

## תרגיל בית במעבדה בשפת C - קבצי H, שימוש ב-include guards, ואתם

בתרגיל בית מודרך זה, ננסה להבין ביחד איך עובדים עם קבצי H.

הוא אמור לקחת לכם כחצי שעה.

נניח שיש לנו תוכנית יחסית פשוטה, המורכבת משתי פונקציות שיושבות בקובץ main.c:

```
#include <stdio.h>

struct Answer {
    float answer;
};

struct Answer WhatIsTheAnswer()
{
    struct Answer TheAnswer;
    TheAnswer.answer = 42.0;
    return TheAnswer;
}

void main()
{
    printf("The answer is %f\n", WhatIsTheAnswer().answer);
}
```

העתיקו את התוכנית לתוך פרוייקט VS, וודאו שבינתיים אתם עוקבים.

כעת, מכיוון שהקורס דורש להפריד את הפונקציות לקבצים שונים, נרצה להוציא את הפונקציה WhatIsTheAnswer לקובץ נפרד, שיקרא בשם המשמעותי answer.c. הוסיפו קובץ בשם זה לפרוייקט ע"י project -> add new item. הזיזו את כל הקוד מלבד הקוד של main לשם (נגיד, בעזרת cut & paste), ונסו להריץ - לא מתקמפל.

הבעיה היא שהקוד של הפונקציה main כבר לא מבין את הטיפוס של הפונקציה WhatIsTheAnswer, כי היא עברה לקובץ אחר, יחד עם הסטראקט הרלבנטי.

כיצד נפתור את הבעיה? נוודא שהקוד בקובץ main מכיר את הטיפוס של הפונקציה, אבל לא את המימוש שלה, בכך שנוסיף לו את השורה הראשונה של הפונקציה, עד הסוגריים המסולסלות לא כולל, ואת הגדרת הסטראקט. שנו את Main.c כך:

```
#include <stdio.h>

struct Answer {
    float answer;
};

struct Answer WhatIsTheAnswer();

void main()
{
    printf("The answer is %f\n", WhatIsTheAnswer().answer);
}
```

```
}
```

יש! חזר לעבוד!

כעת, מכיוון שהקורס דורש שהצהרות על פונקציות ישובו בקבצי .h, נוציא את ההגדרה לקובץ כזה. נוסיף קובץ בשם היחיד ההגיוני - answer.h (כמעט תמיד בקורס, קבצי C ו-H צריכים לבוא ביחד, עם אותו שם), דרך header file -> add new item -> project, ונזיז (לא נעתיק) אליו את השורות שהוספנו ל-main.

הוא צריך להיראות כך:

```
struct Answer {  
    float answer;  
};  
  
struct Answer WhatIsTheAnswer();
```

נסו להריץ שוב - באסה, לא מתקמפל.

הבעיה פה היא שזה שיצרנו קובץ H (להלן הדר), לא גורמת לו להיות מוסף ל-main אוטומטית. בשביל זה, צריך לעשות לו אינקלוד. הוסיפו ל-MAIN את השורה:

```
#include "answer.h"
```

וזה יחזור לעבוד.

עכשיו, יש בעיה בקבצים שלנו. נניח שאני רוצה להוסיף קובץ חדש, בשם PrintTheAnswer.c, ולהוציא אליו את ההדפסה שכרגע מתבצעת ב-main, כך ש-main תצטרך רק לקרוא לפונקציה שהוא מגדיר. PrintTheAnswer.c צריך להיראות כך:

```
#include <stdio.h>  
  
#include "answer.h"  
  
void PrintTheAnswer()  
{  
    printf("The answer is %f\n", WhatIsTheAnswer().answer);  
}
```

PrintTheAnswer.h צריך להיראות כך:

```
#include "answer.h"  
  
void PrintTheAnswer();
```

ו-MAIN צריך להיראות כך:

```
#include <stdio.h>
```

```
#include "answer.h"
#include "PrintTheAnswer.h"

void main()
{
    PrintTheAnswer();
}
```

תקבלו את שגיאת הקומפילציה:

```
error C2011: 'Answer' : 'struct' type redefinition
```

זוה בגלל שהסטראקט מוגדר פעמיים, פעם אחת דרך האינקלוד ש-MAIN עושה ל-`answer.h`, ופעם נוספת דרך האינקלוד ש-`main` עושה ל-`PrintTheAnswer.h`, שבתוכו עושה אינקלוד ל-`answer.h`. שוב.

הבעיה הזאת, שקובץ `h` "מגיע" כמה פעמים דרך שרשרת אינקלודים לאותו קובץ, נפתרת בעזרת המנגנון של `include guards`, הידוע גם כ- `header guards`. הוסיפו ל-`answer.h` את 2 השורות הבאות בהתחלה:

```
#ifndef ANSWER_H
#define ANSWER_H
```

ואת השורה

```
#endif
```

בסוף. נסו לקמפל - עובד. הנה עותק של הקובץ כמו שהוא אמור להיות אצלכם:

```
#ifndef ANSWER_H
#define ANSWER_H

struct Answer {
    float answer;
};

struct Answer WhatIsTheAnswer();

#endif
```

הסיבה היא, שההוראה `#ifndef` אומרת שמה שבא אחריה, "נכנס" לקובץ רק אם הקבוע `ANSWER_H` לא מוגדר. בפעם הראשונה שהקובץ מגיע דרך אינקלוד, הקבוע הזה באמת לא מוגדר, ולכן אנחנו "נכנסים" למה שבה אחריה. מה שבא אחריה, זה מיד הגדרה של הקבוע הזה, בעזרת `#define`. אחריו מגיעות הגדרות הפונקציה והסטראקט שאנחנו רוצים. בפעם השנייה שהקובץ מגיע דרך אינקלוד, הקבוע הזה דווקא כן מוגדר, ולכן לא "ניכנס" למה שבא אחריו. לכן, הסטראקט יוגדר רק פעם אחת.

תקציר מנהלים:

כל קובץ .h שאתם יוצרים, צריך להיפתח בשתי השורות הבאות:

```
#ifndef FILENAME_H
```

```
#define FILENAME_H
```

... here come function definitions, struct definitions...

```
#endif
```

כאשר FILENAME\_H הוא "תרגום שלכם" של שם הקובץ לאותיות גדולות (נקודה היא לא תו חוקי כשנותנים שם לקבוע). לכן אם נרצה להפעיל את הטריק הזה גם עבור PrintTheAnswer.h (עשו זאת בבקשה), נוסיף לו בהתחלה את 2 השורות:

```
#ifndef PRINT_THE_ANSWER_H
```

```
#define PRINT_THE_ANSWER_H
```

```
#include "answer.h"
```

```
void PrintTheAnswer();
```

```
#endif
```