مشخصات زبان برنامه‌نویسی SONEC

زبان برنامه نویسی SONEC (SOcial NEtwork Crawling language) زبان برنامه نویسی جدید است که با هدف جمع‌آوری اطلاعات شبکه‌های اجتماعی طراحی شده است. در این نوشتار مشخصات فنی این زبان شامل ساختار و دستورات شرح داده می‌شود.

واژگان

زبان SONEC یک زبان حساس به متن نبوده مانند سایر زبان برنامه‌نویسی دیگری دارای از مجموعه از واژگان لاتین تشکیل شده است. در ادامه به ترتیب واژگان مختلف معرفی و تعریف می‌شوند:

شناسه (Identifier): یک شناسه دنباله‌ای از حروف و اعداد است که الزاما با یک حرف باید شروع شود و طول آن حداکثر ۳۲ کاراکتر است.

کلمات کلیدی: کلمات رزرو شده در این زبان شامل واژگان استفاده شده در تعریف دستورات مانند if، then, begin, end, while,... هستند. مجموعه دقیق کلمات کلیدی با مطالعه بخش بعدی توسط شما به دست خواهد آمد.

نوع داده: نوع کلمات کلیدی هستند که جهت تعریف نوع متغیر استفاده می‌شوند. در این زبان نوع‌های زیر تعریف شده است: string , number, Bool, node, link, post, feedback, set, stack, queue که به ترتیب در بخشهای بعدی توضیح داده شده است.

جداکننده: enter بعنوان جداکننده دستورات در این زبان استفاده می‌شود.space و Tab نیز می‌تواند برای جدانمودن عبارتهای داخلی یک دستور و همین طور تعریف قلمرو دستورات استفاده شود.

ثابتهای منطقی: true, false ثابتهای منطقی هستند.

ثابتهای رشته‌ای: هر عبارت بین دو " یک ثابت رشته‌ای است. کاراکتر Enter در ثابت رشته‌ای مجاز نیست.

ثابتهای عددی: هر عدد صحیح یا اعشاری به عنوان یک ثابت عددی شناخته می‌شود. اعداد می‌توانند در مبنای ۱۰ یا ۱۶ بیان شوند. اعداد مبنای ۱۶ با 0x در ابتدای آنهای شناخته می‌شوند. ۱۲، ۳۴،۲، ۱۲۳۲۱، 0xA23 نمونه‌هایی از ثابتهای عددی هستند. در اعداد اعشاری دنباله‌ای از اعداد به همراه یک نقطه به عنوان نماد اعشار هستند. ۰,۱ ۲,۰۰۳ و ۳۳. نمونه‌های از ثابتهای اعشاری هستند.

عملگرها: عملگرهای این زبان در زیر نمایش داده شده است.

- چهار عملگر اصلی + - / *
- عملگرهای منطقی and or not xor
- عملگرهای مقایسه <> =< => = < > (<> به معنای مخالف است)
- عملگر پیمانه (باقی‌مانده %)

توضیحات (Comment): توضیحات یک خطی در این زبان با نماد # و توضیحات چند خطی بین دو نماد ## نوشته شود. توضیحات چند خطی ممکن است شامل Enter نیز باشد. توضیحات در کامپایل حذف شده و در نظر گرفته نمی‌شوند.

دستورات

یک برنامه SONEC مشابه با زبان python است. یک برنامه مجموعه‌از از دستورات و تعریف ماژولها است. دستورات این زبان در ادامه شرح داده شده‌اند. منظور از عبارت (exp) در دستورات زیر هر ترکیب مجازی از عملوندها (شامل ثابت‌ها، متغیرها و فراخوانی ماژول) و عملگرها (عملگرهای ریاضی، منطقی و رشته‌ها) و دستوراتی که دارای خروجی هستند مانند pop باشد. بعد از پایان هر دستور باید یک Enter وجود داشته باشد. قوانین تعریف یک بلاک از دستورات مشابه python است. هر بلاک دستورات باید با فرورفتگی یکسان (Tab یا space) تعریف شوند. بلاک کدها می‌توانند تودرتو باشند.

- **دستور مقداردهی اولیه:** اولین خط از برنامه حتما باید دستور مقداردهی اولیه باشد. این دستور به فرمت "{security token}" initialize networkType with تعریف می‌شود. کلمات آبی رنگ کلیدی هستند. networkType یکی از مقادیر Twitter, Instagram, Telegram است. security token اطلاعات احراز هویت لازم برای اتصال به شبکه است که بسته به نوع شبکه متفاوت است. به عنوان نمونه برای اینستاگرام دستور زیر می‌تواند استفاده شود:

Initialize Instagram with “user: jalay; pass: 123”

- **دستورات انتساب:** دستورات انتساب برای انتساب یک عبارت به یک متغیر استفاده می‌شود و به صورت id=exp است. همان‌طور که در بخش بعدی شرح داده شده است، این دستور به منزله تعریف متغیر هم عمل می‌کند.
- **دستورات خواندن و نوشتن:** زبان از دو دستور input و print برای خواندن و نوشتن از ورودی استفاده می‌کند. دستورx=input یک مقدار از ورودی خوانده و آن را در متغیر x قرار می‌دهد (متغیر x باید قبل از آن تعریف شده باشد) و دستور print(exp) مقدار عبارت exp را در چاپ می‌نویسد. کلمات input و print کلمات کلیدی هستند.
- **دستورات ذخیره و بارگذاری:** دو دستور save و load برای ذخیره و بارگذاری اطلاعات در فایل استفاده می‌شود. دستور save obj in filename محتویات obj را در filename به فرمت json ذخیره می‌کند.filename حتما باید یک رشته باشد. دستور load به فرمت x = load filename است. دقت کنید که x باید قبل از آن تعریف شده باشد. این دستورات فقط روی مجموعه قابل استفاده است.
- **دستور if, for, while:** این دستورات در این زبان مشابه با python تعریف و استفاده می‌شود.
- **تعریف ماژول:** ماژول‌ها در این زبان مانند توابع در زبان python هستند با این تفاوت که هر ماژول باید نوع خروجی آن نیز تعریف شود.

```
outputType def name (params)
    Module Codes
```

کلمات آبی رنگ در کد بالا کلمات کلیدی هستند. Params نیز برخلاف python باید شامل تعریف نوع متغیرها باشد. کد زیر یک نمونه ماژول را نمایش می‌دهد.

```
Number def Square(x:number,y:number):
    Return x*y;
```

یک ماژول ممکن است هیچ خروجی نداشته باشد، که در این صورت از بخش outputype صرف نظر می‌شود. دستور return در ماژول‌های بدون خروجی مجاز نیست. کد زیر یک نمونه از ماژول بدون ورودی و خروجی است:

```
def Hello():
    print( “Hello world!”)
```

- **فراخوانی ماژول**: فراخوانی ماژول به صورت id(params) است که در آن id نام ماژول و params ورودی‌های ماژول هستند که با ویرگول باید از هم جدا شوند. در صورتی که ماژول بدون ورودی باشد، بین پرانتزها نیاز به نوشتن عبارتی نیست. ماژول به شرطی که خروجی داشته باشد، می‌تواند در عبارتها استفاده و ترکیب شود.
- **دستور تعریف داده خالی**: دو نوع داده در این زبان وجود دارد: داده ساده و داده مجموعه‌ای. داده‌های ساده اعداد، رشته و عناصر شبکه را در بر می‌گیرد. برای تعریف یک داده خالی کافی است از type() استفاده شود که در آن type نوع داده مورد نظر است. به عنوان نمونه

```
x = node()
```

یک متغیر خالی از جنس node تعریف می‌کند. به همین ترتیب bool(), number(), string(), node(), link(), post(), feedback() برای تعریف نوع‌های متناظر استفاده می‌شود.

در مورد داده‌های مجموعه‌ای شامل مجموعه، پشته یا صف خالی لازم است علاوه بر نوع خود داده نوع عناصر داخلی آن نیز تعریف شود. به ترتیب از دستورات set(elemnttype)، stack(elemnttype) و Queue(elemnttype) استفاده می‌شود. elemttype باید یک نام داده ساده باشد.

- **دستورات len, pop, push, source,destination, owner**: این دستورات برای کار کردن با ساختمان داده‌های اختصاصی زبان تعریف شده است. جزییات این دستورات در بخش بعدی توضیح داده شده است.
- **دستورات جمع‌اوری داده از شبکه**: دستور crawl برای جمع‌آوری شبکه استفاده می‌شود. فرمت این دستور به صورت زیر است:

```
crawl data_type id
```

data_type یکی از مقادیر Node, outgoing-links,ingoing-links, posts, feedbacks است. خروجی این دستور مجموعه‌ای (set) از داده‌های جمع‌آوری شده از شبکه است. id بسته به نوع داده درخواست شده باید دارای جنس تعریف شده در بخش بعدی باشد.

تحلیل معنایی

جهت پیاده‌سازی تحلیل‌گر معنایی در این زبان شما فقط کافی دو بخش جدول نمادها و بررسی نوع را پیاده‌سازی کنید. جدول نمادها می‌بایست شناسه‌های تعریف نشده و تکراری را شناسایی کند. کامپایلر پیاده‌سازی شده از نوع تک گذره (single pass) است، به همین جهت هر متغیر و ماژول پیش از استفاده حتما باید تعریف شود.

تعریف متغیر: این زبان تعریف متغیر صریح ندارد. هر متغیر با اولین دستور انتساب همزمان تعریف نیز می‌شود. برای تعریف یک متغیر کافی است از یک دستور انتساب استفاده کنید. به عنوان نمونه دستور زیر:

name = ۱

در صورتی که متغیر name تا کنون تعریف نشده باشد، این متغیر تعریف می شود و جنس آن برابر عدد قرار خواهد گرفت. توجه کنید که جنس یک متغیر پس از تعریف قابل تغییر نیست. به عنوان نمونه اگر بعد از دستور بالا دستور زیر نوشته شده باشد:

Name = ‘amir’

این یک خطای معنایی است. متغیر name قبلا با نوع عدد تعریف شده است و انتساب رشته در متغیر عددی مجاز نیست.

توجه: در این زبان تعریف متغیرها در قلمرو (scope) مستقل از سایر قلمروها است و به همین جهت یک متغیر یا یک نام یکسان می‌تواند در قلمروهای مختلف معانی مختلف داشته باشد.

انواع داده و عملگرهای مجاز:

۱. **عدد**: برای این نوع داده تمامی عملگرهای محاسباتی و مقایسه‌ای تعریف شده است.
۲. **رشته**: برای این نوع داده فقط عملگرهای + و مقایسه‌ای تعریف شده است.
۳. **منطقی** (بولین): برای این داده عملگرهای and,or,xor تعریف شده است.
۴. **ند شبکه**: این نوع داده برای ذخیره یک گره شبکه است، هیچ عملگری برای این نوع داده تعریف نشده است.
۵. **لینک**: این نوع داده برای ذخیره یک ارتباط در یک شبکه است. فقط دو عملگر برای این نوع داده تعریف شده است: ۱)source و ۲) destination. عملگر source گره ابتدای لینک را برمی‌گرداند و عملگر دوم گره انتهای لینک را بر می گرداند. فراخوانی این عملگرها مشابه زیر است:

    node = source(link)

۶. **پست**: این نوع داده برای ذخیره‌سازی پست‌های منتشر شده توسط یک گره شبکه است. تنها یک عملگر برای این نوع داده تعریف شده است: owner. owner(post) گره صاحب پست را بر می‌گرداند.

۷. **فیدبک**: این نوع داده برای ذخیره‌سازی عکس‌العمل سایر گره‌های شبکه نسبت به یک پست تعریف شده است. فیدبکها ممکن است یک نظر (comment) یا دوست داشتن (like) باشد. مشابه با پست برای این داده تنها یک عملگر تعریف شده است: owner: برای دریافت صاحب فیدبک.

۸. **مجموعه**: این نوع داده برای ذخیره‌سازی مجموعه‌ای از عناصر است. عناصر مجموعه می تواند هر نوع داده‌ای باشد. یک مجموعه نمی تواند عضو تکراری داشته باشد. برای این مجموعه داده عملگرهای زیر تعریف شده است: len(set): تعداد اعضا مجموعه، set+set اعضای دو مجموعه را با هم جمع می‌کند و به عنوان یک مجموعه جدید برمی‌گرداند. + همچنین می‌تواند برای افزودن یک عنصر جدید به یک مجموعه استفاده شود. در این حالت فرمت دستور حتما باید به شکل set+element باشد و element باید همجنس عناصر داخل مجموعه یا قابل تبدیل به آنها باشد.

۹. **صف و پشته**: این دو مجموعه داده برای ذخیره‌سازی پشته و صف تعریف می‌شوند. برای این دو داده سه عملگر زیر تعریف شده است: len تعداد عناصر داخل این مجموعه‌ها را نشان می‌دهد، pop برای خارج کردن اولین عنصر آنها (طبق قانون صف یا پشته) است و به فرمت pop stack/queue فراخوانی می‌شود و خروجی آن یک عنصر است. push جهت افزودن یک عنصر جدید به مجموعه استفاده می‌شود و فرمت آن به صورت push stack/queue in element است.

تبدیل نوع و سایر قوانین معنایی

قوانین معنایی دستورات زبان مشابه با سایر زبانهای کلاسیک است (شرط if باید از نوع bool باشد و ...). در اینجا تنها قوانین مهم یا متفاوت ذکر می‌شود. در این زبان bool<number<string<Node، بنابراین bool قابل تبدیل به عدد و عدد قابل تبدیل به رشته و رشته قابل تبدیل به یک گره شبکه است. منظور از تبدیل رشته به یک گره شبکه در نظر گرفتن محتویات رشته به عنوان شناسه کاربری آن گره در شبکه است. در تبدیل ضمنی گزاره منطقی به عدد، درست معادل یک و غلط معادل صفر در نظر گرفته می‌شود. (توجه کنید که برعکس آن مجاز نیست، برخلاف زبان C).

نوع داده ورودی و خروجی دستور crawl به صورت زیر است:

Data_type	جنس Id	جنس خروجی

Node	Node	Node
Set(link)	Node	outgoing-links
Set(link)	Node	ingoing-links
Set(post)	Node	Posts
Set(feedback)	post	Feedbacks

دستورات ورودی و خروجی: دستور input یک متغیر از جنس رشته از ورودی می‌خواند. دستور print(exp) فقط قادر به نوشتن رشته در خروجی است. بنابراین عبارت exp در این دستور باید از نوع رشته باشد. دستور save و دستور load فقط بر روی مجموعه set عمل می‌کنند.

قوانین در مورد ماژول به شرح زیر است. تعریف و فراخوانی ماژولها در این زبان مشابه python است اما اولین دستور برنامه به صورت ثابت حتما باید دستور مقداردهی اولیه باشد. این دستور فقط یک بار و در خط اول مجاز است. ماژولهای همنام در این زبان مجاز هستند، ولی به این شرط که پارامترهای آنها مختلف باشد و قابل تبدیل به همدیگر نباشند. همچنین توجه شود که تعریف ماژولها همگی باید در قلمرو اصلی root اضافه شود.

یک نمونه از برنامه به زبان SONEC (دقت کنید زبان به حروف کوچک و بزرگ حساس نیست)

```
###This is a sample Written in SONEC
The program crawl 3 user starting from a given user###
    User = node()
    Users = set(user)
    Q = Queue(user)
    Posts = set(post)

    User = input #get the username
    Push user in Q
    While len(users) < 10:
        User = pop q
        User = crawl node user
        Users = users + user
```

```
        Newposts = crawl posts user
        Posts = posts + newposts
        Links = crawl outgoing-links user
        For l in links:
                User = destination(l)
                Push user in q
Save users in “user.data”
Save posts in “posts.data”
```

مراحل انجام و تحویل پروژه

نمره‌دهی پروژه از ۱۰۰ است. تحول پروژه در طی چهار مرحله مستقل است، به این معنا که کیفیت تحویل در یک مرحله تاثیری در مرحله بعدی ندارد و نمره هر مرحله بعد از گذشت موعد آن دیگر قابل جبران نیست. جهت تحویل پروژه تعدادی فایل ورودی به شما داده خواهد شد و برنامه شما باید به ازای هر فایل خروجی صحیح را تولید کند. هر خروجی صحیح بخشی از نمره آن مرحله را تشکیل می‌دهد. به عنوان نمونه در تحویل مرحله اول ۵ فایل وجود خواهد داشت که هر یک ۲ نمره از ۱۰ نمره مرحله اول را تشکیل می‌دهد. خروجی یک فایل حتی اگر فقط بخشی از آن اشتباه باشد کلاً اشتباه در نظر گرفته می‌شود. بنابراین سعی کنید، پروژه را قبل از تحویل با در نظر گرفتن همه حالات مختلف کامل تست کنید چرا که فرصت مجددی نخواهید داشت.

مرحله اول (۱۰ امتیاز): بخش تحلیل‌گر لغوی این زبان را پیاده‌سازی کند به نحوی یک فایل با پسوند QUPLA حاوی کد برنامه دریافت کند و در خروجی دنباله توکن‌های یافت شده در آنرا چاپ کند. اگر فایل دارای خطای لغوی بود، پیام مناسب چاپ کرده اما بتواند باقی آنرا ترجمه کند (فرآیند تحلیل نباید با رسیدن به خطای لغوی متوقف شود)

مرحله دوم (۲۰ امتیاز): در این مرحله شما باید با افزودن تحلیل دستوری مرحله قبل را کامل کنید. برنامه مجددا باید فایل کد را دریافت کند و به عنوان در خروجی تنها کافی است لیستی از خطاهای دستوری را چاپ نماید. برنامه نباید با رسیدن به اولین خطا متوقف گردد و باید بتواند کل برنامه پویش کند.

مرحله سوم (۲۰ امتیاز): هدف از این مرحله پیاده‌سازی جدول نمادهاست. در این مرحله کامپایلر باید بتواند بعد از تحلیل لغوی و معنایی، شناسه‌ها به کار رفته در برنامه را تشخیص و در جدول نمادها اضافه نماید. همچنین

برنامه باید دو نوع خطا را تشخیص و گزارش دهد. ۱) شناسه‌های تکراری: شناسه‌ای که در یک قلمرو دوبار تعریف شده است. ۲) شناسه‌های تعریف نشده: شناسه‌ای که تعریف نشده ولی از آن در عبارات استفاده شده است.

مرحله چهارم (۳۰ امتیاز): آخرین مرحله از تحویل پروژه پیاده‌سازی کامپایلر زبان است که شامل پیاده‌سازی و افزودن بررسی‌کننده نوع به مراحل قبلی است. برنامه نوشته شده باید فایل کد را به عنوان ورودی دریافت کند و اگر برنامه شامل هر نوع خطای معنایی، دستوری یا لغوی است در خروجی نمایش دهد.

مرحله پنجم (۲۰ نمره): در این مرحله برنامه شما باید قادر باشد فایل ورودی را اجرا کند و خروجی مناسب تولید کند این نمره را کسب خواهید کرد. جهت اجرای کد لازم نیست برنامه به زبان ماشین ترجمه شود. شما می‌توانید کدی به یک زبان واسط (مانند python) تولید و سپس آنرا اجرا نماید.

امتیازات ویژه

با انجام بخش‌های زیر شما می‌توانید نمرات ویژه علاوه بر نمره اصلی پروژه کسب کنید. (نمره اصلی پروژه از ۱۰۰ بوده و با احتساب این موارد تا سقف ۱۳۰ نمره قابل افزایش است). که معادل آن حدود دو نمره علاوه بر بیست در نمره نهایی شما لحاظ می‌شود. تحویل این مراحل باید در آخرین مرحله از پروژه است.

امکان ارتباط با سایر زبان‌های برنامه‌نویسی (۱۵ نمره): در صورتی که برنامه شما قادر باشد به شیوه‌ای از سایر زبان‌های برنامه‌نویسی ورودی دریافت یا خروجی ارسال کند، این نمره اضافی را خواهید گرفت. نحوه ارتباط باید بر اساس یکی از روشهای messeging مانند (MQTT یا Message Queuing) باشد. و فرمت ارسال اطلاعات باید json باشد. برای این منظور احتمالا باید چند دستور جدید به زبان اضافه کنید.

ایجاد ویرایشگر زبان (۱۵ نمره): ویرایشگر زبان محیطی است که شما می‌توانید در آن همزمان با تایپ خطاهای کامپایل را مشاهده کنید (همانند محیط Visual Studio). این ویرایشگر همزمان با تایپ (توجه کنید همزمان نه با زدن یک کلید) لازم است ۱) کلمات کلیدی را تشخیص و رنگی کند و ۲) خطاهای کامپایل را به همراه خط آن نمایش دهد (یا زیر آن خط بکشد). جهت ترجمه باید از روشهای مدیریت خطا استفاده شود و عملیات ترجمه با رسیدن به اولین خطا متوقف نشد.