

Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών**Τμήμα Μηχανικών Η/Υ & Πληροφορικής****Πανεπιστήμιο Πατρών****Εαρινό Εξάμηνο 2019****Διδάσκοντες: Ι. Γαροφαλάκης, Σ. Σιούτας****Εργαστηριακή Άσκηση**

Το JSON αποτελεί ένα ανοικτό πρότυπο το οποίο χρησιμοποιεί κείμενο αναγνώσιμο από τον άνθρωπο με σκοπό τη μετάδοση πληροφοριακών αντικειμένων δεδομένων. Η συντομογραφία του JavaScript Object Notation είναι μια μορφή κειμένου που βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, αλλά είναι εντελώς ανεξάρτητη από την οποιαδήποτε γλώσσα προγραμματισμού. Χρησιμοποιείται ως μια εναλλακτική λύση της XML για τη μετάδοση δεδομένων μεταξύ του εξυπηρετητή και των εφαρμογών διαδικτύου.

Το JSON αποτελείται από δυο δομές. Η πρώτη δομή είναι μια συλλογή από ζεύγη ονομάτων /τιμών, στις περισσότερες γλώσσες προγραμματισμού αυτό μπορεί να γίνει αντιληπτό ως ένα αντικείμενο, μια δομή ένα λεξικό ή μια λίστα κλειδιών.

Η δεύτερη δομή είναι μια ταξινομημένη λίστα τιμών, στις περισσότερες γλώσσες προγραμματισμού αυτό μπορεί να γίνει αντιληπτό ως ένας πίνακας, μια λίστα ή μια ακολουθία.

Τα παραπάνω στο JSON παίρνουν τις παρακάτω μορφές:

- **JSON String**

Είναι μια συλλογή από 0 ή περισσότερους UNICODE χαρακτήρες που περιέχονται σε διπλά εισαγωγικά "...", χρησιμοποιώντας αντίστροφους καθετούς \ ως χαρακτήρες διαφυγής.

Παραδείγματα από JSON Strings:

- " "
- "test"
- "info \"with quotes\""

- **JSON Number**

Μοιάζουν με τους αριθμούς της C ή της Java, εκτός όμως από την οκταδική και δεκαεξαδική μορφή. Επιπλέον δεν χρειάζονται τα εισαγωγικά.

Παραδείγματα από JSON Numbers:

- 20 (ακέραιος)
- -20.50 (πραγματικός)
- 5.3e-3 (scientific notation)

- **JSON Array**

Αποτελεί μια συλλογή από τιμές σε σειρά που εμπεριέχονται μεταξύ [] και διαχωρίζονται με , (κόμμα).

Παραδείγματα από JSON Arrays:

- ο Πίνακας με Strings: ["one", "two", "three"]
- ο Πίνακας με Numbers: [50, -40, 0.50]
- ο Πίνακας με διαφορετικούς τύπους: [15, "three", true, null])

Οι κανόνες και περιορισμοί που ισχύουν στην εργαστηριακή άσκηση είναι οι εξής:

- Το στοιχείο “text” θα **πρέπει** να είναι μέχρι 140 χαρακτήρες.
- Το στοιχείο “user” θα **πρέπει** να περιέχει ένα **μοναδικό** “id” ως θετικό ακέραιο, και “name”, “screen_name”, “location” ως αλφαριθμητικά.
- Το στοιχείο “created_at” να έχει τη μορφή “Day_name MMM DD XX:XX:XX YYYY” όπου Day_name = Δευτέρα, Τρίτη, κ.ο.κ., M = Month, D = Day, XX:XX:XX η μορφή του timestamp και Y = Year. Για το timestamp φροντίστε να προκύπτει λογικό αποτέλεσμα (ώρες 0 έως 23 , λεπτά και δευτερόλεπτα απο 0 έως 60).
- Το στοιχείο “id_str” θα **πρέπει** να είναι αλφαριθμητικό και ΜΟΝΑΔΙΚΟ, ωστόσο να περιέχει μόνο έναν θετικό ακέραιο αριθμό, π.χ. “19487012” , “8623490”.
- Μπορούν να χρησιμοποιηθούν οι παρακάτω κωδικοποιήσεις χαρακτήρων:

| | |
|-------|-----|
| + | %2B |
| ! | %21 |
| space | %20 |
| , | %2C |
| & | %26 |

- Δεν είναι απαραίτητο να εισαχθούν όλα τα στοιχεία που διέπουν ένα κλασσικό JSON αρχείο, οπωσδήποτε όμως αυτά που προηγουμένως έχουν αναφερθεί στους κανόνες και περιορισμούς της άσκησης.
- Δεν είναι απαραίτητο να υπάρχει συγκεκριμένη σειρά στα στοιχεία που θα εισαχθούν.

Ένα ενδεικτικό αποδεκτό αρχείο JSON είναι το εξής:

```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1\ Today we\u2019re sharing our vision for the future
of the Twitter API platform!\nhttps://t.co/XweGngmxlP",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform
news, updates & events. Need technical help? Visit
https://twittercommunity.com/ \u2328\ufe0f #TapIntoTwitter"
  },
  "place": {
  },
}
```

```

"entities": {
  "hashtags": [
  ],
  "urls": [
    {
      "url": "https:\\\\t.co\\XweGngmxlP",
      "unwound": {
        "url":
"https:\\\\cards.twitter.com\\cards\\18ce53wgo4h\\3xo1c",
        "title": "Building the Future of the Twitter API
Platform"
      }
    }
  ],
  "user_mentions": [
  ]
}

```

Ερωτήματα

1. (60%)

- i. Δώστε σε **BNF** το συντακτικό ορισμό της γραμματικής της γλώσσας.
- ii. Χρησιμοποιώντας τα προγράμματα Flex και Bison, υλοποιήστε ένα λεξικό και ένα συντακτικό αναλυτή, ο οποίος θα παίρνει ως είσοδο ένα αρχείο γραμμένο στη γλώσσα που περιγράφηκε πιο πάνω και θα ελέγχει **σε ένα πέρασμα** αν το πρόγραμμα είναι συντακτικά ορθό. Το πρόγραμμά σας θα καλείται από τη γραμμή εντολών ως εξής:

prompt> myParser.exe file.txt

και θα επιστρέφει **στην οθόνη** το ίδιο το πρόγραμμα έχοντας εφαρμόσει κατάλληλη στοίχιση (κάθε μπλοκ πάει ένα tab «μέσα») στα διάφορα μπλοκ του προγράμματος, ένα διαγνωστικό μήνυμα για το αν ήταν ορθώς γραμμένο, ή κατάλληλο μήνυμα σφάλματος (πρέπει να φαίνεται η γραμμή όπου υπάρχει το σφάλμα).

2. (40%)

Τροποποιήστε τον αναλυτή σας ώστε:

- Να δέχεται retweets :
 - Να δημιουργεί ένα “retweeted_status” object το οποίο θα περιέχει:
 - “text” : με το αρχικό μήνυμα
 - “user” : το οποίο θα περιέχει εμφωλευμένο το “screen_name” το οποίο θα είναι αλφαριθμητικό
 - Να δημιουργεί ένα “tweet” object το οποίο θα περιέχει:
 - “text” : αλφαριθμητικό το οποίο θα ΠΡΕΠΕΙ να ξεκινάει με τη μορφή “RT @” + όνομα_δημιουργού_αρχικού_tweet + “το υπόλοιπο μήνυμα του χρήστη που κάνει το retweet” ‘
 - “user” : το οποίο θα περιέχει εμφωλευμένο το “screen_name” το οποίο θα είναι αλφαριθμητικό

- Να δέχεται extended tweets :
 - Επιπλέον boolean χαρακτηριστικό “truncated”
 - Εάν το παραπάνω είναι true τότε θα πρέπει να υπάρχει νέο στοιχείο “display_text_range” τέτοιο ώστε να είναι ένα array ακεραίων για το εύρος των διαφαινόμενων χαρακτήρων
 - Επιπλέον χαρακτηριστικό “extended tweet” :
 - “full_text” : αλφαριθμητικό μέχρι 140 χαρακτήρες
 - “display_text_range” : ουσιαστικά ένα array απο μηδέν έως το string length
 - **Εαν υπάρχουν hashtags** τότε υποχρεωτικά
 - Εμφωλευμένο στο “extended tweets” θα μπει το “entities”
 - Στο entities θα εμφωλευτεί το hashtags το οποίο θα διαθέτει εμφωλευμένα τα “text” τα οποία θα είναι το όνομα του hashtag , και το “indices” το οποίο θα είναι ένα indice ακεραίων από εκεί που αρχίζει μέχρι εκεί που τελειώνει το hashtag

Παραδοτέα

- ❖ **Γραπτή Αναφορά** που περιλαμβάνει
 - Την περιγραφή της γραμματικής της γλώσσας σε **BNF**.
 - Τα **ΤΕΛΙΚΑ** αρχεία περιγραφής της γλώσσας, τα οποία δίνονται ως **είσοδος** στα Flex και Bison
 - Screenshots παραδειγμάτων εφαρμογής (τουλάχιστον ένα παράδειγμα για κάθε ερώτημα)
 - Σχόλια
- ❖ Συμπιεσμένα σε **ένα αρχείο (zip)** τα:
 - Την πιο πάνω γραπτή αναφορά
 - Τα **ΤΕΛΙΚΑ** αρχεία εισόδου για το flex και το bison.
 - Τον **ΤΕΛΙΚΟ** κώδικα σε C (και το .h αρχείο) που παράγεται από τα δύο προγράμματα και ο εκτελέσιμος κώδικας του parser.
 - Τα αρχεία που δόθηκαν σαν είσοδο στον parser για να ελεγχθεί η σωστή λειτουργία του

Το αρχείο zip πρέπει να έχει όνομα τους αριθμούς μητρώου των ατόμων της ομάδας διαχωρισμένους με _ και από τον μικρότερο στο μεγαλύτερο (π.χ. **1000_1543_2788_3972.zip**), και να ανεβεί (**ΥΠΟΧΡΕΩΤΙΚΑ**) στο e-class. Σε ξεχωριστό αρχείο word μέσα στο zip να αναφέρονται τα ονοματεπώνυμα, το έτος, οι αντίστοιχοι αριθμοί μητρώου και οι e-mail διευθύνσεις των μελών της ομάδας

Διευκρινήσεις

- ❖ Για τη λειτουργία των εργαλείων Flex και Bison μπορείτε να βρείτε πληροφορίες στη σελίδα του μαθήματος.
- ❖ Η άσκηση θα γίνει σε ομάδες έως 4 ατόμων.

- ❖ Η συμβολή της στον τελικό βαθμό είναι 30%.
- ❖ Η άσκηση είναι υποχρεωτική και πρέπει ο βαθμός της να είναι τουλάχιστον 5.
- ❖ Τελικές ημερομηνίες παράδοσης είναι οι ημερομηνίες γραπτών εξετάσεων περιόδου Ιουνίου και Σεπτεμβρίου αντίστοιχα.
- ❖ Για τυχόν απορίες ή υποδείξεις μπορείτε να απευθύνεστε με e-mail στους μεταπτυχιακούς φοιτητές
 - Περγαντή Δημήτρη (pergantis@ceid.upatras.gr)
 - Αλκιβιάδη Σαββόπουλο (asavv@ceid.upatras.gr)
 - Γεράσιμο Βονιτσάνο (mvonitsanos@ceid.upatras.gr)

(Θα ήταν προτιμότερο πάντως να γίνεται χρήση των συζητήσεων του μαθήματος στο e-class).