Πανεπιστήμιο Ιωαννίνων
Πολυτεχνική Σχολή
Τμήμα Μηχ. Η/Υ & Πληροφορικής
Μάθημα: "Μεταφραστές ΙΙ"
Διδάσκων: Γεώργιος Μανής
Δεκέμβριος 2023

Η γλώσσα προγραμματισμού simpepp

Να υλοποιήσετε τη γλώσσα **simplepp** (simple-python-python). Ως εργαλείο ανάπτυξης να χρησιμοποιήσετε το μέτα-εργαλείο **ANTLR**. Συστηνόμενη γλώσσα υλοποίησης η Python.

Για τη γλώσσα **simplepp** ισχύουν τα εξής τα οποία πρέπει να λάβετε υπόψη σας στον σχεδιασμό της γλώσσας:

- Η γλώσσα πρέπει να είναι υποσύνολο της Python, τα προγράμματα δηλαδή που είναι γραμμένα στη γλώσσα **simplepp** να μπορούν να εκτελεστούν από έναν διερμηνέα της Python
- κάθε πρόγραμμα αποτελείται από κλάσεις και κυρίως πρόγραμμα
- το κυρίως πρόγραμμα δεν δηλώνεται ή λαμβάνεται ως κλάση
- κάθε πρόγραμμα μπορεί να έχει μία έως περισσότερες κλάσεις
- κάθε κλάση δηλώνεται με τη λέξη κλειδί **class** και ακολουθεί το όνομα της κλάσης
- αν η κλάση κληρονομεί άλλη κλάση, τότε το όνομα της κλάσης που κληρονομείται τοποθετείται μέσα σε παρένθεση, αμέσως μετά το όνομα της κλάσης που κληρονομεί
- οι κλάσεις που κληρονομούνται πρέπει να προηγούνται στον κώδικα από τις κλάσεις που τις κληρονομούν
- στην αρχή κάθε κλάσης δηλώνεται μία ακριβώς μέθοδος-δημιουργός με το όνομα **___init___**
- μετά τη μέθοδο-δημιουργό επιτρέπεται να δηλωθούν καμία ή περισσότερες μέθοδοι
- μία μέθοδος δηλώνεται με τη λέξη κλειδί **def**, ακολουθεί το όνομά της και στη συνέχεια, μέσα σε παρένθεση, οι παράμετροί της
- η πρώτη παράμετρος μιας μεθόδου είναι υποχρεωτικά η λέξη **self**, η οποία υποδηλώνει το αντικείμενο στο οποίο εφαρμόζεται
- τα πεδία των κλάσεων και οι τοπικές μεταβλητές των μεθόδων δεν δηλώνονται, όπως ακριβώς γίνεται στη γλώσσα Python
- τα πεδία των κλάσεων και οι τοπικές μεταβλητές των μεθόδων είναι όλα τύπου **int**
- οι παράμετροι των μεθόδων συμπεριλαμβάνουν αντικείμενα
- η πρόσβαση στα πεδία ενός αντικειμένου γίνεται με το όνομα του αντικειμένου, ακολουθούμενο από τελεία και στη συνέχεια το όνομα του πεδίου. Το όνομα του αντικειμένου μπορεί να είναι το **self** αν πρόκειται για αναφορά από ένα αντικείμενο στο εαυτό του

- οι εντολές που υποστηρίζει η γλώσσα είναι οι **if-else, while, print** και **return**, ενώ υποστηρίζεται επίσης και η εκχώρηση. Αν θέλετε να προσθέσετε εντολές, μπορείτε να το κάνετε. Το αρχικό πρόγραμμα πρέπει να εξακολουθεί να μπορεί να εκτελείται σε διερμηνέα Python
- υποστηρίζονται λογικές συνθήκες και αριθμητικές εκφράσεις με τον συνήθη τρόπο και με τη συνήθη προτεραιότητα τελεστών
- η εφαρμογή μίας μεθόδου γίνεται με το όνομα του αντικειμένου στο οποίο εφαρμόζεται, ακολουθούμενο από μία τελεία, το όνομα της μεθόδου και τις παραμέτρους της σε παρένθεση

Ακολουθεί ένα παράδειγμα σε γλώσσα **simplepp** το οποίο είναι το ελάχιστο παράδειγμα που πρέπει να μπορεί η γλώσσα αυτή να μεταγλωττίσει:

##

```
class Person:
    def ___init___(self, pid, born):
        self.pid = pid
        self.born = born

    def getPid(self):
        return self.pid

    def getBorn(self):
        return self.born

    def millenium(self):
        if self.born < 2000:
            return 1
        else:
            return 2
```

##

Πανεπιστήμιο Ιωαννίνων
Πολυτεχνική Σχολή
Τμήμα Μηχ. Η/Υ & Πληροφορικής
Μάθημα: “Μεταφραστές ΙΙ”
Διδάσκων: Γεώργιος Μανής
Δεκέμβριος 2023

```
class Employee(Person):
    def ___init___(self, pid, born, afm, department):
        # comment: Person.___init___(self, 0, 0)
        self.pid = pid
        self.born = born
        self.afm = afm
        self.department = department


    def getDepartment(self):
        return self.department

    def setDepartment(self, department):
        self.department = department

    def getPid(self):
        return self.afm

##########################################################################

class StupidPrint:
    def ___init___(self, employee):
        print(employee.pid)
        print(employee.born)
        print(employee.afm)
        print(employee.department)
```

Πανεπιστήμιο Ιωαννίνων
Πολυτεχνική Σχολή
Τμήμα Μηχ. Η/Υ & Πληροφορικής
Μάθημα: "Μεταφραστές ΙΙ"
Διδάσκων: Γεώργιος Μανής
Δεκέμβριος 2023

##

```
if ___name___ == '___main___':
    george = Person(200223, 2002)
    john = Person(200055, 2000)
    peter = Employee(200122, 2001, 990122, 1)
    print(george.getBorn())
    print(john.getBorn())
    print(peter.getBorn())
    print(peter.millenium())
    print(peter.getDepartment())
    peter.setDepartment(2)
    print(peter.getDepartment())
    print(george.getPid())
    print(peter.getPid())
    stupid = StupidPrint(peter)
```

##

Η γραμματική της γλώσσας πρέπει να είναι σχεδιασμένη χωρίς περιττά στοιχεία και να οι κανόνες και τα τερματικά σύμβολα να είναι επιλεγμένα έτσι ώστε να είναι τα καταλληλότερα για τις απαιτήσεις που έχουν τεθεί παραπάνω.

Αφού ολοκληρώσετε την υλοποίηση της άσκησης, να καταγράψετε σε μία αναφορά:

α) τη σχεδίαση του μεταγλωττιστή

β) την υλοποίηση του μεταγλωττιστή

γ) τον έλεγχο ορθής λειτουργίας

Στην περιγραφή της σχεδίασης μην παραλείψετε να αναφέρετε τον τρόπο με τον οποίο εξαγάγατε την πληροφορία από το αρχικό πρόγραμμα, τις δομές που χρειαστήκατε για να αποθηκεύσετε την πληροφορία αυτή και τον τρόπο με τον οποίο από τη γραμματική και την αποθηκευμένη πληροφορία παραγάγατε τον

τελικό κώδικα. Αν χρησιμοποιήσατε αντικειμενοστραφή σχεδίαση να εκμεταλλευτείτε την επιλογή αυτή και να οργανώσετε την περιγραφή με βάση τις κλάσεις που σχεδιάσατε και υλοποιήσατε.

Στην περιγραφή της υλοποίησης να κινηθείτε σε επίπεδο antlr και να περιγράψετε πώς εφαρμόσατε στον antlr τη σχεδίαση που περιγράψατε στο (α).

Στην περιγραφή του ελέγχου ορθής λειτουργίας πρέπει να δείξετε τα βήματα που πραγματοποιήσατε ώστε να βεβαιωθείτε ότι ο κώδικάς σας λειτουργεί ορθά. Να συμπεριλάβετε δικά σας προγράμματα σε γλώσσα simplepp και την έξοδο που αυτά παρήγαγαν. Το κείμενό σας πρέπει να πείθει ότι έγιναν όλοι οι απαραίτητοι έλεγχοι, δηλαδή να δικαιολογήσατε γιατί επιλέξατε τα συγκεκριμένα αυτά προγράμματα και ότι τα προγράμματα αυτά αρκούν για ένα λογικά εκτενή και βαθύ έλεγχο.

Καταληκτική ημερομηνία παράδοσης: **18 Ιανουαρίου** (μία μέρα πριν την εξέταση).

Η παράδοση θα γίνει με ηλεκτρονικό ταχυδρομείο.

Καταληκτική ημερομηνία παρουσίασης της άσκησής σας (στο γραφείο ή σε teams): **25 Ιανουαρίου**