

## מטלה - מספרים פיסיקליים, העמסת אופרטורים

בשנת 1999, לווין של נאס"א בשווי של 125 מיליון דולר התרסק בגלל אי-התאמה ביחידות - אחד הצוותים שפיתחו את הלוויין עבד ביחידות מטטריות והצוות השני עבד ביחידות בריטיות (ראו דוגמאות נוספות כאן: <http://mentalfloss.com/article/25845/quick-6-six-unit-conversion-disasters>).

כדי שזה לא יקרה שוב, הם שכרו אתכם וביקשו מכם לכתוב מחלקה המייצגת מספר עם יחידות. במחלקה הזאת אפשר, למשל, לייצג את המספר "3 מטר" ואת המספר "40 סנטימטר", והסכום שלהם לא יהיה 43 אלא 3.4 מטר - המחלקה תדאג לבצע את ההמרה המתאימה. בנוסף, המחלקה לא תאפשר לחבר מספרים עם מימדים לא תואמים, למשל, חיבור של "3 מטר" עם "5 שניות" יגרום לזריקת חריגה.

לצורך המטלה, נתמקד בשלושת הממדים של הפיסיקה הניוטונית: אורך, זמן, ומסה, וביחידות הבאות:

- אורך - מטר (m) קילומטר (km) אינץ' (in) ורגל (ft).
  - זמן - שניה (sec) דקה (min) שעה (hour) יום (day).
  - מסה - גרם (g) קילוגרם (kg) טון (ton) ליברה (lb).
- הגדירו מחלקה בשם `PhysicalNumber` עם הפעולות הבאות (ראו בקובץ המצורף `PhysicalNumberDemo.cpp`):

- שישה אופרטורים חשבוניים: חיבור (+) הוספה (=+) פלוס אונרי (+), ושלושת האופרטורים המקבילים לחיסור (-). כאמור, חיבור של שני מספרים מאותו מימד יתבצע תוך המרת היחידה של המספר השני ליחידה של המספר הראשון; חיבור של שני מספרים ממימדים שונים יגרום לחריגה.
- שישה אופרטורי השוואה: גדול, גדול-או-שווה, וכו'. לפי אותם כללים של האופרטורים החשבוניים.
- הגדלה ב-1 (++) והקטנה ב-1 (--).
- אופרטור קלט (<<) ואופרטור פלט (>>).

### קבצים

בנוסף לקבצים של הפתרון, יש להכין קובץ בשם `PhysicalNumberTest.cpp` הכולל בדיקות-יחידה מפורטות.

מצורפים לתרגיל זה הקבצים:

- `PhysicalNumberDemo.cpp` - תוכנית ראשית לדוגמה.
- `PhysicalNumberTest.cpp` - תוכנית ראשית הכוללת בדיקות-יחידה לדוגמה; המשימה שלכם היא להרחיב את הקובץ הזה כך שיכלול בדיקות-יחידה מפורטות למצבים שונים.

### הגשה ובדיקה אוטומטית

כמו במטלה הקודמת, יש להגיש קישור למאגר בגיטהאב. אנחנו נבצע את הפקודות הבאות:

1. `git clone <הקישור שלכם>`

2. נעתיק לתוך התיקיה שלכם תוכנית `PhysicalNumberTest.cpp` משלנו.

3. `clang++-5.0 -std=c++17 PhysicalNumberTest.cpp PhysicalNumberTest.cpp Unit.cpp`

4. `./a.out`

## תיכנון הזמן

כמו במטלה הקודמת, יש להגיש את הפתרון בשני שלבים.

**שלב א** - שבוע לאחר פירסום המטלה, יש להגיש פתרון המקבל ציון של 50 לפחות בבדיקות האוטומטיות שלנו. זה אומר שלכל האופרטורים צריך להיות מימוש בסיסי כדי שהתוכנית תתקמפל. כדאי להתחיל לממש את אופרטור הפלט כיוון שרוב הבדיקות משתמשות בו.

במקביל, יש להגיש קובץ `PhysicalNumberTest.cpp` הכולל בדיקות-יחידה מקיפות ומפורטות משלכם. בשלב זה כמובן חלק גדול מהבדיקות שלכם לא יעברו.

**שלב ב** - שבועיים לאחר פירסום המטלה, יש להגיש פתרון מלא, המקבל ציון של 100 בבדיקות האוטומטיות שלנו.

## בדיקה בתירגולים

בתירגולים תתבקשו להסביר על הקוד שלכם ולהראות שהוא מקיים כללים בסיסיים של הנדסת תוכנה: חלוקה לקבצים, תיעוד, שמות משמעותיים למשתנים, בדיקות תקינות ומניעת דליפת-זיכרון.

## דגשים

- יש לחזור על החומר של ההרצאות לפני שמתחילים לכתוב, ולהשתמש בו לפי הצורך.
- מוותר להשתמש בתכונות מתקדמות של שפת C++ גם אם עדיין לא נלמדו בהרצאות.
- אין להעתיק תרגילים שלמים מסטודנטים אחרים. מותר להיעזר בקטעי קוד מהאינטרנט, אולם **יש לציין בבירור את המקור**, לוודא שהקוד עובד, ולוודא שאתם מבינים למה הוא עובד.

## הרחבות ושאלות

[למחשבה בלבד - לא להגשה]

ניתן להרחיב את המטלה בכמה דרכים:

- הוספת **אופרטור כפל ואופרטור חילוק**. שימו לב - האופרטורים האלה מתנהגים בצורה שונה מחיבור וחיסור. למשל, 10 קילומטר (אורך) חלקי 2 שעות (זמן) זה 5 קמ"ש (מהירות). איך לדעתכם כדאי לייצג את הצירופים השונים של יחידות כך שיהיה אפשר, למשל, לחשב "10 קמ"ש + 5 מטר לשניה"?

2. הוספת מימד נוסף: **כסף**. לשם כך יש להמיר בין מטבעות שונים לפי השער היציג הנוכחי שלהם, שאפשר למצוא באינטרנט. איך לדעתכם אפשר לקרוא את השער היציג הנוכחי ולהשתמש בו בחישובים?