

# Βιβλιοθήκες Αφηρημένοι τύποι δεδομένων

# Βιβλιοθήκες

- Τμήματα λογισμικού **ευρύτερης χρησιμότητας** που έχουν σχεδιαστεί με σκοπό να διευκολύνουν την ανάπτυξη πολλών **διαφορετικών** εφαρμογών
- Αναπτύσσονται όπως ένα (συμβατικό) ανεξάρτητο τμήμα λογισμικού – με ξεχωριστή μετάφραση και (εκ των υστέρων) σύνδεση με τον κώδικα των προγραμμάτων «πελατών» που τις χρησιμοποιούν
- Οι βιβλιοθήκες χρησιμοποιούνται χωρίς γνώση για την εσωτερική υλοποίηση τους – αρχή του black box
- Τεκμηριώνονται μέσα από **αναλυτικές περιγραφές** χρήσης της προγραμματιστικής διεπαφής τους
  - βλέπε εγχειρίδια & βιβλία προγραμματισμού

# Αφηρημένος τύπος δεδομένων (ΑΔΤ)

- Εισαγωγή νέας λειτουργικότητας με την μορφή ενός **νέου τύπου δεδομένων**
  - που υλοποιείται με βάση την αρχή του black box
- Ξεχωριστό τμήμα λογισμικού
- Χρησιμοποιείται μέσω μιας (τεκμηριωμένης) διεπαφής

# Πολύ μεγάλοι ακέραιοι

- Έστω ότι χρειαζόμαστε ειδικούς ακεραίους που μπορεί να λάβουν **πολύ μεγάλες τιμές**
  - που υπερβαίνουν (κατά πολύ) τα όρια των μεγάλων ακεραίων που υποστηρίζονται από την γλώσσα/μηχανή
- Μπορούμε να εισάγουμε ένα νέο τύπο δεδομένων, π.χ. `VeryBigInt`, με αντίστοιχες λειτουργίες αρχικοποίησης, πρόσθεσης, αφαίρεσης ...
- Ο προγραμματιστής χρησιμοποιεί την παραπάνω λειτουργικότητα χωρίς να νοιάζεται για το πως αυτή υλοποιείται «εσωτερικά»

## Διαπαφή για τον τύπο `VeryBigInt`

```
typedef ... VeryBigInt;  
  
void vbi_set(VeryBigInt *i, char *s);  
char *vbi_get(VeryBigInt *i);  
  
int vbi_add(VeryBigInt *i1, const VeryBigInt *i2);  
int vbi_diff(VeryBigInt *i1, const VeryBigInt *i2);
```

`vbi_set`

`vbi_get`

`vbi_add`

`vbi_sub`

αντικείμενο `VeryBigInt`  
ως black box

```

/* ενδεικτική διεπαφή VeryBigInt */

typedef ... VeryBigInt;

int vbi_set(VeryBigInt *i, char *s);
/* αρχικοποιεί την μεταβλητή με βάση την ακέραια
   τιμή που περνιέται ως αλφαριθμητικό, και
   επιστρέφει 1 / 0 για επιτυχία / αποτυχία */

char *vbi_get(VeryBigInt *i);
/* επιστρέφει την τιμή ως αλφαριθμητικό,
   η μνήμη του οποίου πρέπει να αποδευτεθεί
   από τον καλών κώδικα */

int vbi_add(VeryBigInt *i1, const VeryBigInt *i2);
/* i1 = i1 + i2 και επιστρέφει 1 για επιτυχία
   ή 0 αν ξεπεραστεί το υποστηριζόμενο πεδίο τιμών */

int vbi_sub(VeryBigInt *i1, const VeryBigInt *i2);
/* i1 = i1 - i2 και επιστρέφει 1 για επιτυχία
   ή 0 αν ξεπεραστεί το υποστηριζόμενο πεδίο τιμών */

```

```
int main(int argc, char *argv[]) {
    int res; VeryBigInt i1,i2; char s[512],*s2;

    scanf("%511s,s");

    res = vbi_set(&i1,s);
    if (!res) { printf("set error for %s\n",s); return; }

    scanf("%511s,s");
    res = vbi_set(&i2,s);
    if (!res) { printf("set error for %s\n",s); return; }

    res = vbi_add(&i1,&i2);
    if (!res) { printf("range error\n"); return; }

    s2 = vbi_get(&i1);

    printf("%s\n",s2);

    free(s2);

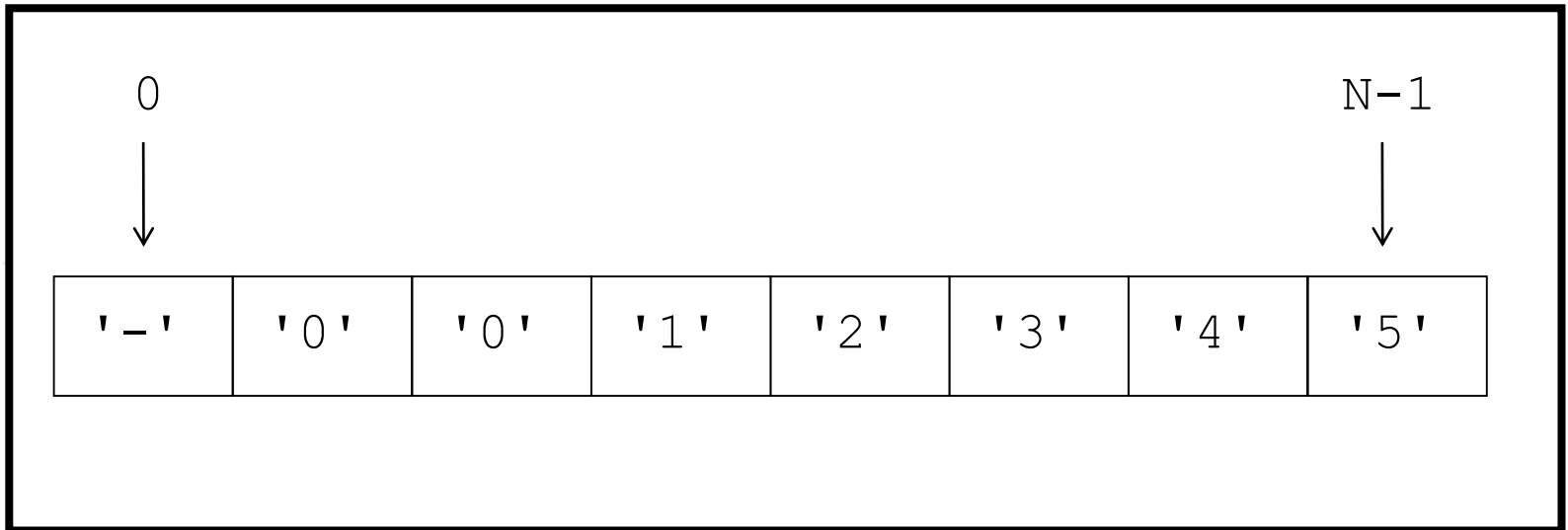
}
```

# Ενδεικτική υλοποίηση

- Κάθε μεταβλητή αντιστοιχεί σε ένα **πίνακα από χαρακτήρες**, στον οποίο αποθηκεύονται τα δεκαδικά ψηφία της τρέχουσας τιμής ως απλοί χαρακτήρες
- Το πρόσημο της τιμής αποθηκεύεται στον πίνακα
- Η `vbi_set` αρχικοποιεί τον πίνακα με βάση το αλφαριθμητικό που περνιέται σαν παράμετρος.
- Η `vbi_get` δεσμεύει ένα πίνακα χαρακτήρων στη δυναμική μνήμη και αποθηκεύει την τιμή ως κανονικό (εκτυπώσιμο) αλφαριθμητικό (με τερματικό).
- Οι `vbi_add` και `vbi_sub` υλοποιούν τις πράξεις με συμβολικό τρόπο – ψηφίο προς ψηφίο



```
VeryBigInt i;  
vbi_set(&i, "-1234");
```



# Δημιουργία στατικής βιβλιοθήκης

- Σχεδίαση της προγραμματιστικής διεπαφής
- Δημιουργία του αντίστοιχου header file
  - π.χ., `vbi.h`
- Υλοποίηση της λειτουργικότητας σε αρχείο κώδικα
  - π.χ., `vbi.c`
- Μεταγλώττιση και δημιουργία object file
  - π.χ., `gcc -Wall -c vbi.c -o vbi.o`
- Δημιουργία βιβλιοθήκης
  - π.χ., `ar rcs libvbi.a vbi.o`
  - το όνομα πρέπει να αρχίζει με `lib` και να έχει κατάληξη `.a`

# Χρήση στατικής βιβλιοθήκης

- Συμπεριλαμβάνουμε το header file της βιβλιοθήκης
  - π.χ., `#include "vbi.h"`
- Χρήση στον κώδικά μας των συναρτήσεων που παρέχει η βιβλιοθήκη
  - όπως τεκμηριώνονται στις οδηγίες χρήσης και το header file
- Μεταγλώττιση και σύνδεση με την βιβλιοθήκη
  - π.χ. `gcc myprog.c -o myprog -lvbi -L.`
- Στην εντολή σύνδεσης αφαιρούμε από το όνομα της βιβλιοθήκης το πρόθεμα `lib` και την κατάληξη `.a`, ενώ η τοποθεσία του κώδικα της βιβλιοθήκης δίνεται αμέσως μετά το `-L`