

# מכללת הדסה, החוג למדעי המחשב

## תכנות מונחה עצמים ופיתוח משחקים

### סמסטר ב', תשפ"ב

תרגיל 2

תאריך אחרון להגשה:

הנביאים – יום א', 27/03/2022, בשעה 23:59

שטראוס-גברים – יום ה', 24/03/2022, בשעה 23:59

שטראוס-נשים – מוצאי שבת, 26/03/2022, בשעה 23:59

## מטרת התרגיל:

תרגול תבניות (Templates).

## תיאור כללי:

בתרגיל זה נבנה אפליקציית מסוף (Console, Terminal) המשמשת למילוי טופס לרישום של סטודנט לשנת לימודים מסוימת. הסטודנט מכניס מספר פרטים אישיים וכן פרטים על לימודיו המתוכננים לשנת הלימודים המדוברת. התוכנה מוודאת שהפרטים תקינים, כולל אימות של פרטי הלימודים מול המידע הידוע מראש על אפשרויות הלימודים.

## בסיס העבודה:

אתם מקבלים קובץ main.cpp המכיל את הפונקציה main ומספר פונקציות עזר, ועליכם להוסיף קבצים מתאימים עם הקוד הנדרש כדי לגרום לקובץ להתקמפל ולרוץ. את הקובץ הזה אסור לשנות. אם ישנה בעיה בקובץ, אנא שלחו זאת בפורום. שימו לב שהקוד בקובץ בעצם מגדיר דרישות מסוימות שהמחלקות שנכתוב חייבות לממש כדי שהקובץ הזה אכן יתקמפל ויפעל כראוי.

## תיאור כללי של ריצת התוכנית:

1. התוכנית רצה בקונסול, ללא ממשק גרפי, כפי שצוין לעיל.
2. התכנית מציגה למשתמש שדה אחר שדה בבקשה למלא את הפרטים הנדרשים. כל השדות הם חובה ואם לא מילא את כולם, התכנית תמתין.
3. לאחר שהמשתמש סיים למלא את הטופס, התוכנית בודקת את תקינות השדות.

4. אם קיימים שדות לא תקינים, התוכנית תציג למשתמש הודעה על כך, ותציג את כל השדות שהזין כשליד השדות הבעייתיים תופיע שגיאה מתאימה (או אחרי רשימת השדות, אם הבעיה היא בהתאמה בין שדות). כעת התוכנית תבקש מהמשתמש להזין מחדש את המידע בדומה לסעיף 2, אולם הפעם היא תבקש זאת רק עבור השדות הלא-תקינים, ותחזור לסעיף 3.
5. אם כל השדות תקינים, התוכנית תציג הודעה המודעה על ההרשמה, תציג את המידע שהזין, ותצא.

## רשימת השדות וסוגי המידע החוקיים עבורם:

1. שם – לא יכול להכיל ספרות. אין צורך לבדוק יותר מכך, כלומר יכולים להופיע בשם מקפים, פסיקים וכו'.
2. ת.ז. – מאומתת לפי ספרת הביקורת. את האלגוריתם ניתן למצוא בוויקיפדיה בערך "ספרת ביקורת".
3. שנת לידה – לצורך התרגיל, הגיל המינימלי לרישום הוא 16. לגבי גיל מקסימלי, הנחת העבודה שלנו היא שאדם מעל גיל 100 לא הולך להירשם לאקדמיה וזו כנראה טעות. אין צורך להבדיל בהודעת השגיאה בין המקרים של גיל צעיר מידי ומבוגר מידי, מספיק לומר שזה לא בטווח. כמו כן, בחישוב הגיל לא ניכנס לרזולוציה של חודשים. ההנחה היא שרק אם ההפרש בין השנה הנוכחית לשנת הלידה הוא בטווח המוגדר זה תקין מבחינתנו.
4. חוג – המשתמש יכניס מספר בין 1 ל-3 המייצג את החוגים הבאים:
  1. מדעי המחשב (Computer Science)
  2. ספרות (Literature)
  3. רפואה (Medicine)
5. שנת לימודים – המשתמש יכניס את שנת הלימודים. אנחנו מניחים שבמוסד האקדמי הנוקשה והמקובע שלנו לא מאפשרים פריסת לימודים שונה מהמסלול הרשמי, ולכן אנחנו יכולים לאמת את הערך מול מספר השנים הקבוע לכל מסלול: 4 למדעי המחשב, 7 לרפואה ו-3 לספרות. אם המשתמש מכניס מספר קטן מ-1 או גדול מהמקסימום של כל החוגים, זו תהיה שגיאה באימות השדה הזה (ושוב, אין צורך להבדיל בין סוגי השגיאות ומספיק לציין שהמספר לא בטווח האפשרי). אם המשתמש מכניס מספר בין 1 למקסימום האפשרי אבל המספר הזה לא אפשרי בחוג המסוים, זו תהיה שגיאה באימות הטופס.
6. מספר קורסים – מספר קורסים שהמשתמש מתכוון לקחת השנה. אמנם מספר השנים מקובע אצלנו, אבל יש מעט גמישות בתכנון פריסת הקורסים ולכן לא בהכרח כולם לוקחים את אותם קורסים בשנה המסוימת. כאן המספר נקבע רק לפי שנת הלימודים (ללא הבדל בין החוגים). בשתי השנים הראשונות ניתן לקחת בין שני קורסים ל-6, בשנים 3-4 ניתן לקחת בין שני קורסים ל-10, ומשנה 5 והלאה ניתן לקחת בין שני קורסים ל-8. כאן, בחלק של אימות השדה, נוודא רק אם המשתמש הכניס מספר קטן מ-2 ונציג הודעה מתאימה. אם הוא הכניס מספר גדול מידי, זה ייתפס כחלק מאימות הטופס שיאמת את ההתאמה בין מספר הקורסים לשנת הלימודים.

# מסכי הדגמה לפלט התוכנית:

המשתמש מזין נתונים (הקלט מסומן בצהוב):

```
+-----+
| Hello and welcome!                                     |
| In order to register please fill in the fields below   |
+-----+
```

What is your name?

Yehezkel2

What is your ID?

123456789

What is your year of birth?

1884

What faculty are you registering to?

(1-Computer Science, 2-Medicine, 3-Literature)

4

Which year is it in your studies? (1-7)

0

How many courses are you taking this year?

11

התוכנית מציגה למשתמש הודעת שגיאה ואת תוכן הטופס עם הודעות שגיאה מתאימות ומבקשת מחדש רק את השדות שאינם תקינים. נשים לב שמספר הקורסים נחשב תקין כרגע במימוש הספציפי הזה כי האימות של השדה בפני עצמו בודק רק שמספר הקורסים לא קטן מ-2 והאימות מול השנה לא מוגדר היטב כשהערך בשדה השנה עצמו לא תקין ולכן ייתכן שלא תופיע שגיאה בשלב הזה. המשתמש כאן הכניס מחדש נתונים תקינים אולם כעת גילינו שמספר הקורסים לא תואם לשנת הלימודים:

```
+-----+
| There was an error in at least one of the fields!       |
| Please correct the error(s)                             |
+-----+
```

-----

--

What is your name? = Yehezkel2 Can't contain  
digits

-----

--

-----  
--  
What is your ID? = 123456789  
digit

Wrong control

-----  
--  
-----  
--  
What is your year of birth? = 1884  
range

Out of

-----  
--  
-----  
--  
What faculty are you registering to?

(1-Computer Science, 2-Medicine, 3-Literature) = 4  
range

Out of

-----  
--  
-----  
--  
Which year is it in your studies? (1-7) = 0  
range

Out of

-----  
--  
-----  
--  
How many courses are you taking this year? = 11

-----  
--  
What is your name?

Yehezkel-Bernat

What is your ID?

123456782

What is your year of birth?

1984

What faculty are you registering to?

(1-Computer Science, 2-Medicine, 3-Literature)

1

Which year is it in your studies? (1-7)

3

בגלל חוסר ההתאמה של מספר הקורסים והשנה, עדיין מוצגת הודעת שגיאה, עם סיבת השגיאה אחרי השדות (כי זו לא בדיקה של שדה מסוים אלא של שילוב של שני שדות), וכעת התוכנית מבקשת מחדש את שני השדות הללו, אבל רק אותם:

```
+-----+
| There was an error in at least one of the fields!      |
| Please correct the error(s)                            |
+-----+
```

```
-----
--
```

What is your name? = Yehezkel-Bernat

```
-----
--
-----
--
```

What is your ID? = 123456782

```
-----
--
-----
--
```

What is your year of birth? = 1984

```
-----
--
-----
--
```

What faculty are you registering to?

(1-Computer Science, 2-Medicine, 3-Literature) = Computer Science

```
-----
--
-----
--
```

Which year is it in your studies? (1-7) = 3

```
-----
--
```

-----  
--  
How many courses are you taking this year? = 11  
-----

--  
Courses and year don't match  
Which year is it in your studies? (1-7)

3

How many courses are you taking this year?

5

מכיוון שכעת הטופס תקין, אחרי שהמשתמש הכניס מספר קורסים ומספר שנה המתאימים זה לזה, התוכנית  
מציגה הודעת תודה ואת תוכן הטופס ויוצאת:

```
+-----+
| Thank you!                                     |
| This is the data you sent:                     |
+-----+
```

-----  
--  
What is your name? = Yehezkel-Bernat  
-----

--  
-----  
--  
What is your ID? = 123456782  
-----

--  
-----  
--  
What is your year of birth? = 1984  
-----

--  
-----  
--  
What faculty are you registering to?  
(1-Computer Science, 2-Medicine, 3-Literature) = Computer Science  
-----

--

-----  
--  
Which year is it in your studies? (1-7) = 3  
-----  
--  
-----  
--

How many courses are you taking this year? = 5  
-----  
--

## מה לא נדרש:

- לא נדרש מכם להתמודד עם קלט לא תקין כאשר המשתמש מכניס טקסט במקום מספרים, כלומר אינכם נדרשים להתייחס למצב שכזה.
- לא נדרש מכם לאפשר להכניס רווחים בשם, ולא נדרש לוודא תקינות אם יש רווחים או לא.
- אין צורך להקפיד שההדפסה למסך תהיה זהה למוצג לעיל (אל תתחילו לספור מקפים כדי להתאים בדיוק...)

## הערות לתיכון התוכנית:

ברור שצריך לחשוב על תיכון ראוי שיתאים לקוד הקיים כדי לאפשר לו להתקמפל ולרוץ באופן תקין, אולם חשוב גם שלא נצטמצם אך ורק למה שנדרש בקוד הקיים ונשאיר מקום להתפתחות עתידית בפרט בנושאים כמו אפשרות להוספת מספר validators לאותו שדה.

## הערות נוספות:

- שמות ה-include ימים בקובץ main.cpp יכולה לרמוז לכם על המחלקות, אולם מן הסתם קיימות מחלקות שלא נכללות כאן ישירות. כמובן שאתם רשאים להוסיף מחלקות כפי הצורך, אך אסור לכם לשנות את הקובץ main.cpp.
- שימו לב גם ל-validator-ים השייכים לטופס עצמו (ראו בקוד של פונקציית ה-main) הרלוונטיים לבדיקות הנעשות על ערכים בין כמה שדות. בדיקה כזו שנכשלת גורמת לכך שנבקש מהמשתמש מחדש את כל השדות הרלוונטיים, ולא משנה האם השדות הללו תקינים כל אחד בפני עצמו או לא.
- ניתן להניח שכל שדה מקבל רק ולידטורים מהסוג שמתאים אליו.
- אין להשתמש ב-stringstream או כלי דומה כדי להעביר את המידע בין ה-validator והשדה.

- **חשוב לזכור:** מחלקה תבניתית (Template class) יכולה גם לרשת ממחלקה אחרת, כמו שכל מחלקה אחרת יכולה לרשת, ובפרט היא יכולה לרשת ממחלקה לא תבניתית.

## הערות טכניות שימושיות:

- כדאי להריץ את הקוד מהרגע הראשון. איך תעשו זאת? תחילה תשימו בהערה את כל הקוד, ואת כל ה-include. תממשו את אחת המחלקות, תוציאו את ההערה מהשורה שמשתמשת בה ומה-include שלה, תקמפלו ותריצו. ותבדקו שאכן הקוד שלכם עובד כשורה. וכן הלאה. כך תוכלו לממש את התוכנית בהדרגה ולבדוק את הקוד שלכם מהרגע הראשון, וזה יקל עליכם בדיבאג בהמשך.
- כדי לבדוק האם תו מסוים הוא ספרה, ניתן להשתמש בפונקציה `std::isdigit()` המוגדרת בקובץ `.ctype`.
- כדאי מאוד לעבור על הקוד של ה-main כדי להבין מה בדיוק צריך לעשות ומה כבר נכלל בקוד של ה-main. למשל, האימות של שנת הלידה מוגדר כך שלמעשה אין צורך לכתוב קוד נוסף שמחשב מה טווח השנים הרצוי.

## קובץ ה-README:

יש לכלול קובץ README שיקרא README.doc, README.docx או README.txt (ולא בשם אחר). הקובץ יכול להיכתב בעברית ובלבד שיכיל את הסעיפים הנדרשים.

קובץ זה יכיל לכל הפחות:

1. כותרת.
2. פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.
3. הסבר כללי של התרגיל.
4. תיכון (design): הסבר קצר מהם האובייקטים השונים בתוכנית, מה התפקיד של כל אחד מהם וחלוקת האחריות ביניהם ואיך מתבצעת האינטראקציה בין האובייקטים השונים.
5. רשימה של הקבצים שיצרנו, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
6. מבני נתונים עיקריים ותפקידיהם.
7. אלגוריתמים הראויים לציון.
8. באגים ידועים.
9. הערות אחרות.

יש לתמצת ככל שניתן אך לא לוותר על אף חלק. אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת ומתחתיו פסקה ריקה. נכתוב ב-README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

## אופן ההגשה:

הקובץ להגשה: יש לדחוס כל קובץ הקשור לתרגיל, למעט מה שיצוין להלן, לקובץ ששמו `exN_firstname_lastname.zip`, כאשר N הוא מספר התרגיל ו-`firstname_lastname` הוא השם המלא (לדוגמא אלברט איינשטיין יגיש כך את התרגיל הראשון: `ex1_albert_einstein.zip`). במקרה של הגשה



בזוג, שם הקובץ יהיה לפי התבנית exN\_firstname1\_lastname1\_firstname2\_lastname2.zip, עם שמות המגישים בהתאמה (ללא רווחים; כלומר, גם בשמות עצמם יש להחליף רווחים בקו תחת, כפי המודגם לעיל). כמו כן, במקרה של הגשה בזוג, **רק אחד** מהמגישים יגיש את הקובץ ולא שניהם.

לפני דחיסת תיקיית הפרויקט שלכם יש למחוק את הפריטים הבאים:

- תיקייה בשם out, אם קיימת
- תיקייה בשם vs.

שתי התיקיות האלה נמצאות בתיקייה הראשית (זו שאנחנו פותחים בעזרת VS). התיקייה vs. לפעמים מוסתרת, אבל אם תפתחו את קובץ ה-zip שיצרתם, בוודאי תוכלו למצוא אותה ולמחוק אותה. ככלל אצבע, אם קובץ ה-zip שוקל יותר ממ"ב אחד או שניים, כנראה שלא מחקתם חלק מהקבצים הבינאריים המוזכרים.

**וודאו כי קובץ ה-zip מכיל תיקייה ראשית אחת, ורק בתוכה יהיו כל הקבצים ותתי התיקיות של הפרויקט.**

את הקובץ יש להעלות ל-Moodle של הקורס למשימה המתאימה.

**הגשה חוזרת:** אם מסיבה כלשהי סטודנט מחליט להגיש הגשה חוזרת יש לוודא ששם הקובץ זהה לחלוטין לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ האחרון שיוגש.

כל שינוי ממה שמוגדר פה לגבי צורת ההגשה ומבנה ה-README עלול לגרור הורדת נקודות בציון.

מספר הערות:

1. נשים לב לשם הקובץ שאכן יכלול את שמות המגישים.

2. נשים לב לשלוח את תיקיית הפרויקט כולה, לא רק את קובצי הקוד שהוספנו. תרגיל שלא יכלול את כל הקבצים הנדרשים, לא יתקבל וידרוש הגשה חוזרת (עם כללי האיחור הרגילים).

המלצה כללית: אחרי הכנת הקובץ להגשה, נעתיק אותו לתיקייה חדשה, נחלץ את הקבצים שבתוכו ונבדוק אם ניתן לפתוח את התיקייה הזו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה כזו.

**בהצלחה!**