבע בשלב איתחול הסימולציה.

ספינת-משא (freighter)

כלי-שיט זה מעביר מכולות (containers) מנמל לנמל, ע"פ פקודות העמסה ופריקה, ומשייט במהירות נתונה בין נקודה לנקודה בכפוף לצריכת דלק קבועה. בהגיעה לנמל המיועד לפריקה, ספינת המשא תפרוק את מספר המכולות שבפקודת הפריקה; בהגיעה לנמל המיועד להעמסה, היא תמלא את קיבולת המשא שלה למספר המקסימלי של מכולות ותשאף למלא את טנק הדלק (תלוי במלאי הדלק הקיים בנמל). יצירת הספינה מתרחשת בזמן הסימולציה.

ספינת-פטרול (patrol_boat)

כלי-שיט זה מפטרל בין הנמלים השונים במרחב הימי, עוגן בהם, ובהשלמת מסלול חוזר לנקודת היציאה. יצירת הספינה מתרחשת בזמן הסימולציה.

ספינת-פיראטים (cruiser)

כלי-שיט זה תוקף ושודד ספינות-משא או תוקף ספינות-פטרול; הוא איננו רשאי לעגון בנמלים. יצירת הספינה מתרחשת בזמן הסימולציה.

מודל (model)

זהו אובייקט יחיד אשר נדרש להיות מוגדר בתבנית Singleton; אחריותו לנהל מעקב אחר עולם הסימולציה על כל היבטיו, החל מניהול הזמן וכלה באחסון האובייקטים. בפרט, עליו להחזיק באמצעות מצביעים את כל כלי השיט והנמלים המשתתפים בסימולציה, ולספק שירותי גישה אליהם. בנוסף, האובייקט אחראי לספק שירותי עדכון לאובייקט התצוגה (view).

תצוגה (view)

אובייקט זה הינו בעל אחריות אחת והיא הצגת מפת העולם באמצעות גרפיקה מבוססת-ascii. כל אובייקט מיוצג במפה באמצעות שני התווים הראשונים של שמו. עדכון המידע מבוסס על אינטראקציה עם המודל.

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2018

יחידת-בקרה (controller)

זהו אובייקט יחיד שאחריותו לנהל את האינטראקציה מול המשתמש, ולנתב את הקלטים המתקבלים עבור המודל. אחריות זו כוללת גם את ניהול השגיאות בקלטי המשתמש.

איפיון מפורט של מרכיבי הסימולציה

מדובר בסימולציה בה צעדי הזמן נשלטים באמצעות פקודות בדידות, כאשר העולם "קופא" בכל צעד זמן, עד למעבר הזמן הבא. מעבר של צעד זמן מתרחש ע"פ פקודה של המשתמש, וכולל עדכון של כל העצמים בעולם הסימולציה.

כלי-שיט (כללי)

אובייקט כלי השיט הכללי יכול להימצא במצבים הבאים:

- מצב עצירה ("Stopped"); הוא מצבו ההתחלתי של כל כלי-שיט.
- מצב עגינה ("Docked") בנמל נתון
- מצב תקוע ("Dead in the water") במיקום נתון לאחר שאזל מלאי הדלק
- מצב תנועה אל מיקום / תנועה אל נמל / תנועה אל כיוון ("Moving to...") במהירות נתונה

כמו כן, הוא מצופה לתמוך בפונקציונליות הבאה:

- עצירה: פקודה זו תבטל את יעדיה של הספינה, תאפס את מהירותה, ותעדכן את מצבה ל-"Stopped".
- קביעת יעד שיט: פקודה זו תקבע את יעד השיט של הספינה באמצעות צמד קואורדינטות, תקבע את מהירות השיט, ותעדכן את מצבה ל - "Moving to <destination position>"
- קביעת כיוון שיט: פקודה זו תקבע את כיוון השיט של הספינה באמצעות זווית, תקבע את מהירות השיט, ותעדכן את מצבה ל-"Moving on course ...".
- הצגת סטטוס: הדפסת מצב כלי השיט הכוללת את שמו, מיקומו, ואת סטטוס השיט שלו (תלוי במצבו; אחד מהבאים):
"Moving to <destination position> on course ..."
"Moving to <port name> on course ..."
"Moving on course ..."
"Docked at <port name>"
"Stopped"
"Dead in the water"

ספינת-משא

בנוסף לפונקציונליות הנזכרת, ספינת המשא מצופה לתמוך בפונקציונליות הבאה:

- סטטוס קונקרטי: בנוסף לשמה, מיקומה וסטטוס השיט שלה, ספינת המשא תציין את תכולת הדלק שברשותה, כח עמידתה, מהירותה ואת היקף המשא אותו היא נושאת (ניתן להוסיף מידע טקסטואלי אחר). למשל,

```
Freighter Emma at (22.93, 22.93), fuel: 490.0 kl, resistance: 8, Moving to PearlHarbor on course 225.00 deg, speed 10.00 nm/hr, Containers: 0, moving to loading destination
```

- קביעת נמל יעד לפריקה או העמסה: פקודה זו תקבע את נמל היעד של הספינה באמצעות שם נמל חוקי, תקבע את מהירות השיט, ותעדכן את מצבה ל - "Moving to <port name>"
- עגינה: מצב עגינה בנמל יכול להתרחש כאשר כלי-השיט נמצא במרחק שאינו עולה על 0.1nm ממיקום הנמל; הפקודה תאפס את מהירותה של הספינה, תקבע את מיקומה למיקומו של הנמל, ותעדכן את מצבה ל-"Docked". הפעולה אורכת צעד זמן יחיד.
- פריקה: פעולת פריקה תתרחש במצב עגינה, ובמהלכה יועברו מספר נתון של מכולות מן הספינה אל מחסני הנמל; הפעולה אורכת צעד זמן יחיד. באם הפקודה דורשת מספר גדול מהמלאי, ייפרקו כל המכולות ותצא הודעת אזהרה.
- העמסה: פעולת העמסה תתרחש במצב עגינה, ובמהלכה **תתמלא קיבולת הספינה במכולות**; פעולה זו אורכת צעד זמן יחיד.
- תדלוק: פעולת התדלוק תתרחש במצב עגינה, ובמהלכה יתמלא טנק הדלק של הספינה בהתאם למלאי הדלק של הנמל, ועל בסיס תור הגינות ("first come first served"). הפעולה אורכת צעד זמן יחיד.

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2018

פרטי איפיון נוספים:

כל ספינות המשא הינן בעלות טנק דלק בנפח 500.0 אלפי ליטרים, מהירות מקסימלית של 40 nm/h, ותצרוכת דלק קבועה של 1000.0 ליטרים ל-nm. בעת יצירת ספינת משא, טנק הדלק שלה מלא. לספינת-משא קיים פרמטר הקובע את קיבולת המשא שלה ופרמטר המאפיין את יכולת העמידה שלה בפני מתקפות פיראטים (resistance).

ספינת פטרול

ספינת-פטרול משייטת בין הנמלים השונים באופן אוטומטי. בהינתן פקודת מסע לנמל מסויים (Destination), הספינה תפליג לנמל זה, משם תפליג לנמל הקרוב ביותר אליו, ממנו אל הנמל הקרוב ביותר שטרם ביקרה בו, וכך הלאה. במקרה של שוויון מרחקים בין נמלים קרובים שטרם ביקרה בהם, ייבחר הנמל הראשון לפי סדר אלפביתי. מטרתה לבקר בכל נמל פעם אחת בלבד, ולאחר שעשתה זאת תשלים את מסלולה בחזרה אוטומטית לנמל הראשון בו ביקרה ותסיים את השייט. כל ביקור של ספינת-פטרול בנמל נמשך שלושה צעדי-זמן: בראשון הספינה תנסה לתדלק בנמל, בשני היא תעגון ללא פעילות (בזמן שהשייטים יורדים ליבשה), ובשלישי היא תקבע את יעדה הבא. בסיום מסלול השייט (הנמל הראשון והאחרון), היא תבצע רק עגינה ותכריז על סיום השייט. כל ספינת-פטרול הינה בעלת טנק דלק בנפח 900.0 אלפי ליטרים, מהירות מקסימלית של 15 nm/h, ותצרוכת דלק קבועה של 2000.0 ליטרים ל-nm. בעת יצירת ספינת-פטרול, טנק הדלק שלה מלא. לספינת-פטרול קיים פרמטר הקובע את יכולת העמידה שלה בפני מתקפות פיראטים (resistance).

ספינת פיראטים

יכולת ספינת-פיראטים מוגדרות באמצעות כח-התקיפה שלה (ביחידות מקבילות ליכולת העמידה של ספינות המשא וספינות הפטרול) וטווח הפעולה שלה (ביחידות מרחק nm). פרמטרי הדלק של כלי-שיט זה אינם מסומלצים, ובפורמט תצוגת הסטטוס אין איזכור למיכל הדלק. מהירותה המקסימלית של ספינת הפיראטים היא 75 nm/h. בנוסף, אובייקט זה נדרש לתמוך בפעולות הבאות:

- סטטוס קונקרטי: בנוסף לשמה ומיקומה, ספינת הפיראטים תציין את כח-התקיפה שלה, כיוון ומהירות השייט שלה:

Cruiser Xerxes at (5.00, 25.00), force: 6, Moving on course 270.00 deg, speed 20.00 nm/hr

- תקיפה: פעולת התקיפה צריכה להיות מכוונת כלפי ספינת-משא או ספינת-פטרול אשר נמצאות בטווח התקיפה של ספינת הפיראטים בתחילת זמן הפעולה (צעד הזמן הבא בסימולציה), אחרת הפעולה תסתיים בכישלון. אם יעד התקיפה נמצא בטווח, הפעולה מתבצעת בצעד זמן יחיד ובמהלכה יושווה ערך כח התקיפה לערך כח העמידה של יעד התקיפה. **התקיפה תצליח רק אם בהשוואה מתקיים יחס גדול, או תיכשל אחרת (יחס קטן או שווה).**
 - במקרה של ספינת-משא, אם התקיפה צלחה, ספינת המשא תאבד את כל מטענה (מספר המכולות שלה יתאפס) וערך כח התקיפה של ספינת הפיראטים יגדל ביחידה. אם התקיפה כשלה, ספינת המשא תשמור על מצבה לפני התקיפה, ואילו ערך כח התקיפה של ספינת הפיראטים יקטן ביחידה.
 - במקרה של ספינת-פטרול – אם התקיפה צלחה, כח התקיפה של הפיראטים יגדל ביחידה וכח העמידה של ספינת הפטרול יקטן ביחידה; אם התקיפה כשלה, יתבצעו הגדלה/הקטנה הפוכים.
- בכל מקרה של תקיפה שיצאה אל הפועל, יבוטלו כל פקודות המסע של הספינה המותקפת, והיא תעבור למצב "Stopped".

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2018

איפיון המודל

המודל נדרש לאחסן את כל העצמים של עולם הסימולציה.

המודל נושא באחריות לעדכון מצבם של כל האלמנטים בעולם הסימולציה במעברי זמן בדידים של שעה אחת. כלומר, על המודל לבצע את כל החישובים והעדכונים המתרחשים בכל מעבר זמן, אשר קבוע להיות יחידת זמן של שעה אחת.
על בנאי המודל ליצור בכל מקרה נמל בשם Nagoya במיקום (5, 50), מלאי דלק התחלתי 1000000, ויכולת ייצור שעתית 1000.

איפיון יחידת הבקרה

יחידת הבקרה מנהלת את האינטראקציה מול המשתמש באמצעות קריאת פקודות, ובהתאם, מאפשרת שליטה במודל ובתצוגה. עיבוד טקסטואלי של פקודה מתבצע ע"פ הסדר הבא:

המילה הראשונה חייבת להיות המחרוזת `exit`, שם של כלי-שיט קיים, או פקודת מודל או תצוגה.

- באם המחרוזת הינה `exit`, על התכנית לסיים את הסימולציה ואת ריצתה בשלב זה.
- באם המחרוזת זהה לשם המלא של כלי-שיט, המילה שלאחריה נדרשת להיות פקודה חוקית שתעובד בהתאם.
- אחרת, המחרוזת נדרשת בהכרח להיות פקודה חוקית עבור המודל או התצוגה.

בכל הפקודות, תו ההפרדה הסטנדרטי בין הארגומנטים הינו רווח בודד.

פקודות יחידת הבקרה עבור התצוגה:

פקודה בשם `default` לשחזור פרמטרי ברירת המחדל של המפה (ללא ארגומנטים)

פקודה בשם `size` לקביעת גודל המפה (ארגומנט יחיד, שלם חיובי s , המגדיר את מספר העמודות והשורות: $30 \geq s > 6$)

פקודה בשם `zoom` לקביעת קנה-המידה של המפה (ארגומנט יחיד, ממשי חיובי, לקביעת היחס של מילי-ימי nm לתא שטח במפה)

פקודה בשם `pan` לקיבוע ראשית הצירים של המפה (צמד ארגומנטים ממשיים)

פקודה בשם `show` לבקשה מיחידת התצוגה לצייר את המפה המעודכנת

פקודות יחידת הבקרה עבור המודל:

פקודה בשם `status` בעקבותיה על כל האובייקטים בסימולציה לתאר את מצבם העדכני (ללא ארגומנטים)

פקודה בשם `go` לעדכון כל האובייקטים ביחידת זמן בדידה של שעה (ללא ארגומנטים)

פקודה בשם `create` ליצירת כלי-שיט חדש (5 או 6 ארגומנטים); באם כבר קיים כלי-שיט בשם זה, הדבר נחשב לשגיאה:

- שם כלי-השיט (מחרוזת אלפביתית עם לכל היותר 12 תווים)
- סוג כלי השיט (Freighter, Patrol_boat, Cruiser)
- צמד קואורדינטות לציון המיקום ההתחלתי (בסוגריים)
- מספר שלם לציון יכולת עמידה / יכולת תקיפה של כלי השיט.
- אופציונלי: מספר שלם: קיבולת מכולות עבור ספינת המשא או טווח התקיפה עבור ספינת הפיראטים.

(הפקודות הבאות יינתנו לאחר שם כלי-שיט):

פקודה בשם `course` לקביעת כיוון השיט עבור כלי נתון (2 ארגומנטים: זווית במעלות ומהירות השיט)

פקודה בשם `position` לקביעת יעד השיט עבור כלי נתון (3 ארגומנטים: צמד קואורדינטות ומהירות השיט)

פקודה בשם `destination` לקביעת נמל היעד (2 ארגומנטים: שם נמל חוקי ולאחריו מהירות השיט)

פקודה בשם `load_at` לבחירת נמל יעד לטעינת מכולות (ארגומנט יחיד: שם נמל חוקי)

פקודה בשם `unload_at` לבחירת נמל יעד לפריקת מכולות (שני ארגומנטים: שם נמל חוקי ומספר מכולות לפריקה)

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2018

פקודה בשם `dock_at` לבחירת נמל לעגינה (ארגומנט יחיד: שם נמל חוקי)

פקודה בשם `attack` לקביעת כלי-שיט לתקיפה בצעד הזמן הבא (ארגומנט יחיד: שם כלי-שיט חוקי)

פקודה בשם `refuel` לביצוע בקשת תדלוק מהנמל (ללא ארגומנטים)

פקודה בשם `stop` לעצירת השיוט (ללא ארגומנטים); **פקודה זו תגרום לביטול כל פקודות המסע העתידיות**, באם קיימות

איפיון התצוגה

אובייקט התצוגה מספק פלט מרחבי בתצורת מפה עליו פרושים איברי הסימולציה. התצוגה מורכבת ממטריצה ריבועית, אשר בברירת המחדל ובתחילת הסימולציה מאותחלת להיות בגודל `25X25`; (גודלה לפחות 6, ולכל היותר 30).

תא מייצג טווח ערכים של קואורדינטות מרחביות (x,y) , כאשר גודלו של כל תא מייצג את קנה-המידה; ברירת המחדל היא 2.0. ראשית הצירים, אשר יכולה להיות כל נקודה מרחבית, מוגדרת להיות הקואורדינטה בפינה השמאלית התחתונה של המפה. מבחינת ייצוג, כל תא במפה מיוצג ע"י שני תווים, כאשר תא ריק מיוצג ע"י נקודה `'.'` ולאחר מכן רווח `' '` :
:

פורמט קלט

התכנית תקבל כקלט קובץ המתאר את הנמלים במרחב השיט. לאחר קומפילציה של התכנית לכדי קובץ הרצה בשם `simNautica`, כך ניתן יהיה להריץ אותה במקרה של עבודה דרך טרמינל:

```
>> ./simNautica portus.dat
```

1. הנמלים במרחב השיט, אשר יוזנו לתכנית בתחילתה בקובץ הקלט (`portus.dat` בדוגמא שלעיל), יתוארו באמצעות שמם, מיקומם הגיאוגרפי (באמצעות צמד קואורדינטות בסוגריים), מלאי הדלק ההתחלתי, ויכולת ייצור הדלק לשעה. להלן דוגמא לתוכן של קובץ קלט תקין:

```
Valdez (20.00, 20.00) 100000 100
PearlHarbor (10.00, 10.00) 500000 500
Yokohama (0.00, 30.00) 100000 0
```

2. במקרה של קלט לא חוקי (למשל, שגיאת קבצים, הזנת תווים/שדות לא חוקיים, וכיוצא באלה), התכנית תסיים את ריצתה בשלב זה ותדפיס הודעת שגיאה מתאימה לערוץ השגיאות הסטנדרטי.
3. בהנחה שהקלט של המשתמש חוקי, על התכנית לפתוח ערוץ קלט סטנדרטי מול המשתמש לשם קליטת פקודות.

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2018

קונסול לקליטת פקודות

היות ומדובר בסימולציה עם מעברי זמן בדידים ונשלטים, על התכנית להדפיס בכל שלב את צעד הזמן הנוכחי. הפניה אל המשתמש בזמן X לקבלת פקודה תתרחש בפורמט הבא:

Time X: Enter command:

ניתן להזין מספר רב של פקודות controller בכל צעד זמן, ורק הפקודה go תביא להתקדמות הזמן (סיום צעד הזמן הנוכחי) תוך חישוב כל העדכונים הנדרשים.

להלן רצף חוקי של פקודות קונסול לאחר איתחול התכנית באמצעות קובץ הדוגמא :portus.dat

```
Time 0: Enter command: create Ajax Cruiser (15.00, 15.00) 6 1
Time 0: Enter command: create Emma Freighter (30.00, 30.00) 8 1000
Time 0: Enter command: create Xerxes Cruiser (25.00, 25.00) 6 2
Time 0: Enter command: create Colombo Freighter (-5.0, 20.0) 8 1000
Time 0: Enter command: create Shikishima Patrol_boat (0.0, 0.0) 10
Time 0: Enter command: Colombo destination Yokohama 10
Time 0: Enter command: Emma load_at PearlHarbor
Time 0: Enter command: Emma unload_at Valdez 100
Time 0: Enter command: Emma destination PearlHarbor 10
Time 0: Enter command: Ajax course 180 20
Time 0: Enter command: Xerxes course 270 20
Time 0: Enter command: Shikishima destination Nagoya 10
Time 0: Enter command: go
Time 1: Enter command: status

Port Nagoya at position (50.00, 5.00), Fuel available: 1001000.0 kl
Port Valdez at position (20.00, 20.00), Fuel available: 100100.0 kl
Port PearlHarbor at position (10.00, 10.00), Fuel available: 500500.0 kl
Port Yokohama at position (0.00, 30.00), Fuel available: 1000.0 kl

Freighter Colombo at (-0.53, 28.94), fuel: 490.00 kl, resistance: 8, Moving to Yokohama
on course 26.57 deg, speed 10.00 nm/hr, Containers: 0, no cargo destinations

Freighter Emma at (22.93, 22.93), fuel: 490.0 kl, resistance: 8, Moving to PearlHarbor on
course 225.00 deg, speed 10.00 nm/hr, Containers: 0, moving to loading destination

Cruiser Ajax at (15.00, -5.00), force: 6, Moving on course 180.00 deg, speed 20.00 nm/hr
Cruiser Xerxes at (5.00, 25.00), force: 6, Moving on course 270.00 deg, speed 20.00 nm/hr

Patrol_boat Shikishima at (9.95, 1.00), fuel: 880.00 kl, resistance: 10, Moving to Nagoya
on course 84.29 deg, speed 10.00 nm/hr
```

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2018

קובץ טקסטואלי של פקודות אלו, הכולל הצגת מפה באמצעות `show`, נתון לכם כנספח במודל בשם `console.dat`. בנוסף, מצורף קובץ טקסטואלי המדגים את פקודות התצוגה עבור אותה סימולציה בזמן 0 - כנספח במודל בשם `view_console.dat`.

הנחות

- שם נמל הוא מחרוזת אלפביתית עם 12 תווים לכל היותר.
- יחידת הבסיס למדידת דלק היא אלף ליטרים (kl).
- תנועת כלי השיט תמיד תתרחש בקו ישר בין נקודות מקור ליעד.
- נניח כי אין רוחות או זרמים המתנגדים למהירות השיט הנתונה.
- אלמנטים במרחב השיט הינם בעלי גודל אפס, ולפיכך אין להתחשב בתסריט של התנגשות בין עצמים סמוכים במרחב.

אילוצים ודגשים

- עליכם להסתמך על מודול גיאומטריה של המרחב הימי ועל שלד אובייקט יחידת-הבקרה, המסופקים לכם באמצעות קבצי מקור.
- תבניות-תכן: אובייקט המודל נדרש להיות סינגלטון בעל גישה גלובלית; יצירת כלי-השיט נדרשת להסתמך על תבנית *factory* כלשהי.
- המימוש נדרש להיות בשפת C++, עם מתן עדיפות לתחביר התקן החדש.
- בתרגיל זה עליכם להשתמש במצביעים החכמים של C++11/14 בכל שימוש במצביעים.
- יש לבדוק תקינות קלטים; במקרה של אי-תקינות, יש להפעיל מנגנון חריגות ולהציג הודעות שגיהא מתאימות.
- עליכם לשאוף ליעילות ביצועים, ובפרט, לממש *move semantics* עבור מחלקות שעשויות להפיק מכך תועלת.
- עליכם לוודא כי התכנית עוברת קומפילציית `g++ -std=c++11` התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המחלקות והפונקציות השונות.

הגשה

- הכינו קובץ בשם `README.txt` הכולל את שם ותעודת הזהות של הסטודנטים/ויות המגישים/ות; בקובץ זה אתם מוזמנים לכלול הערות ותיעוד כללי לגבי המימוש שלכם.
- עליכם להגיש במערכת Moodle קובץ ארכיב בשם `ex3.xxx` המכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, וכן את הקובץ `Readme.txt`.

אי-הקפדה על ההנחיות תגרור הורדה בציון התרגיל. לא תתקבלנה הגשות באיחור!

Bibliography

1. D. Kieras, EECS 381, University of Michigan, 2017
2. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
3. B. Stroustrup, The C++ Programming Language, 4th ed., Addison-Wesley, 2013