Βιβλιοθήκη τρίτου κατασκευαστή (Third-Party library)

1. Γενικά

Third-Party library είναι μια βιβλιοθήκη που έχει αναπτυχθεί από άλλον προγραμματιστή. Η σημασία αξιοποίησης βιβλιοθήκης τρίτου κατασκευαστή είναι εμφανής από το παρακάτω απόσπασμα.

"In computer programming, a **third-party software component** is a reusable software component developed to be either freely distributed or sold by an entity other than the original vendor of the development platform. The third-party software component market thrives because many programmers believe that **component-oriented development** improves the efficiency and the quality of developing custom applications." Wikipedia

2. Η βιβλιοθήκη i2p

Ένας από τους βασικούς στόχους της άσκησης Fractions είναι να εξοικειωθείτε με τη διαδικασία αξιοποίησης βιβλιοθήκης τρίτου κατασκευαστή.

Για την άσκηση αυτή σας δίνεται η βιβλιοθήκη **i2p** (αρχείο libi2p.dll) η οποία συνοδεύεται από το αντίστοιχο αρχείο επικεφαλίδας (header file) (αρχείο i2p/h).

Η βιβλιοθήκη **i2p** σας δίνεται σε 2 εκδόσεις, η V1 δεν αξιοποιεί δομές ενώ η V2 αξιοποιεί δομές.
Έκδοση 1 (V1): αρχείο i2p.h και αρχείο libi2p.dll που περιέχεται στο libi2p.zip
Έκδοση 2 (V2): αρχεία i2p.h και libi2p.dll που περιέχονται στο libi2pV2.zip

Την έκδοση 1 χρησιμοποιεί η Δράση [Δ1] του εργαστηρίου της 6ης εβδομάδας.
Την έκδοση 2 χρησιμοποιεί η Δράση [Δ3] του εργαστηρίου της 8ης εβδομάδας.

3. Αξιοποίηση βιβλιοθήκης

Για να αξιοποιήσετε μια βιβλιοθήκη θα πρέπει να κάνετε τις παρακάτω ενέργειες:

3.1. Ε1-Κατεβάστε την βιβλιοθήκη

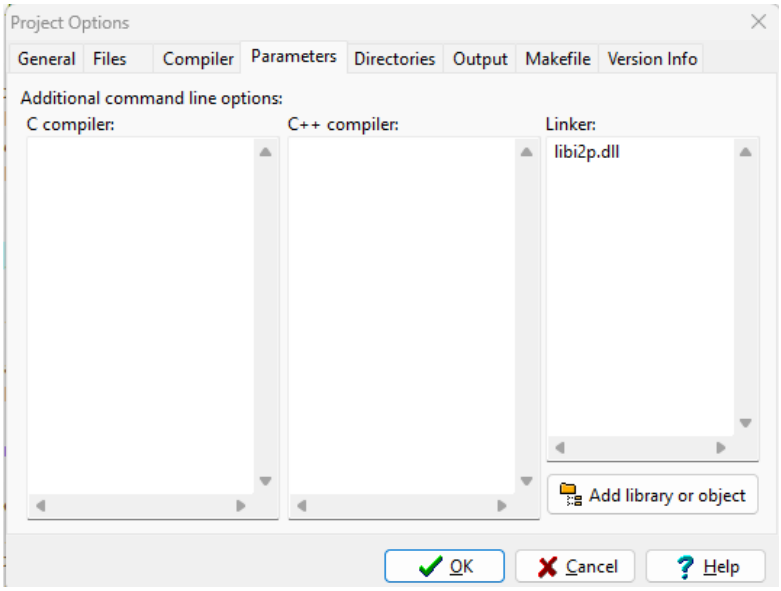
Κατεβάστε και τοποθετήστε την βιβλιοθήκη (αρχείο .h και αρχείο .dll) στο ευρετήριο του project σας.

3.2. Ε2-Συμπεριλάβετε το αρχείο επικεφαλίδας

Συμπεριλάβετε με την εντολή προεπεξεργαστή #include στο αρχείο πηγαίου κώδικα σας το αρχείο επικεφαλίδας, π.χ. #include "i2p.h"

3.3. Ε3-Ενημερώστε το IDE σας για το αρχείο .dll.

Για παράδειγμα για να ενημερώσετε το DevC++ επιλέξτε
`Project->Projects Options>Parameters` και
στην συνέχεια `Add library or Object` (Σχήμα 1).
Επιλέξτε από το ευρετήριο του project το αρχείο της βιβλιοθήκης. Επιλέξτε OK.



Σχήμα 1. Ενημέρωση IDE για το αρχείο lib12p.dll

3.4. Ε4-Καλέστε συνάρτηση της βιβλιοθήκης

Καλέστε μια συνάρτηση της βιβλιοθήκης πάντα σύμφωνα με το function prototype της που θα βρείτε στο αρχείο επικεφαλίδας της βιβλιοθήκης.

4. Παράδειγμα κλήσης συνάρτησης

Η βιβλιοθήκη i2p περιέχει την συνάρτηση readExpression() η οποία και στις 2 εκδόσεις της :

a) ζητάει από τον χρήση να εισάγει από την βασική είσοδο μια έκφραση όπως αυτή ορίζεται από την άσκηση Fractions.
b) Διαβάζει από την βασική είσοδο τα στοιχεία της έκφρασης που εισάγει ο χρήστης.

Στα στοιχεία αυτά θα πρέπει με κάποιο τρόπο να αποκτήσει πρόσβαση η συνάρτηση που κάλεσε την readExpression (δηλαδή η main για την άσκηση Fractions).

Στο σημείο αυτό είναι η διαφορά των 2 εκδόσεων της readExperssion. Τα δύο παραδείγματα που ακολουθούν παρουσιάζουν την διαφορά αυτή.

4.1. 1η έκδοση της readExperssion

Η 1η έκδοση της readExperssion, η οποία έχει το παρακάτω function prototype,

```
void readExpression(char *operatorPtr, int *op1nPtr,int
*op1dPtr,int *op2nPtr,int *op2dPtr);
```

αποθηκεύει τα στοιχεία της έκφρασης που διαβάζει από τον χρήστη στις θέσεις μνήμης που δείχνουν τα πραγματικά ορίσματα της συνάρτησης, δηλαδή οι δείκτες operatorPtr, op1nPtr, κλπ. Βάζει δηλαδή τα δεδομένα που διαβάζει σε θέσεις μνήμης που έχει δεσμεύσει η συνάρτηση που κάλεσε την readExperssion.
Με τον τρόπο αυτό η συνάρτηση που κάλεσε την readExpression έχει πρόσβαση στα στοιχεία της έκφρασης και η readExpression δεν χρειάζεται να επιστρέψει κάτι (οπότε έχει ως επιστρεφόμενη τιμή void).

Αυτό σημαίνει πως η συνάρτηση που θα καλέσει την readExperssion θα πρέπει να έχει δηλώσει τις κατάλληλες μεταβλητές για την αποθήκευση των τιμών της έκφρασης και να περάσει στην readExperssion τους δείκτες αυτών των μεταβλητών.

Παράδειγμα αξιοποίησης της συνάρτησης

Το Σχήμα 2 δίνει τον πηγαίο κώδικα από ένα παράδειγμα αξιοποίησης της συνάρτησης readExperssion. Προσέξτε την δήλωση των μεταβλητών για τα στοιχεία της έκφρασης ως γενικές μεταβλητές για να τι βλέπει και η displayExpression. Το Σχήμα 3 δίνει ένα screenshot από μία εκτέλεση του προγράμματος.

main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "i2p.h"

void displayExpression();

char operator;
int op1ar,op1par,op2ar,op2par;

int main(int argc, char *argv[]) {

    readExpression(&operator,&op1ar,&op1par
    displayExpression();
    return 0;
}

void displayExpression(){
    printf("\nExpression to evaluate: %c %c
}
```

Σχήμα 2. Παράδειγμα κλήσης της readExpression της 1ης έκδοσης της i2p.

```
C:\Code\courses\I2P2023-24_
Enter the expression (e.g. + 1/4 2/4): + 3/8 1/16

Expression to evaluate: + 3/8  1/16

--------------------------------
Process exited after 10.01 seconds with return value 0
Press any key to continue . . .
```

Σχήμα 3. Screenshot εκτέλεσης του προγράμματος κλήσης της readExpression (1ης έκδοσης της i2p).

4.2. 2η έκδοση της readExperssion

Η 2 έκδοση της readExperssion, η οποία έχει το παρακάτω function prototype,

```
Expression readExpression(void);
```

αποθηκεύει τα στοιχεία της έκφρασης που διαβάζει από τον χρήστη σε μία δική της μεταβλητή τύπου Expression και στη συνέχεια επιστρέφει αυτή την μεταβλητή στην συνάρτηση που την κάλεσε.
Αυτό σημαίνει πως η συνάρτηση που θα την καλέσει να πρέπει να έχει ορίσει μια μεταβλητή τύπου Expression και να καλέσει την readExpression για να της αποδώσει τιμή. Οι παρακάτω προτάσεις κάνουν αυτό ακριβώς. Η πρώτη δηλώνει τη μεταβλητή exp ως τύπου Expression και η 2η καλεί την readExpression για να της αποδώσει τιμή.

```
Expression exp;
exp = readExpression();
```

Με τον τρόπο αυτό η συνάρτηση που κάλεσε την readExpression έχει πρόσβαση στα στοιχεία της έκφρασης καθώς αυτά βρίσκονται στην δική της μεταβλητή (exp - local variable).

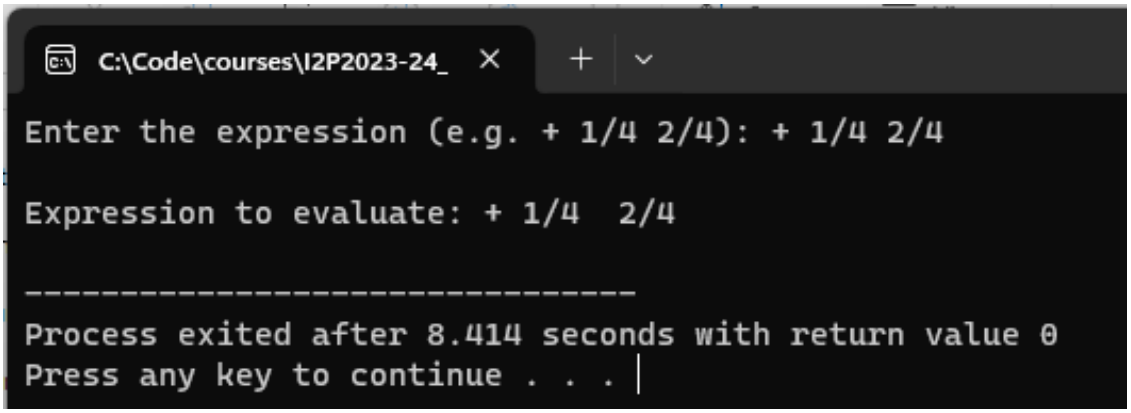
Παράδειγμα αξιοποίησης της συνάρτησης

Το Σχήμα 4 δίνει τον πηγαίο κώδικα από ένα παράδειγμα αξιοποίησης της συνάρτησης readExperssion. Προσέξτε τη δήλωση της μεταβλητής για τα στοιχεία της έκφρασης ως τοπικής μεταβλητής στην main οπότε δεν την βλέπει η displayExpression και αναγκαστικά την περνάμε ως όρισμα σε αυτήν κατά την κλήση της. Το Σχήμα 5 δίνει ένα screenshot από μία εκτέλεση του προγράμματος.

main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "i2p.h"

void displayExpression(Expression exp);

int main(int argc, char *argv[]) {
    Expression exp;

    exp=readExpression();
    displayExpression(exp);
}

void displayExpression(Expression exp){
    printf("\nExpression to evaluate: %c %
}
```

Σχήμα 4. Παράδειγμα κλήσης της readExpression της 2ης έκδοσης της i2p.



Σχήμα 5. Screenshot εκτέλεσης του προγράμματος κλήσης της readExpression (2ης έκδοσης της i2p).