

תרגיל בית מספר 3

נושא: סימולציית ימי-ביניים באמצעות MVC

תאריך הגשה: מוצ"ש, 01/07/2023, 23:59

הגשה אפשרית בצמדים

בהצלחה רבה!

תיאור התרגיל

תרגיל זה עוסק בסימולציה תלוית-זמן של תקופת ימי הביניים, והיא מיועדת לעשות שימוש בפרדיגמת **Model-View-Controller**. אתם נדרשים לתת פתרון מונחה-עצמים בשפת C++ ולעשות שימוש בתבנית-תכן מסוימת (*The Singleton design patterns*). ראשית תינתן הצגה כללית של מרכיבי עולם הסימולציה ותיאור אחריות המחלקות השונות:

המרחב הגיאוגרפי

המרחב הגיאוגרפי מוגדר באמצעות מרחב אוקלידי דו-מימדי; יחידת המידה הינה 10 ק"מ ($10km$), וההנחה היא כי המרחב אינו סופי (אין אכיפה של תנאי שפה, מעבר למגבלות האחסון של הטיפוס float). 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 אובייקט יוגדר באמצעות צמד קואורדינטות, אך כיוון תנועה יכול להיות מוגדר באמצעות זווית.

חוה (Farm)

חוה הינה אובייקט סימולציה בעל מיקום קבוע ותכולה של ארגזי יבול חקלאי (ללא מגבלת אחסון); לכל חווה יש כושר ייצור של יבול כתלות בזמן, והיבול מועבר באמצעות ארגזים אל המצודות. קיום החווה נקבע בשלב איתחול הסימולציה.

מצודה (Castle)

מצודה הינה אובייקט סימולציה בעל מיקום קבוע ותכולה של ארגזי יבול חקלאי (ללא מגבלת אחסון); היבול המאוחסן מתקבל רק באמצעות פריקה פרטנית של ארגזי איכרים. קיום המצודה נקבע בשלב איתחול הסימולציה.

איכר (peasant)

סוכן זה מעביר ארגזי יבול מחוות למצודות ע"פ פקודות עבודה. בהגיעו למצודת היעד, האיכר יפרוק את כל היבול שהביא ויחזור לחווה ממנה הגיע. האיכר נוצר במהלך הסימולציה, שם גם נקבעים המהלכים שלו.

אביר (knight)

סוכן זה מפטרל בין החוות והמצודות השונות במרחב הגיאוגרפי, ובהשלמת מסלול חוזר לנקודת היציאה. האביר נוצר במהלך הסימולציה, שם גם נקבעים המהלכים שלו.

ביריון (thug)

סוכן זה תוקף ושודד איכרים. הביריון נוצר במהלך הסימולציה, שם גם נקבעים המהלכים שלו.

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 הסוכנים והאתרים 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 עולם ימי-הביניים באמצעות גרפיקה מבוססת-ascii. כל אובייקט מיוצג במפה באמצעות שני התווים הראשונים של שמו. עדכון המידע מבוסס על אינטראקציה עם המודל.

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.

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

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, מייצג שעה עגולה אחת, וכולל עדכון של כל העצמים בעולם הסימולציה.

סוכן (כללי)

אובייקט סוכן כללי (agent) יכול להימצא במצבים הבאים:

- מצב עצירה ("stopped"); הוא מצבו ההתחלתי של כל סוכן.
- מצב מת ("dead") במיקום נתון (לאחר שהותקף)
- מצב תנועה אל מיקום / תנועה אל אתר / תנועה אל כיוון ("Moving to...")

כמו כן, הוא מצופה לתמוך בפונקציונליות הבאה:

- עצירה: פקודה זו תבטל את פעולותיו/יעדיו של הסוכן ותעדכן את מצבו ל-"Stopped".
- קביעת יעד: פקודה זו תקבע את יעד הרכיבה של הסוכן באמצעות צמד קואורדינטות ותעדכן את מצבו ל - "Moving to <destination position>"
- קביעת כיוון: פקודה זו תקבע את כיוון הרכיבה של הסוכן באמצעות זווית ותעדכן את מצבו ל- "Moving on course ...".
- הצגת סטטוס: הדפסת מצב הסוכן הכוללת את שמו, מיקומו, ואת סטטוס הרכיבה שלו (ראו דוגמאות מפורשות בהמשך).

איכר

האיכר מקבל פקודות עבודה הכוללות מקור (אתר חווה) ויעד (אתר מצודה). בהינתן פקודת עבודה חוקית (`start_working`), האיכר יצעד אל החווה, יעמיס ארגזים במשך שעה אחת, יצעד משם אל המצודה, ויפרוק את הארגזים הללו במשך שעה אחת. באם לא ניתנה לו פקודה נוספת, הוא יישאר במצודת היעד בתום פקודת העבודה.

סוכן זה מאופיין ע"י מצב בריאותו (כל האיכרים מאותחלים עם 10 יחידות כח; הערך המקסימלי הוא 20 יחידות כח). כל איכר צועד במהירות קבועה של 5 km/h. האיכר יכול לשאת לכל היותר 5 ארגזים בפקודת עבודה; באם יש בחווה פחות מ-5 ארגזים, האיכר ייקח את כל יתרת הארגזים. השלמת פקודת עבודה באופן מלא תביא לתוספת של יחידת כח אחת לבריאותו של האיכר.

בנוסף לפונקציונליות הנזכרת, האיכר מצופה לתמוך בפונקציונליות הבאה:

- יצירת איכר חדש באמצעות הפרמטרים הבאים:
 - שם האיכר (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - צמד קואורדינטות לציון המיקום ההתחלתי
- סטטוס קונקרטי: בנוסף לשמו, מיקומו וסטטוס הפעולה שלו, האיכר יציין את מספר הארגזים שהוא נושא ואת מצב בריאותו (ניתן להוסיף מידע טקסטואלי אחר). למשל,
- פקודה בשם `start_working` לקביעת עבודה (ארגומנטים: שני אתרים – שמות חווה ומצודה חוקיים).
- רצף של פקודות עבודה חוקיות יתורגם לביצוע הפקודות ברצף זו אחר זו (כלומר, מתן פקודה במהלך ביצוע פקודה אחרת).
- [האיכר אינו תומך בפקודות התנועה של הסוכן הכללי (`course/position`)].

אביר

האביר מפטרל על סוסו בין האתרים השונים באופן אוטומטי. בהינתן פקודת רכיבה לאתר מסוים (`destination`), האביר ירכב לאתר זה, משם ימשיך לאתר הקרוב ביותר אליו, ממנו אל האתר הקרוב ביותר שטרם ביקר בו, וכך הלאה. במקרה של שוויון מרחקים בין אתרים קרובים שטרם ביקר בהם, ייבחר האתר הראשון לפי סדר אלפביתי. מטרת האביר לביקר בכל אתר פעם אחת בלבד, ולאחר שעשה זאת ישלים את מסלולו בחזרה אוטומטית לאתר הראשון בו ביקר ויסיים את הפטרול. האביר יכול גם לקבל פקודות רכיבה פרטניות של סוכן כללי (`course/position`).

כל ביקור של אביר באתר אינו אורך זמן. כל אביר רוכב במהירות קבועה של 10 km/h, והוא מצופה לתמוך בפונקציונליות הבאה:

- יצירת אביר חדש באמצעות שני הפרמטרים הבאים:
 - שם האביר (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - שם אתר לציון המיקום ההתחלתי
- פקודה בשם `destination` לקביעת אתר היעד (ארגומנט: שם אתר חוקי)
- פקודות בשם `course` לקביעת כיוון הרכיבה (ארגומנט: זווית במעלות) וכן `position` לקביעת יעד הרכיבה (ארגומנטים: צמד קואורדינטות).

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

ביריון

הביריון הוא פרא-אדם המסתובב במרחב הגיאוגרפי ושוודד למחייתו איכרים המעבירים יבול. סוכן זה מאופיין ע"י כח התקיפה שלו (איתחול כל הביריונים הוא 5 יחידות כח; הערך המקסימלי האפשרי הינו 20 יחידות כח). הביריון רוכב על קרנף (*rhinoceros*) במהירות משתנה, עם ערך מקסימלי של 30 km/h. אובייקט זה נדרש לתמוך בפעולות הבאות:

- יצירת ביריון חדש באמצעות הפרמטרים הבאים:
 - שם הביריון (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - צמד קואורדינטות לציון המיקום ההתחלתי
- פקודות בשם **course** לקביעת כיוון (ארגומנטים: זווית במעלות ומהירות הרכיבה) וכן **position** לקביעת יעד הרכיבה (ארגומנטים: צמד קואורדינטות ומהירות הרכיבה).
- תקיפה: פעולת התקיפה אמורה להיות מכוונת כלפי איכר אשר נמצא ברדיוס ק"מ אחד מהביריון בתחילת זמן הפעולה (צעד הזמן הבא בסימולציה), אחרת הפעולה תסתיים בכישלון. אם יעד התקיפה נמצא בטווח, הפעולה מתבצעת מיידית (כלומר בצעד הזמן הנוכחי, שהוא בהכרח בשעה עגולה), ובמהלכה נבדקת נוכחות של אבירים בקרבת האיכר המותקף.

התקיפה תצליח רק אם מתקיימים בו-זמנית שלושה תנאים בזמן ובמיקום התקיפה:

*** הביריון נמצא בתוך רדיוס של 1 ק"מ מהאיכר**

*** כח הביריון עולה על כח האיכר (מספר יחידות כח הביריון גדול ממספר יחידות בריאות האיכר)**

*** לא קיים אביר ברדיוס של 2.5 ק"מ ממיקום האיכר**

- במקרה של איכר, כל מתקפה עליו תביא לאיבוד של יחידת בריאות אחת ממצב בריאותו. אם התקיפה צלחה, האיכר יאבד את כל יבולו (מספר הארגזים שלו יתאפס), פקודת העבודה שלו תבוטל מיידית, והוא יעבור למצב "Stopped". אם התקיפה כשלה, האיכר ימשיך בהשלמת הפקודה שלו כמתוכנן.
- במקרה של ביריון – בין אם התקיפה צלחה או לא, הביריון יעבור למצב "Stopped" במיקום השוד. אם התקיפה צלחה, כוחו של הביריון יגדל ביחידה אחת; אם התקיפה כשלה, כוחו יקטן ביחידה אחת.
- עבור שני הסוכנים המעורבים בתקיפה, באם מצב כוחו של סוכן מתאפס בעקבותיה, הוא יעבור למצב "Dead".
- בכל מקרה, האביר אינו משתתף באינטראקציה עצמה במידה והוא מצוי באיזור.

איפיון המודל

המודל נדרש לאחסן את כל העצמים של עולם הסימולציה.

המודל נושא באחריות לעדכון מצבם של כל האלמנטים בעולם הסימולציה במעברי זמן בידידים של שעה אחת. כלומר, על המודל לבצע את כל החישובים והעדכונים המתרחשים בכל מעבר זמן, אשר קבוע להיות יחידת זמן של שעה אחת.

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 סוכן 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 סוכן, 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 10 ק"מ לתא שטח במפה)
- פקודה בשם **pan** לקיבוע ראשית הצירים של המפה (צמד ארגומנטים ממשיים)
- פקודה בשם **show** לבקשה מיחידת התצוגה לצייר את המפה המעודכנת

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

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 סוכן חדש (איכר, אביר או בריון, 3 או 4 ארגומנטים בהתאם לטיפוס); באם כבר קיים סוכן בשם זה, הדבר נחשב לשגיאה:

- שם הסוכן (מחרוזת אלפביתית עם לכל היותר 16 תווים)
- סוג הסוכן: Peasant / Knight / Thug
- ציון המיקום ההתחלתי: שם אתר חוקי עבור אביר, צמד קואורדינטות עבור איכר/בריון (בסוגריים, מופרדים ביניהם בפסיק)

(הפקודות הבאות יינתנו לאחר שם סוכן):

- פקודה בשם `course` לקביעת כיוון הרכיבה עבור סוכן נתון (ארגומנט: זווית במעלות; במקרה של בריון, לאחריה תימסר מהירות הרכיבה)
- פקודה בשם `position` לקביעת יעד הרכיבה עבור סוכן נתון (ארגומנטים: צמד קואורדינטות; במקרה של בריון, לאחריהם תימסר מהירות הרכיבה)
- פקודה בשם `destination` לקביעת אתר היעד עבור האביר (ארגומנט: שם אתר חוקי)
- פקודה בשם `stop` לעצירת הרכיבה (ללא ארגומנטים)
- פקודה בשם `attack` עבור הבריון לקביעת איכר לתקיפה בצעד הזמן הבא (ארגומנט יחיד: שם איכר חוקי)

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 בפניה השמאלית התחתונה של המפה. מבחינת ייצוג, כל תא במפה מיוצג ע"י שני תווים, כאשר תא ריק מיוצג ע"י נקודה ' '. ולאחר מכן רווח ' ' .

שימו לב כי יחידת המידה המרחבית היא 10 ק"מ, ויחידת מידת מהירות התנועה של הסוכנים היא קמ"ש.

פורמט קלט

התכנית תקבל כקלט בדיוק שני קבצים לתיאור האתרים במרחב הגיאוגרפי: קובץ המתאר את המצודות ולאחריו קובץ המתאר את החוות. לאחר קומפילציה של התכנית לכדי קובץ הרצה בשם `simMedieval`, כך ניתן יהיה להריץ אותה במקרה של עבודה דרך טרמינל:

```
$ ./simMedieval castles.dat farms.dat
```

1. במקרה של קובץ המצודות, כל שורה תתאר מצודה באמצעות שמה, מיקומם הגיאוגרפי (באמצעות צמד קואורדינטות בסוגריים), וכמות ארגזי היבול ההתחלית שבה. להלן דוגמה לתוכן של קובץ מצודות תקין:

```
Rivendale, (20.00, 10.00), 100  
Blackness, (10.00, 20.00), 20  
Sunnybrook, (30.00, 0.00), 50
```

2. במקרה של קובץ החוות, כל שורה תתאר חווה באמצעות שמה, מיקומם הגיאוגרפי (באמצעות צמד קואורדינטות בסוגריים), כמות ארגזי היבול ההתחלית שבה, ויכולת הייצור השעתית שלה. להלן דוגמה לתוכן של קובץ קלט תקין:

```
Campbell, (20.00, 20.00), 10, 4  
Shire, (10.00, 10.00), 2, 3
```

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

Dunvegan, (0.00, 30.00), 6, 3

3. 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.
4. בהנחה שהקלט של המשתמש חוקי, על התכנית לפתוח ערוץ קלט סטנדרטי מול המשתמש לשם קליטת פקודות.

קונסול לקליטת פקודות

היות ומדובר בסימולציה עם מעברי זמן בדידים ונשלטים, על התכנית להדפיס בכל שלב את צעד הזמן הנוכחי. הפנייה 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 בשעת המעבר. אם יש סתירה בין פקודות לסוכן מסויים, הפקודה האחרונה דורסת את קודמותיה. להלן רצף חוקי של פקודות קונסול לאחר איתחול התכנית באמצעות קבצי הדוגמא : castles.dat farms.dat

```
Time 0: Enter command: create Bug Peasant (0.00, 0.00)
Time 0: Enter command: create Iriel Peasant (0.00, 0.00)
Time 0: Enter command: create Merry Peasant (0.00, 0.00)
Time 0: Enter command: create Pippin Peasant (0.00, 0.00)
Time 0: Enter command: create Zug Peasant (0.00, 0.00)
Time 0: Enter command: create Loki Thug (14.00, 14.00)
Time 0: Enter command: create Gog Thug (24.00, 24.00)
Time 0: Enter command: create Wallace Knight Rivendale
Time 0: Enter command: Loki course 180 30
Time 0: Enter command: Gog course 270 30
Time 0: Enter command: Wallace destination Sunnybrook
Time 0: Enter command: go
Time 1: Enter command: status
Castle Rivendale at position (20.00, 10.00), Inventory: 100
Castle Blackness at position (10.00, 20.00), Inventory: 20
Castle Sunnybrook at position (30.00, 0.00), Inventory: 50
Farm Campbell at position (20.00, 20.00), Inventory: 14
Farm Shire at position (10.00, 10.00), Inventory: 5
Farm Dunvegan at position (0.00, 30.00), Inventory: 9
Peasant Bug at (0.00, 0.00), Stopped
Peasant Iriel at (0.00, 0.00), Stopped
Peasant Merry at (0.00, 0.00), Stopped
Peasant Pippin at (0.00, 0.00), Stopped
Peasant Zug at (0.00, 0.00), Stopped
```

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

Thug Loki at (14.00, 11.00), Heading on course 180.00 deg, speed 30.00 km/h

Thug Gog at (21.00, 24.00), Heading on course 270.00 deg, speed 30.00 km/h

Knight Wallace at (20.71, 9.29), Heading to Sunnybrook, speed 10.00 km/h

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

הנחות

- שם אתר הוא מחרוזת אלפביתית עם 16 תווים לכל היותר.
- תנועת הסוכנים תמיד תתרחש בקו ישר בין נקודות מקור ליעד.
- האלמנטים בסימולציה הינם בעלי גודל אפס, ולפיכך אין להתחשב בתסריט של התנגשות בין עצמים סמוכים במרחב.

אילוצים ודגשים

- תבניות-תכן: אובייקט המודל נדרש להיות סינגלטון בעל גישה גלובלית.
- המימוש נדרש להיות בשפת C++, עם מתן עדיפות לתחביר התקן החדש.
- בתרגיל זה עליכם להשתמש במצביעים החכמים של C++0x בכל שימוש במצביעים.
- יש לבדוק תקינות קלטים; במקרה של אי-תקינות, יש להפעיל מנגנון חריגות ולהציג הודעות שגיאה מתאימות.
- עליכם לשאוף ליעילות ביצועים, ובפרט, לממש *move semantics* עבור מחלקות שעשויות להפיק מכך תועלת.
- עליכם לוודא כי התכנית עוברת קומפילציית `g++ -std=c++0x` 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. בנוסף, צרפו קובץ `README` ובו כיתבו את אופן הקומפילציה וההרצה.

הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('49'), שם התרגיל ('ex3') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי בפורמט הבא: `49_ex3_studID.zip`, או בפורמט משורשר `49_ex3_ID1_ID2.zip` במקרה של הגשת צמד.
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות `*.txt`, `*.h`, `*.cpp`.
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם `49_ex3_012345678.zip` הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

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

Bibliography

1. D. Kieras, EECS 381, University of Michigan, 2019
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