



دانشکده مهندسی کامپیوتر

طراحی و پیاده سازی رابط گرافیکی کاربر برای نرم افزار کلیپس با ویژگی تطبیق پذیر بودن

پروژه پایانی برای دریافت درجه کارشناسی در رشته مهندسی کامپیوتر

علی اسدی

استاد راهنما

دکتر محمد رضا کنگاوری

پاییز ۱۴۰۲

الحمد لله رب العالمين
والصلاة والسلام على
سيدنا محمد وآله الطيبين
الطاهرين

MRTsoft

تأییدیه ی هیأت داوران جلسه ی دفاع از پایان نامه

نام دانشکده: دانشکده مهندسی کامپیوتر

نام دانشجو: علی اسدی

عنوان پایان نامه: طراحی و پیاده سازی رابط گرافیکی کاربر برای نرم افزار کلیپس با ویژگی تطبیق پذیر بودن

تاریخ دفاع: پاییز ۱۴۰۲

رشته: مهندسی کامپیوتر

ردیف	سمت	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه	امضا
۱					
۲					
۳					

تأییدیه ی صحت و اصالت نتایج

باسمه تعالی

اینجانب علی اسدی به شماره دانشجویی ۹۶۵۲۱۰۳۸ دانشجوی رشته مهندسی کامپیوتر مقطع تحصیلی کارشناسی تأیید مینمایم که کلیه ی نتایج این پایان نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب مینمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده ی اینجانب خواهد بود و دانشگاه هیچ گونه مسؤولیتی در این خصوص نخواهد داشت

نام و نام خانوادگی: علی اسدی

تاریخ و امضا

مجوز بهره برداری از پایان نامه

بهره برداری از این پایان نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین میشود، بلامانع است:

- بهره برداری از این پایان نامه برای همگان بلامانع است.
- بهره برداری از این پایان نامه با اخذ مجوز از استاد راهنما، بلامانع است.
- بهره برداری از این پایان نامه تا تاریخ ممنوع است.

استاد راهنما

تاریخ:

امضا:

تقدیم به

پدر و مادرم

تشکر و قدردانی

سپاس خداوندگار حکیم را که با لطف بی کران خود، آدمی را زیور عقل آراست
در آغاز وظیفه خود می دانم از زحمات بی دریغ استاد راهنمای خود، جناب آقای دکتر کنگاوری صمیمانه
تشکر و قدردانی کنم که قطعاً بدون راهنمایی های ارزنده ایشان، این مجموعه به انجام نمیرسید
در پایان، بوسه میزنم بر دستان خداوندگاران مهر و مهربانی پدر و مادر عزیزم و بعد از خدا، ستایش میکنم وجود
مقدس شان را و تشکر می کنم از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که
بهترین پشتیبان من بودند.

علی اسدی

پاییز ۱۴۰۲

چکیده

کلیپس CLIPS یک محیط نرم افزاری برای تولید سیستم های خبره است که فاقد یک رابط گرافیکی بر خط است. کلیپس یک زبان برنامه نویسی توصیفی است که برخلاف زبان های رویه ای که در آن ها باید روند و چگونگی کار برای سیستم بیان شود تا سیستم ساخته شود، در این زبانها تعریف صورت مسئله درسیستم، منجر به حل مسئله می گردد

هدف این پروژه طراحی و تولید یک رابط کاربری گرافیکی برای کلیپس است که استفاده از کلیپس را برای طراحان و کاربران تسهیل می نماید.

برای تولید این رابط کاربری از زبان برنامه نویسی جاوا اسکریپت و کتابخانه react استفاده شده و در نهایت خروجی دو فایل به زبان های پایتون (.py) و کلیپس (.clp) بوده که کاربران می توانند از آن ها در محیط پایتون و برنامه کلیپس استفاده کنند.

برای پیاده سازی یک برنامه خبره نیاز به یک سری اطلاعات داریم که چگونگی پیاده سازی این سیستم را برای ما تشریح کند. این اطلاعات توسط تحقیق و پژوهش از فرد خبره در آن زمینه به دست می آید که قوانین ما را تشکیل می دهند

پس از آن که این اطلاعات را کسب کردیم نیاز به دسته بندی و ایجاد یک درخت تصمیم برای سیستم خبره خود داریم. این درخت می تواند همان سناریو سیستم شما باشد که نحوه کسب دانش و واکنش ها به دانش ها را مشخص می کند.

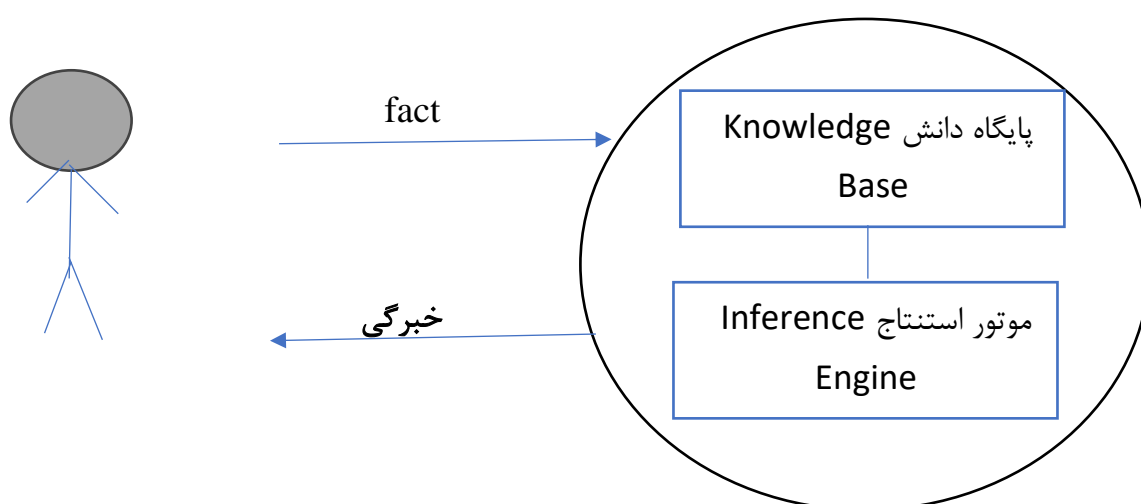
۱. مقدمه.....	۱۱
۲. برنامه کلیپس.....	۱۳
۳. راه اندازی برنامه.....	۳۰
۴. نحوه کار برنامه.....	۳۲
۵. پیاده سازی یک سناریو.....	۳۶
۶. منابع و مراجع.....	۵۲

۱..مقدمه

سیستم خبره یک برنامه کامپیوتری هوشمند است که از دانش و روش های استنتاج برای حل مسائلی استفاده می کند که به دلیل مشکل بودن نیاز به تجربه و مهارت انسان دارد. این سیستم یک واقعیت (Fact) را از بیرون از سیستم دریافت می کند و با توجه به آن واقعیت با پاسخ و راه حل مناسب (خبرگی) را به عنوان خروجی می دهد. این سیستم در حالت کلی از ۲ قسمت تشکیل شده است:

پایگاه دانش Knowledge Base

موتور استنتاج Inference Engine



واقعیت fact

واقعیات همان اطلاعاتی است که به عنوان ورودی به سیستم خبره داده می شود.

قواعد rule

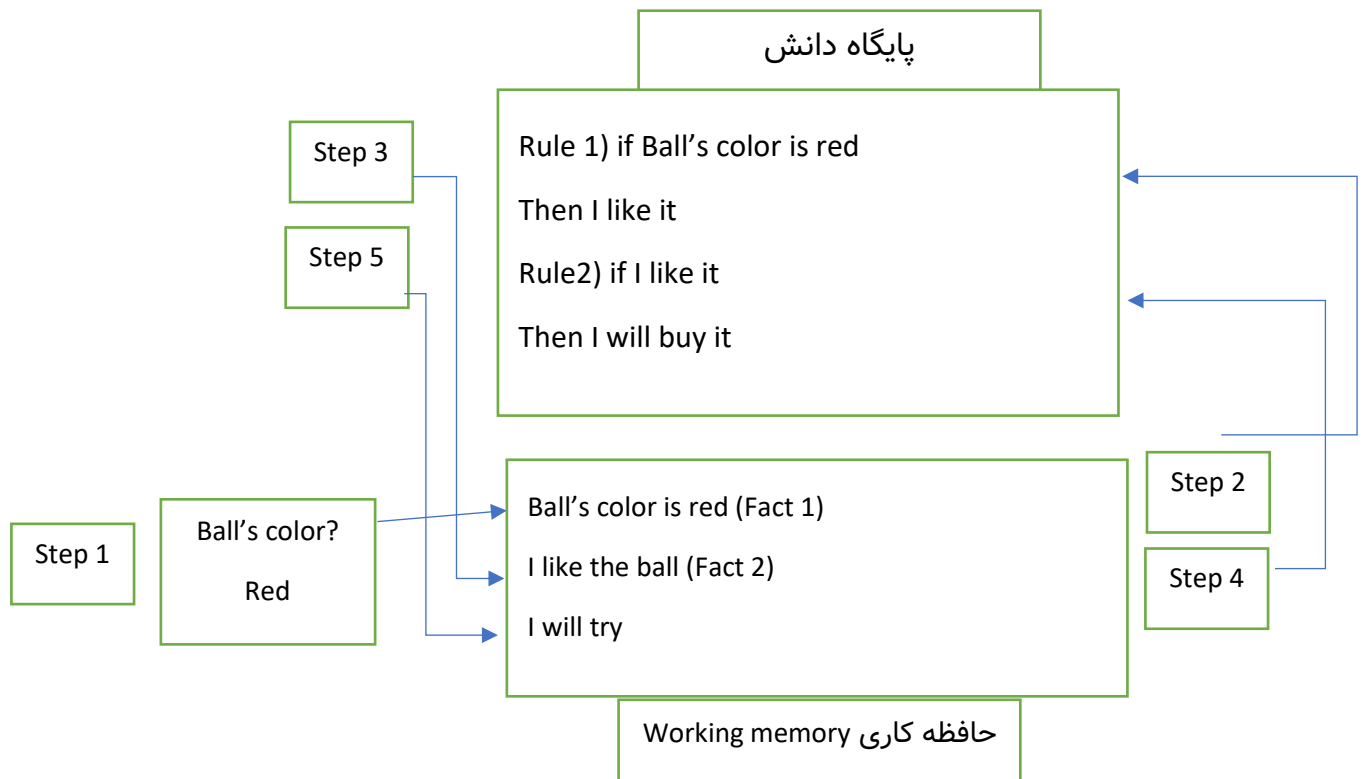
ساختار دانشی است که یعنی اطلاعات دانسته به اطلاعات دیگری می تواند به معلومات قبلی اضافه و یا دانسته فرض شوند. ساختار قاعده یا چند پیش فرض در قسمت "اگر" را با نتیجه گیری ها در قسمت "آنگاه" مرتبط می سازد. یک قاعده می تواند "در غیر این صورت" هم داشته باشد.

در واقع قاعده ساختاری است که با برقرار شدن قسمت الگو "pattern" قسمت عمل "action" اجرا می شود که اگر این عمل خودش معادل یک الگو دیگر شود عمل مربوط به آن اجرا می شود و به همین ترتیب تا آخر ادامه پیدا می کند.

If pattern

Then action

Else action



در این مثال واقعیات (Facts) در حافظه کاری و قواعد (Rules) در پایگاه دانش قرار دارند. قاعده Rule1 اجرا می شود و قسمت الگوی آن Fact 1 تطبیق دارد یعنی (Ball's Color Is Red) پس چون قسمت الگو با Fact1 تطبیق داشت قسمت عمل Rule1 یعنی I like It انجام می شود. سپس Rule2 اجرا می شود چون الگوی آن با Fact 2 که این قسمت توسط عمل Rule 1 ساخته شده است تطبیق دارد قسمت عمل Rule2 یعنی Twill Buy It اجرا می شود.

نرم افزار کلیپس (CLIPS)

از این نرم افزار برای ساخت سیستم های خبره استفاده می شود.

برخی از ویژگی های نرم افزار به شرح زیر می باشد.

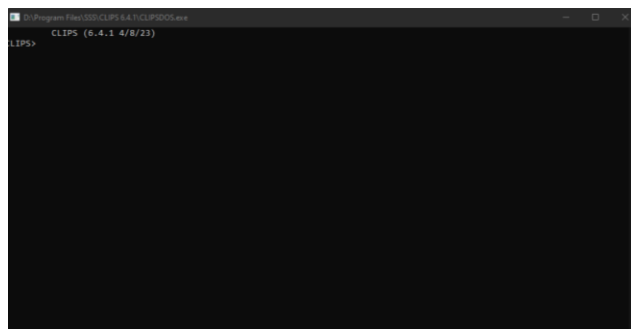
- از سر نام های عبارات C Language Implementation Production System گرفته شده است
- برای اولین بار توسط NASA در واحد فناوری نرم افزار (Software Technology Branch) تولید شده است.
- این نرم افزار رایگان بوده و در حال حاضر نسخه ۶.۳ آن موجود می باشد.
- این نرم افزار بر روی سیستم عامل های متفاوت مانند ویندوز لینوکس. مک نصب می گردد.
- این نرم افزار با استفاده از زبان C ساخته شده است.
- این نرم افزار چند کاره می باشد یعنی می توانید بصورت مدل های زیر با آن برنامه نویسی کنید.
 - بر مبنای قواعد (Rule-based)
 - بر مبنای شی گرای (Object-oriented)
 - بر مبنای روال ها (Procedural programming)
- نرم افزار کلیپس فقط قواعد زنجیره ای پیشرو را پشتیبانی می کند.

دستیابی به نرم افزار

این نرم افزار یک برنامه رایگان می باشد که هم اکنون نسخه ۶.۳ آن موجود می باشد می توانید این نرم افزار را برای سیستم عامل های موجود یعنی ویندوز. لینوکس و مک تهیه نمایید به همین منظور می توانید به سایت زیر مراجعه کرده و آن را دانلود نمایید.

<http://clipsrules.sourceforge.net>

برنامه کلیپس شامل دو کنسول برای نوشتن و اجرا نمودن برنامه ها می باشد. یک کنسول تحت داس می باشد محیط شبیه سیستم عامل داس Command Prompt دارد و تمام محیط تعاملی به صورت دستوری می باشد. و دارای یک محیط ویژوالی مانند دیگر نرم افزار های تحت ویندوز می باشد.



مولفه های پایه نرم افزار

- فیلد field
- واقعیات fact
- قواعد rule
- الگو template
- کلاس class

فیلدها

برای ساخت پایگاه دانش باید اطلاعات از ورودی (صفحه کلید یا فایل) خوانده شود سپس دستورات اجرا شوند. در طی فرایند اجرای برنامه، نرم افزار کلیپس نشانه ها Symbols، کاراکتر هایی که یک معنی می دهد را با یکدیگر ترکیب و گروه بندی می کند و یک توکن می سازد. در واقع فیلد یک نوع خاصی از توکن می باشد.

فیلد عددی

دو نوع فیلد عددی وجود دارد.

عدد صحیح Integer مانند ۲۳۸ ، -۳۲

عدد ممیز شناور Float مانند $2.73E3$ ، 0.215 ، $-25.E7$ همان نماد علمی می باشد.

فیلد نشانه ای

- فیلد های نشانه ای حاوی کاراکتر های اسکی می باشند.
- این فیلد ها دارای محدودیت هایی می باشند.
- نشانه ها در کلیپس حساس به حروف Case Sensitive هستند.
- نشانه ها نمی توانند با کاراکتر های ؟ و \$ شروع شوند چون این ۲ کاراکتر برای معرفی متغیر ها رزرو شده اند.
- نمونه هایی از نشانه ها : foo ،

محدودیت فیلد نشانه ای

فیلد های نشانه ای نمی توانند شامل سمبل های زیر باشند.

- کاراکتر های اسکی چاپ نشدنی (مانند کاراکتر فاصله Space)
- دابل کوتیشن ” ”
- پرانتز با و بسته ()
- امپرسند &

- خط عمودی|
- علامت کوچکتر<
- علامت تیلدا~
- سمی کولن;

فیلد رشته ای

فیلد های رشته ای در بین ۲ دابل کوتیشن ” ” قرار می گیرند. تنها محدودیت در این فیلد ها استفاده از بک اسلش \ می باشد. که برای چاپ بک اسلش از \\ و برای چاپ ” از \ استفاده می شود. در نمونه هایی آورده شده است.

- “foo”
- “a and b”
- “a\”quote”
- “1 number”

فیلد آدرس

این فیلد ها آدرس داده ها را نگه داری می کنند مانند زبان C.

واقعیات (Fact)

Fact ها در واقع واقعیات هایی هستند که کاربر به عنوان ورودی به سیستم خبره به می دهد و از سیستم خبره ، خبرگی دریافت می کنند. کلیپس برای حل مسئله ها نیاز به اطلاعات و داده ها دارد. پایه واحد های داده که در قواعد (Rule) استفاده می شود همان Fact می باشد. هر Fact از قسمت های زیر تشکیل شده است.

- نام واقعیت (Relation name)
- تعداد صفر یا بیشتر اسلات (Slot name) به همراه مقدار (Slot value)

Syntax:

```
Relation Name(
    (SlotName1 SlotValue1)
    (SlotName2 SlotValue2)
)
```

مثال

```
Person (
    (Family "Asghari")
    (Age 24)
)
```

ساختار Deftemplate:

با استفاده از این دستور می توان ساختار **fact** ها را بصورت گروهی ایجاد نمود. باید دقت نمود که این دستور فقط ساختار اولیه **Fact** را می سازد و هیچ **Fact** را ایجاد نمی کند و باید با دستور **Assert** آن ساختار ایجاد شده توسط **Deftemplate** را مقدار دهی نماییم.

Syntax:

```
( deftemplate <relation-name> [<optional-comment>]
  <slot-definition>* )
```

<slot-definition>

(slot <slot-name>) | (multislot <slot-name>)

)

<relation-name>: نام الگو می باشد.

<optional-comment>: می توان در این قسمت توضیحاتی برای الگو قرار داد. این قسمت اختیاری می باشد.

(slot <slot-name>): این قسمت اجازه می دهد یک فیلد با یک مقدار ذخیره گردد.

(multislot <slot-name>): این فیلد اجازه می دهد چندین مقدار در یک اسلات ذخیره شود.

در مثال زیر یک فقط ساختار ایجاد شده است و تا زمان مقدار دهی با دستور **Assert** این الگو به **Fact** تبدیل نمی شود.

(Deftemplate Person “This Template is For one Person”

(MultiSlot Name)

(Slot Age)

(Slot Eye-Color)

(Slot Hair-Color)

)

دستور Assert

مقداردهی ساختار Template با دستور Assert انجام می شود.

Example:

```
(Assert Person(  
  (Name John B. Jakson)  
  (Age 25)  
  (Eye-Color Brown)  
  (Hair-Color Black)  
)  
)
```

با دستور **Assert** می تواند **Fact** هایی بصورت جداگانه نیز تعریف کرد که در این صورت نیازی به تعریف **Template** نمی باشد.

(Assert(Color Red))

مثال فوق یک **Fact** با نام **Color** با مقدار **Red** به **Fact-List** اضافه می کند.

(Assert(Color Red)

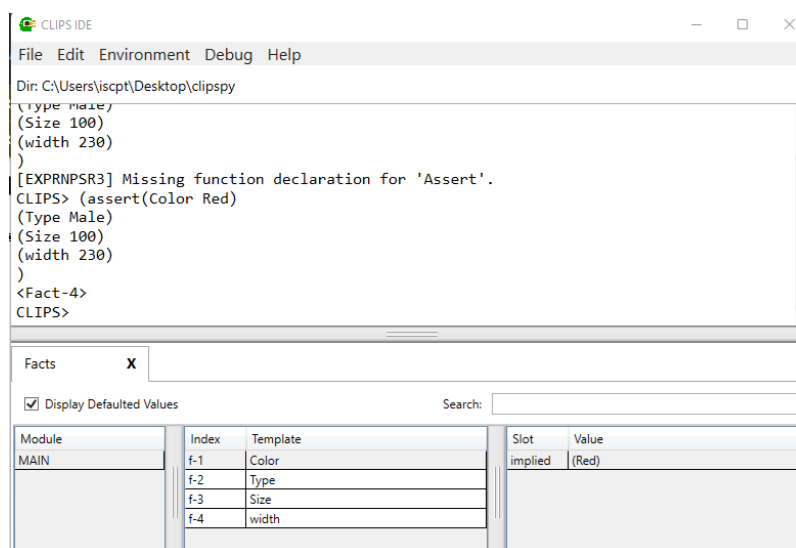
(Type Male)

(Size 100)

(width 230)

)

مثال فوق چهار **fact** به ترتیب به نام های **Color** و **Type** و **Size** و **width** با مقادیر **Red** و **Male** و **۱۰۰** و **۲۳۰** به طور همزمان اضافه می نماید



محدودیت ها برای اسلات

با اعمال محدودیت ها می توان اسلات های یک تمپلیت را مجبور به وارد کردن داده های با قالب خاص نمود و یا مقداری پیش فرض برای اسلات ها در نظر گرفت.

constraint-attribute> ::= <type-attribute > |
 < allowed-constant-attribute | >
 < range-attribute | >
 < cardinality-attribute >
 < default-attribute >
) < type-attribute> ::= (type <type-specification) >
 < <type-specification> ::= <allowed-type>+ | ?VARIABLE
 < allowed-type> ::= SYMBOL | STRING | LEXEME |
 INTEGER | FLOAT | NUMBER |
 INSTANCE-NAME | INSTANCE-ADDRESS |
 INSTANCE | FACT-ADDRESS |
 EXTERNAL-ADDRESS
 <allowed-constant-attribute>
 ::= (allowedsymbols <symbol-list>) |
 (allowedstrings <string-list>) |
 (allowed-lexemes <lexeme-list>) |
 (allowedintegers <integer-list>) |
 (allowedfloats <float-list>) |
 (allowednumbers <number-list>) |
 (allowed-instance-names <instance-list>) |
 (allowedvalues <value-list>)

مثال

```
deftemplate person
```

```
(slot name (type STRING))
```

```
(slot age (type INTEGER))
```

```
(slot gender (allowed-symbols male female))
```

```
)
```

نام فقط می تواند نوع

رشته ای قبول نماید

سن فقط می تواند نوع

عدد صحیح قبول نماید

جنیست (Gender) فقط می تواند شامل یکی از ۲ مقدار "Male" یا "Female" باشد.

به عنوان مثال در صورتی که سعی کنید مقداری بر خلاف محدودیت تعریف شده وارد نمایید با خطایی از سوی نرم افزار کلیپس مواجه خواهید شد.

متغیر ها Variable

در نرم افزار کلیپس میتوان ۲ نوع متغیر تعریف نمود.

• متغیر تک مقداری (Single Value) **<Symbol>?**

• متغیر چند مقداری (Multi Values) **<Symbol>\$?**

ویلکارد ها (Wildcard)

ویلکاردها علامت هایی هستند که بجای هر کاراکتر یا کاراکتر خاصی بکار می روند.مانند * که در هنگام جستجو به معنای ”هر” بکار می رود.

بدست آوردن شماره ایندکس Fact

برای انجام بعضی از عملیات ها نیاز به شماره ایندکس **Fact** می باشد به همین دلیل از روش زیر شماره ایندکس را بدست می آوریم.باید بدانیم که شماره ایندکس منظور همان شماره ای است که در پنجره **Fact-List** در سمت چپ **Fact** نشان داده می شود و این شماره را خود نرم افزار کلیپس بصورت اتوماتیک به **Fact** ها اختصاص می دهد.

<variable-symbol> <- <pattern-CE>

<variable-symbol>:نام یک متغیر است که شماره ایندکس در آن قرار می گیرد.

<pattern-CE>:نام **Fact** است که می خواهیم شماره ایندکس آنرا بدست بیاوریم.

<- : علامت انتساب می باشد.

f-0 (initial-fact)

f-1 (data 1 blue)

f-2 (data 1 blue red)

f-3 (data 1 blue red 6.9)

در لیست بالا تعداد چهار **Fact** وجود دارد شماره ایندکس 0 یعنی **f-0** که توسط خود نرم افزار کلیپس رزرو می شود و در واقع یک مقدار دهی اولیه انجام می دهد حتی اگر هیچ **Fact** تعریف نشده باشد این شماره به عنوان **f-0 (initial-fact)** توسط خود نرم افزار استفاده می شود و **Fact** ها از شماره ۱ به بعد شماره گذاری می شوند.

حذف fact با دستور Retract

با استفاده از این دستور می توان یک **Fact** را حذف نمود. باید دقت نمود که این دستور با داشتن شماره ایندکس **Fact** می تواند آنرا حذف نماید که از روش قبل شماره ایندکس را بدست می آوریم.

(**retract** <**retract-specifier**>+ | *)

<**retract-specifier**>: شماره ایندکس **Fact** ای را که قرار است حذف نماییم، مشخص می کنیم.

دستور Save-Fact جهت ذخیره Fact ها

با استفاده از این دستور می توان کلیه **Fact** های موجود (**Fact** ها در پنجره **Fact**) را در داخل فایل ذخیره نمود.

دستور Load-Facts جهت باز نمودن Fact ها

با استفاده از این دستور می توان **Fact** هایی که قبلا توسط دستور **Save-Facts** در فایل ذخیره نموده ایم را به لیست **Fact-list** اضافه نماییم.

تعریف قواعد (Rules)

با این دستور می توان قواعد را تعریف نماییم. نرم افزار کلیپس یک برنامه **Rule-based** نیز می باشد یعنی بر مبنای قواعد نیز کار می کند.

(**defrule** <**rule-name**> [<**optional comment**>]

<**patterns**>*

=>

<**actions**>*

)

<**rule-name**>: این قسمت شامل نام قاعده یا **Rule** می باشد.

<optional comment>: در این قسمت می توان توضیحاتی در مورد آن قاعده نوشت. توضیحات اختیاری می باشد.

<patterns>: این قسمت الگو را مشخص می نماید یعنی در واقع همان قسمت الگو عبارت شرطی می باشد.

<actions>: این قسمت اعمال و دستوراتی می باشد که در صورت برقرار شدن (True) قسمت **Pattern** اجرا می شود. نکته: علامت * در **Syntax** ها به معنی تعداد ۰ یا بیشتر می باشد * <patterns> یعنی چندین الگو می توان تعریف نمود.

(Defrule Rule1

(car(color red))

=>

(assert(Action ok))

)

یعنی اگر **fact** با نام **car** مقدار فیلد **Color** آن **Red** بود سپس یک **Fact** با دستور **Assert** به نام **Action** ایجاد کن و مقدار آنرا **ok** قرار بده.

نکته: علامت => معادل کلمه کلیدی **Then** در شبه کد می باشد

ایجاد **Fact** گروهی با دستور **Deffacts**

با استفاده از این دستور می توان **Fact** هایی بصورت گروهی تعریف کرد و به عنوان دانش اولیه با پایگاه دانش اضافه می شود.

(deffacts <deffacts-name> [<comment>]

<pattern>*)

<deffacts-name>: نام **fact** را مشخص می کند.

نکته: **Fact** هایی که با دستور **Deffact** تعریف می شوند در پنجره **Facts** نمایش داده نمی شوند.

[<comment>]: توضیحاتی برای **Fact** می باشد که نوشتن این توضیحات اختیاری می باشد.

<pattern>: نام الگو یا همان **Fact** ها را مشخص مینماید.

(deffacts BaseFact1

(Color Red)

(type Stuff)

(Height 100)

)

مثال فوق **Fact** با نام **_aseFact1** شامل فیلدهای اولیه **Color,type,Height** را به عنوان دانش اولیه به پایگاه داده اضافه می نماید.

چاپ رشته / متغیر با دستور Printout

(printout <logical-name> <print-items>*)

<logical-name>: در این قسمت می توان نوع خروجی را مشخص نمود با کاراکتر **(Terminal)**

tمانیتور به عنوان خروجی در نظر گرفته می شود.

<print-items>: در این قسمت متن مورد نظر با آیتم هایی که قرار است چاپ شوند را مشخص می نماییم.

نکته: عبارت **crLf** تعیین می کند که بعد از چاپ نمودن آیتم ها مکان نما به خط بعدی رود. این کلمه معادل **\n** در زبان **c++** میباشد.

نمایش لیست دستورات

با استفاده از دستورات زیر می توان لیست بعضی از ساختارهایی که تا بحال یاد گرفته ایم را مشاهده نماییم.

(**list-defrules**): این دستور لیست قواعد موجود را نمایش می دهد.

(list-deftemplates): این دستور لیست الگوهای موجود را نمایش می دهد.

(list-deffacts): این دستور لیست **Fact** هایی را که با دستور **Deffacts** تعریف کرده ایم را نمایش می دهد.

(Facts): این دستورات لیست تمام **fact(Fact)** های موجود در پنجره **Facts** که با دستورات **assert** یا **Deftemplate** اضافه نموده ایم را نمایش می دهد.

دستورات اساسی

(clear): این دستور تمام **Fact** ها و **Rule** ها را از حافظه کاری حذف می کند معادل این است یکبار از نرم افزار خارج شویم و دوباره وارد شویم (**Shutdown + Restart**).

(reset): این دستور اطلاعات مربوط به **Fact** ها را حذف میکند و **Agenda** را ریست می کند.

(Agenda): قواعد فعال را نمایش می دهد.

(Refresh): این دستور قواعد را بروز رسانی می نماید.

(Run [<Limit>]): با این دستور قواعد فعال در حافظه کاری اجرا می شود که در قسمت **<Limit>** تعداد قواعدی را که می خواهیم اجرا شود را مشخص می کنیم. مشخص نمودن تعداد اختیاری می باشد. در صورتی که تعداد را مشخص ننماییم یعنی (**Run**) در این صورت کلیه دستورات فعال در پنجره **Agenda** اجرا می شوند.

بعضی توابع سودمند

تابع save

با این تابع مجموعه از ساختارهایی که وجود دارند (**Fact** و قواعد و الگوها و ...) را در فایل ذخیره می نماید.

(save <file-name>)

<file-name>: مسیر و نام فایل به همراه پسوند آن را مشخص می نماید.

تابع Load

توسط این دستور می توان ساختار هایی که قبلا ذخیره نموده ایم (Save) را باز نماید و آماده اجرا کند.

(load <file-name>)

<file-name>: مسیر و نام فایل به همراه پسوند آن را مشخص می نماید.

تابع Open

با استفاده از این تابع می توان فایل برنامه کلیپس که قبلا ذخیره نموده ایم را باز نماییم.

(open <file-name> <logical-name> [<mode>])

<file-name>: این قسمت شامل نام فایل به همراه آدرس کامل و پسوند فایل می باشد. حتما اگر در مسیر علامت بک اسش وجود دارد باید آنرا مشخص نماییم.

نکته: برای مشخص نمودن علامت \ باید این علامت را ۲ بار پشت سر هم تایپ نماییم تا برای کلیپس معنی \ این بدهد یعنی \ \ را تایپ می کنیم.

<logical-name>: این قسمت یک نام منطقی می باشد که قبلا در برنامه جاری نباید استفاده شده باشد.

[<mode>]: این قسمت نحوه دستیابی به فایل را مشخص می نماید. این قسمت اختیاری می باشد.

انواع حالت های Mode

"r" read access only

"w" write access only

"r+" read and write access

"a" append access only

در صورتی که تابع **Open** با موفقیت اجرا شود مقدار **True** و در غیر اینصورت مقدار **False** را بر می گرداند.

تابع Close

این تابع بر عکس تابع **Open** کار میکند و فایلی را که قبلاً با تابع **Open** باز شده است را می بندد.

(close [<logical-name>])

- **[<logical-name>]**: نام منطقی فایل است که هنگام **Open** کردن به آن اختصاص دادیم.
- اگر تابع **Close** بدون پارامتر فراخوانی شود تمام فایل های باز را می بندد.
- در صورتی که تابع **Open** با موفقیت اجرا شود مقدار **True** و در غیر اینصورت مقدار **False** را بر می گرداند.

تابع Read

با استفاده از این تابع می توان مقداری را از صفحه کلید دریافت نمود یا محتویات یک فایل را خواند.

(read [<logical-name>])

[<logical-name>]: نام منطقی فایل است که هنگام **Open** کردن به آن اختصاص دادیم. این پارامتر اختیاری است و اگر فایل را مشخص نکنیم مقداری را از صفحه کلید دریافت می کند.

تابع Bind

با استفاده از این تابع یک مقدار را به یک متغیر منتقل می نماییم.

(bind <variable> <expression>*)

<variable>: نام یک متغیر را مشخص می نماید.

<expression>: یک عبارت را که قرار است در متغیر قرار گیرد مشخص می نماید.

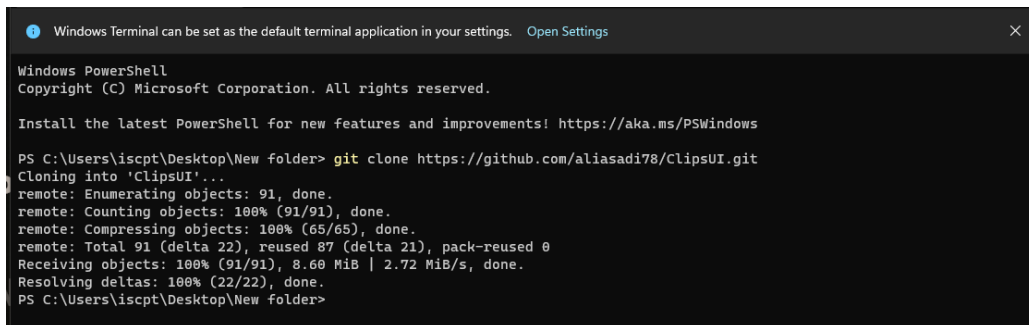
۲. راه اندازی برنامه

برنامه کلیپس را از آدرس زیر دانلود کنید

<https://www.clipsrules.net>

ابتدا با استفاده از دستور زیر پروژه را دانلود می کنیم.

`git clone https://github.com/aliasadi78/ClipsUI.git`



```
Windows Terminal can be set as the default terminal application in your settings. Open Settings X
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\iscpt\Desktop\New folder> git clone https://github.com/aliasadi78/ClipsUI.git
Cloning into 'ClipsUI'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (65/65), done.
remote: Total 91 (delta 22), reused 87 (delta 21), pack-reused 0
Receiving objects: 100% (91/91), 8.60 MiB | 2.72 MiB/s, done.
Resolving deltas: 100% (22/22), done.
PS C:\Users\iscpt\Desktop\New folder>
```

با دستور `cd CLIPSUI` وارد پوشه می شویم

برای اجرا پروژه نیاز به نصب Node js داریم اگر روی سیستم نصب دارید با دستور زیر نیازمندی های پروژه را نصب می کنیم.

`npm install`

```

PS C:\Users\iscpt\Desktop\clipsy\CLIPS_UI> npm install
npm WARN deprecated @babel/plugin-proposal-private-methods@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-private-methods instead.
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1568 packages, and audited 1569 packages in 22s

254 packages are looking for funding
  run 'npm fund' for details

6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.
PS C:\Users\iscpt\Desktop\clipsy\CLIPS_UI>

```

و با دستور زیر برنامه رو اجرا می کنیم.

npm start

برنامه در آدرس localhost:3000 اجرا می شود.



۳. نحوه کار برنامه

برنامه از ۳ قسمت اصلی تشکیل شده است:

قسمت Fact

قسمت Template

قسمت Rule

The screenshot displays the application's user interface, which is organized into three primary functional areas, each with its own form and submission button:

- Fact Section:** Located at the top, it features a text input field containing the word "fact" and a green button labeled "SUBMIT FACT".
- Template Section:** Positioned on the bottom left, it includes three input fields for "templateName", "templateDescription", and "templateDetailNumber", followed by a wide green button labeled "SUBMIT TEMPLATE".
- Rule Section:** Positioned on the bottom right, it contains three input fields for "templateName", "templateDescription", and "ruleDetailNumber" in the top row, and three more for "operation", "operationName", and "question" in the bottom row, with a wide green button labeled "SUBMIT RULE" below them.

Navigation and action buttons are placed around these sections:

- An "ADD FACTS" button is centered between the Fact and Template sections.
- An "ADD TEMPLATE" button is centered below the Template section.
- An "ADD RULE" button is centered below the Rule section.
- A "GENERATE CODE" button is centered at the very bottom of the interface.

برای نوشتن حقایق حقیقت را نوشته و دکمه submit fact را می زنیم تا حقیقت ثبت شود و برای اضافه کردن حقایق بیشتر دکمه add fact را می زنیم.

Template از ۳ قسمت نام تمپلیت، توضیح تمپلیت و تعداد جزئیات تشکیل شده است. تعداد جزئیات را وارد می کنیم

قسمت اول تایپ مقدار را مشخص می کنیم که Slot است یا MultiSlot

قسمت دوم نام را وارد می کنیم

قسمت سوم هم تایپ داده را وارد می کنیم که مقادیر معتبر آن عبارتند از:

INTEGER
FLOAT
STRING
SYMBOL
MULTIFIELD
FACT_ADDRESS
INSTANCE_NAME
INSTANCE_ADDRESS
EXTERNAL_ADDRESS

مانند قسمت قبل تمپلیت را ثبت و اضافه می کنیم.

Template

templateName

person

templateDescription

templateDetailNumber

3

typeOfValue-0

slot

detailName-0

name

detailType-0

STRING

typeOfValue-1

slot

detailName-1

surname

detailType-1

STRING

typeOfValue-2

slot

detailName-2

birthdate

detailType-2

SYMBOL

SUBMIT TEMPLATE

Template

templateName

templateDescription

templateDetailNumber

SUBMIT TEMPLATE

ADD TEMPLATE

Rule

templateName

templateDescription

ruleDetailNumber

operation

operationName

question

SUBMIT RULE

ADD RULE

GENERATE CODE

Rule از ۶ قسمت نام قاعده، توضیح قاعده و تعداد الگوها، عملیات بعد برقراری الگوها و نام عملیات و قسمت پرسش سوال برای گرفتن ورودی تشکیل شده است. تعداد الگوها و خود الگوها را وارد می کنیم.

Template

templateName

person

templateDescription

templateDetailNumber

3

typeOfValue-0

slot

detailName-0

name

detailType-0

STRING

typeOfValue-1

slot

detailName-1

surname

detailType-1

STRING

typeOfValue-2

slot

detailName-2

birthdate

detailType-2

SYMBOL

SUBMIT TEMPLATE

Template

templateName

templateDescription

templateDetailNumber

SUBMIT TEMPLATE

ADD TEMPLATE

Rule

templateName

my-rule

templateDescription

ruleDetailNumber

1

pattern-0

my-fact first-slot

operation

t "My Rule fired!" crlf

operationName

print

question

next rule?

SUBMIT RULE

ADD RULE

GENERATE CODE

و در آخر دکمه Generate Code را می زنیم تا خروجی را مشاهده کنیم.

۴. پیاده سازی یک سناریو

سیستم ساده پزشکی

فرض کنید با مطالعه و پژوهش در مورد یک سیستم ساده پزشکی و یا مصاحبه با پزشکان (منظور افراد خبره در این زمینه) به اطلاعات زیر دست یافتید:

پزشک برای تشخیص بیماری فرد مراجعه کننده، ابتدا از او چند سوال پرسیده و با توجه به پاسخ های فرد بیمار (که همان حقایق می باشد)، اقدام به تشخیص بیماری می کند.

از جمله این سوال ها این است که “محل درد شما کجاست؟”

مثلا فرد در جواب می تواند بگوید “شکم”، “گلو”، “سینه” و “سایر جاها”. حال با توجه به جوابی که در قبال این سوال داده می شود (حقیقت تولید شده) مرحله بعدی شروع می شود.

برای هر جواب در این مرحله، مراحل بعدی می تواند متفاوت باشد. مثلا اگر محل درد “شکم” انتخاب شود، تشخیص داده شود که فرد “آپاندیس” دارد

یا اگر محل درد “گلو” انتخاب شود در مرحله بعدی از فرد پرسیده شود که آیا “تب” دارد؟ اگر در جواب این سوال بگوید “بله”، تشخیص داده شود که بیماری “گلودرد باکتریایی” است و اگر “خیر”، تشخیص داده شود که فرد “گلودرد ویروسی” دارد

اگر محل درد “سینه” انتخاب شود، تشخیص داده شود که فرد “سکته” کرده است.

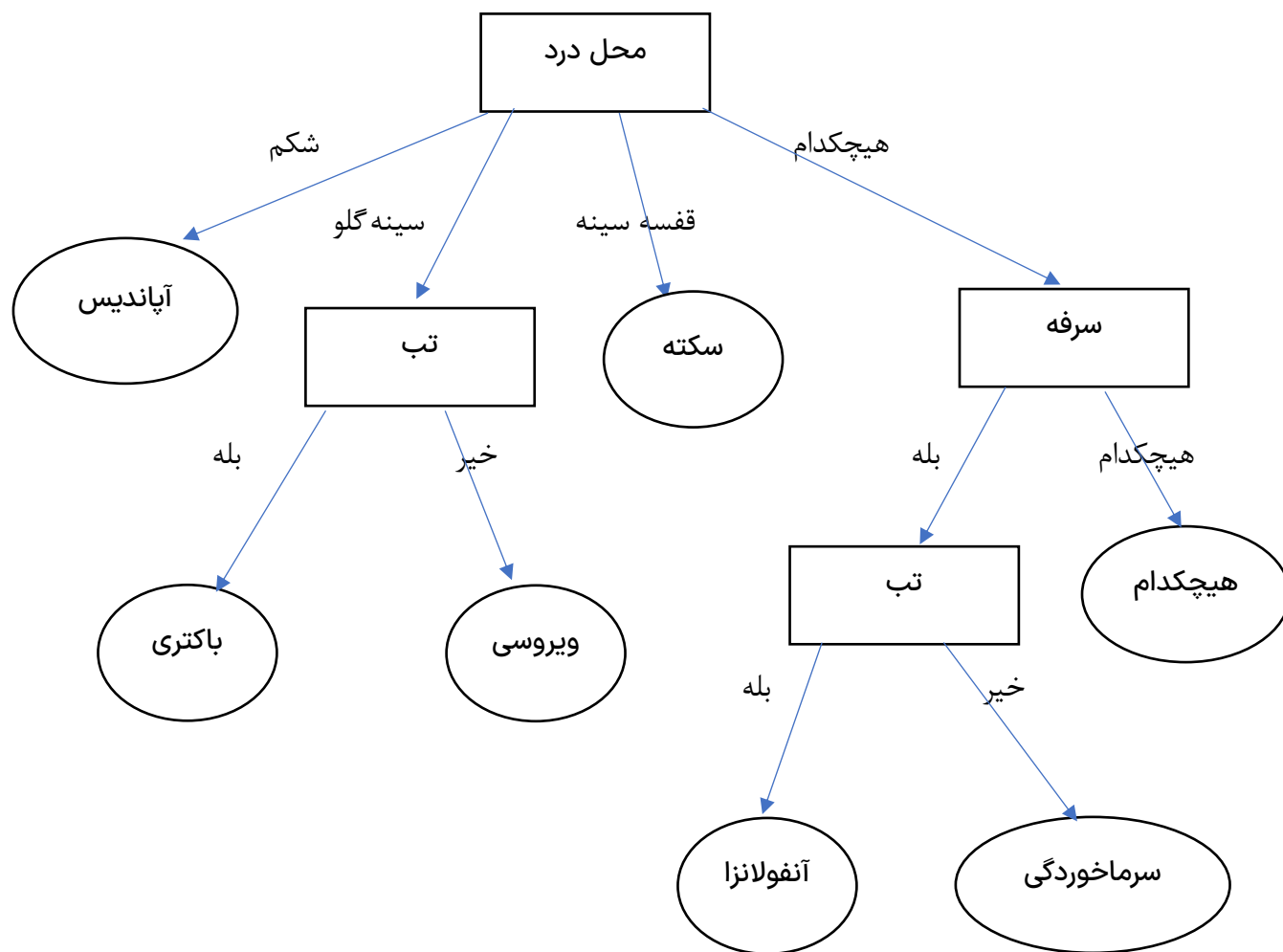
یا اگر محل درد “سایر” انتخاب شود در مرحله بعدی از فرد پرسیده شود که آیا “سرفه” می کند؟ از در جواب این سوال بگوید “خیر”، این سیستم ساده نتوانسته بیماری فرد را تشخیص دهد ولی اگر “بله” انتخاب شود در مرحله بعدی از فرد پرسیده شود که آیا “تب” دارد؟ اگر در جواب این سوال بگوید “بله”، تشخیص داده شود که بیماری “آنفلانزا” است و اگر “خیر”، تشخیص داده شود که فرد دچار “سرماخوردگی” شده است.

نکته: در هر مرحله هر سوال را یک قانون تصور نمود که با توجه به حقایق بدست آمده در مرحله قبلی اجرا

می‌شود.

درخت تصمیم گیری

شما می‌توانید از آنچه به عنوان اطلاعات در بخش قبلی به دست آورده اید یک درخت تصمیم گیری ایجاد کنید. البته توجه کنید این درخت سلسله مراتبی است و از ریشه به سمت برگ‌ها به جلو می‌رود. بنابراین در بعضی سیستم‌ها ترتیب این مراحل می‌تواند در تصمیم‌گیری بسیار مهم باشد.



حالا قواعد را مانند شکل وارد برنامه می‌کنیم.

Menu

=>

t crlf crlf crlf

" mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid" crlf crlf

" 1.) shekam. " crlf crlf

" 2.) galo." crlf crlf

" 3.) sineh." crlf crlf

" 4.) sayer." crlf crlf

" 5.) EXIT OF SYSTEM.." crlf crlf crlf

" Your answer: "

(selectedindex)

A-shekam

selectedindex 1

=>

t crlf crlf crlf" shoma bimary Apandis darid " crlf

crlf " Thank you for using my Program...

"crlf crlf

Rule

templateName

Menu

templateDescription

ruleDetailNumber

0

operation

t crlf crlf crlf " mahal dard ki

operationName

print

question

selectedindex

SUBMIT RULE

Rule

templateName

A-shekam

templateDescription

ruleDetailNumber

1

pattern-0

selectedindex 1

operation

t crlf crlf crlf" shoma bimary

operationName

print

question

SUBMIT RULE

Q-galo-tab

selectedindex 2

=>

t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: "

ifYesNoTab

A-galo-tab-yes

(selectedindex 2)

(ifYesNoTab yes)

=>

t crlf crlf crlf "shoma glodard bakterraei darid " crlf crlf

Rule

templateName

Q-galo-tab

templateDescription

ruleDetailNumber

1

pattern-0

selectedindex 2

operation

t crlf crlf crlf " aya shoma ta

operationName

print

question

ifYesNoTab

SUBMIT RULE

Rule

templateName

A-galo-tab-yes

templateDescription

ruleDetailNumber

2

pattern-0

selectedindex 2

pattern-1

ifYesNoTab yes

operation

t crlf crlf crlf "shoma glodard

operationName

print

question

SUBMIT RULE

A-galo-tab-no

(selectedindex 2)

(ifYesNoTab no)

=>

t crlf crlf crlf "shoma glodard virosi darid " crlf crlf

A-sineh

(selectedIndex 3)

=>

t crlf crlf crlf" shoma sekteh kardeid " crlf crlf " Thank you for using my Program... "crlf crlf

The image displays two screenshots of a 'Rule' configuration interface. Each form contains the following fields:

- templateName**: A-galo-tab-no (top) / A-sineh (bottom)
- templateDescription**: (empty)
- ruleDetailNumber**: 2 (top) / 1 (bottom)
- pattern-0**: selectedIndex 2 (top) / selectedIndex 3 (bottom)
- pattern-1**: ifYesNoTab no (top) / (empty) (bottom)
- operation**: t crlf crlf crlf "shoma glodard virosi darid " (top) / t crlf crlf crlf" shoma sekteh kardeid " (bottom)
- operationName**: print (both)
- question**: (empty)

Each form has a green **SUBMIT RULE** button at the bottom.

Q-sayer-solfeh

(selectedIndex 4)

=>

t crlf crlf crlf " aya shoma solfe mikonid? (Yes | No) " crlf crlf " Your answer: "
(ifYesNoSolfeh)

Q-sayer-solfeh-yes-tab ""

(selectedIndex 4)

(ifYesNoSolfeh yes)

=>

t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: "

(ifYesNoTab)

Rule		
templateName Q-sayer-solfeh	templateDescription	ruleDetailNumber 1
pattern-0 selectedIndex 4		
operation t crlf crlf crlf " aya shoma sc	operationName print	question ifYesNoSolfeh
SUBMIT RULE		

Rule		
templateName Q-sayer-solfeh-yes-tab	templateDescription	ruleDetailNumber 2
pattern-0 selectedIndex 4		
pattern-1 ifYesNoSolfeh yes		
operation t crlf crlf crlf " aya shoma ta	operationName print	question ifYesNoTab
SUBMIT RULE		

A-sayer-solfeh-yes-tab-yes ""

(selectedIndex 4)

(ifYesNoSolfeh yes)

(ifYesNoTab yes)

=>

t crlf crlf crlf" shoma anfolanza darid " crlf crlf " Thank you for using my
Program... "crlf crlf

Rule

templateName A-sayer-solfeh-yes-tab-yes	templateDescription	ruleDetailNumber 3
pattern-0 selectedIndex 4		
pattern-1 ifYesNoSolfeh yes		
pattern-2 ifYesNoTab yes		
operation t crlf crlf crlf" shoma anfolanza darid	operationName print	question

SUBMIT RULE

A-sayer-solfeh-yes-tab-no ""

(selectedIndex 4)

(ifYesNoSolfeh yes)

(ifYesNoTab no)

=>

t crlf crlf crlf" shoma sarma khordid" crlf crlf " Thank you for using my Program...
"crlf crlf

Rule

templateName A-sayer-solfeh-yes-tab-no	templateDescription	ruleDetailNumber 3
pattern-0 selectedindex 4		
pattern-1 ifYesNoSolfeh yes		
pattern-2 ifYesNoTab no		
operation t crlf crlf crlf" shoma sarma	operationName print	question

SUBMIT RULE

A-sayer-solfeh-no ""

(selectedindex 4)

(ifYesNoSolfeh no)

=>

t " moteasefane nmitavanam bimari shoma ra tashkhis daham" crlf crlf " Thank
you for using my Program... "crlf crlf

Rule

templateName

A-sayer-solfeh-no

templateDescription

ruleDetailNumber

2

pattern-0

selectedindex 4

pattern-1

ifYesNoSolfeh no

operation

t " moteasefane nmitavana

operationName

print

question

SUBMIT RULE

و نتیجه دو کد به زبان های پایتون و کلیپس خروجی می دهد که باید آن ها را اجرا کنیم.

```

(defrule Menu ""
  =>
  (printout t crlf crlf crlf " mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid" crlf crlf " 1.) shekam. " crlf crlf " 2.) galo." crlf crlf " 3.) sineh." crlf crlf " 4.) sayer." crlf crlf " 5.) EXIT OF SYSTEM.." crlf crlf crlf " Your answer: ")
  (assert ( selectedindex (read))))
  (defrule A-shekan ""
    (selectedindex 1)
    =>
    (printout t crlf crlf crlf "shoma binary Apanidis darid " crlf crlf " Thank you for using my Program..."crlf crlf )
    )
  (defrule A-shekan ""
    (selectedindex 1)
    =>
    (printout t crlf crlf crlf "shoma binary Apanidis darid " crlf crlf " Thank you for using my Program..."crlf crlf )
    )
  (defrule Q-galo-tab ""
    (selectedindex 2)
    =>
    (printout t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: " )
    (assert ( ifVeshoTab (read))))
    (defrule A-galo-tab-yes ""
      (selectedindex 2)
      (ifVeshoTab yes)
      =>
      (printout t crlf crlf crlf "shoma glodard bakteraei darid " crlf crlf )
    )
    (defrule A-galo-tab-no ""
      (selectedindex 2)
      (ifVeshoTab no)
      =>
      (printout t crlf crlf crlf "shoma glodard virosi darid " crlf crlf )
    )
    (defrule A-sineh ""
      (selectedindex 3)
      =>
      (printout t crlf crlf crlf "shoma sekteh kardedid " crlf crlf " Thank you for using my Program..."crlf crlf )
    )
    (defrule Q-sayer-solfeh ""
      (selectedindex 4)
      =>
      (printout t crlf crlf crlf " aya shoma solfe mikonid? (Yes | No) " crlf crlf " Your answer: ")
      (assert ( ifVeshoSolfeh (read))))
      (defrule A-sayer-solfeh-yes-tab ""
        (selectedindex 4)
        (ifVeshoSolfeh yes)
        =>
        (printout t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: ")
        (assert ( ifVeshoTab (read))))
        (defrule A-sayer-solfeh-yes-tab-yes ""
          (selectedindex 4)
          (ifVeshoSolfeh yes)
          (ifVeshoTab yes)
          =>
          (printout t crlf crlf crlf "shoma anfolanza darid " crlf crlf " Thank you for using my Program..."crlf crlf )
        )
        (defrule A-sayer-solfeh-yes-tab-no ""
          (selectedindex 4)

```

```

import clips
environment = clips.Environment()
DEFRULE_STRING0= "" (defrule Menu ""
  =>
  (printout t crlf crlf crlf " mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid" crlf crlf " 1.) shekam. " crlf crlf " 2.) galo." crlf crlf " 3.) sineh." crlf crlf " 4.) sayer." crlf crlf " 5.) EXIT OF SYSTEM.." crlf crlf crlf " Your answer: ")
  (assert ( selectedindex (read))))""
environment.build(DEFRULE_STRING0)
DEFRULE_STRING1= "" (defrule A-shekan ""
  (selectedindex 1)
  =>
  (printout t crlf crlf crlf "shoma binary Apanidis darid " crlf crlf " Thank you for using my Program..."crlf crlf )
  )""
environment.build(DEFRULE_STRING1)
DEFRULE_STRING2= "" (defrule A-shekan ""
  (selectedindex 1)
  =>
  (printout t crlf crlf crlf "shoma binary Apanidis darid " crlf crlf " Thank you for using my Program..."crlf crlf )
  )""
environment.build(DEFRULE_STRING2)
DEFRULE_STRING3= "" (defrule Q-galo-tab ""
  (selectedindex 2)
  =>
  (printout t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: " )
  (assert ( ifVeshoTab (read))))""
environment.build(DEFRULE_STRING3)
DEFRULE_STRING4= "" (defrule A-galo-tab-yes ""
  (selectedindex 2)
  (ifVeshoTab yes)
  =>
  (printout t crlf crlf crlf "shoma glodard bakteraei darid " crlf crlf )
  )""
environment.build(DEFRULE_STRING4)
DEFRULE_STRING5= "" (defrule A-galo-tab-no ""
  (selectedindex 2)
  (ifVeshoTab no)
  =>
  (printout t crlf crlf crlf "shoma glodard virosi darid " crlf crlf )
  )""
environment.build(DEFRULE_STRING5)
DEFRULE_STRING6= "" (defrule A-sineh ""
  (selectedindex 3)
  =>
  (printout t crlf crlf crlf "shoma sekteh kardedid " crlf crlf " Thank you for using my Program..."crlf crlf )
  )""
environment.build(DEFRULE_STRING6)
DEFRULE_STRING7= "" (defrule Q-sayer-solfeh ""
  (selectedindex 4)
  =>
  (printout t crlf crlf crlf " aya shoma solfe mikonid? (Yes | No) " crlf crlf " Your answer: ")
  (assert ( ifVeshoSolfeh (read))))""
environment.build(DEFRULE_STRING7)
DEFRULE_STRING8= "" (defrule A-sayer-solfeh-yes-tab ""
  (selectedindex 4)
  (ifVeshoSolfeh yes)
  =>
  (printout t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: ")
  (assert ( ifVeshoTab (read))))""
environment.build(DEFRULE_STRING8)
DEFRULE_STRING9= "" (defrule A-sayer-solfeh-yes-tab-yes ""
  (selectedindex 4)
  (ifVeshoSolfeh yes)
  (ifVeshoTab yes)
  =>
  (printout t crlf crlf crlf "shoma anfolanza darid " crlf crlf " Thank you for using my Program..."crlf crlf )
  )
  (defrule A-sayer-solfeh-yes-tab-no ""
    (selectedindex 4)

```

```

import clips
environment = clips.Environment()
DEFRULE_STRING0= "" (defrule Menu ""
=>
(printout t crlf crlf crlf " mahal dard kojast?lotfan ykl az gozinehaye zir ra entekhab konid" crlf crlf " 1.) shekam. " crlf crlf " 2.) galo." crlf crlf " 3.) sineh." crlf crlf " 4.)
sayer." crlf crlf " 5.) EXIT OF SYSTEM.." crlf crlf crlf " Your answer: ")
(assert ( selectedindex (read))))""
environment.build(DEFRULE_STRING0)
DEFRULE_STRING1= "" (defrule A-shekam ""
(selectedindex 1)
=>
(printout t crlf crlf crlf " shoma bimary Apandis darid " crlf crlf " Thank you for using my Program... "crlf crlf )
)""
environment.build(DEFRULE_STRING1)
DEFRULE_STRING2= "" (defrule A-shekam ""
(selectedindex 1)
=>
(printout t crlf crlf crlf " shoma bimary Apandis darid " crlf crlf " Thank you for using my Program... "crlf crlf )
)""
environment.build(DEFRULE_STRING2)
DEFRULE_STRING3= "" (defrule Q-galo-tab ""
(selectedindex 2)
=>
(printout t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: ")
(assert ( ifYesNoTab (read))))""
environment.build(DEFRULE_STRING3)
DEFRULE_STRING4= "" (defrule A-galo-tab-yes ""
(selectedindex 2)
(ifYesNoTab yes)
=>
(printout t crlf crlf crlf "shoma glodard bakteriaei darid " crlf crlf)
)""
environment.build(DEFRULE_STRING4)
DEFRULE_STRING5= "" (defrule A-galo-tab-no ""
(selectedindex 2)
(ifYesNoTab no)
=>
(printout t crlf crlf crlf "shoma glodard virosi darid " crlf crlf)
)""
environment.build(DEFRULE_STRING5)
DEFRULE_STRING6= "" (defrule A-sineh ""
(selectedindex 3)
=>
(printout t crlf crlf crlf " shoma sekteh kardeid " crlf crlf " Thank you for using my Program... "crlf crlf)
)""
environment.build(DEFRULE_STRING6)
DEFRULE_STRING7= "" (defrule Q-sayer-solfeh ""
=>
(printout t crlf crlf crlf "shoma glodard virosi darid " crlf crlf)
)""
environment.build(DEFRULE_STRING5)
DEFRULE_STRING6= "" (defrule A-sineh ""
(selectedindex 3)
=>
(printout t crlf crlf crlf " shoma sekteh kardeid " crlf crlf " Thank you for using my Program... "crlf crlf)
)""
environment.build(DEFRULE_STRING6)
DEFRULE_STRING7= "" (defrule Q-sayer-solfeh ""
(selectedindex 4)
=>
(printout t crlf crlf crlf " aya shoma solfe mikonid? (Yes | No) " crlf crlf " Your answer: ")
(assert ( ifYesNoSolfeh (read))))""
environment.build(DEFRULE_STRING7)
DEFRULE_STRING8= "" (defrule Q-sayer-solfeh-yes-tab ""
(selectedindex 4)
(ifYesNoSolfeh yes)
=>
(printout t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: ")
(assert ( ifYesNoTab (read))))""
environment.build(DEFRULE_STRING8)
DEFRULE_STRING9= "" (defrule A-sayer-solfeh-yes-tab-yes ""
(selectedindex 4)
(ifYesNoSolfeh yes)
(ifYesNoTab yes)
=>
(printout t crlf crlf crlf" shoma anfolanza darid " crlf crlf " Thank you for using my Program... "crlf crlf)
)""
environment.build(DEFRULE_STRING9)
DEFRULE_STRING10= "" (defrule A-sayer-solfeh-yes-tab-no ""
(selectedindex 4)
(ifYesNoSolfeh yes)
(ifYesNoTab no)
=>
(printout t crlf crlf crlf" shoma sarma khordid" crlf crlf " Thank you for using my Program... "crlf crlf)
)""
environment.build(DEFRULE_STRING10)
DEFRULE_STRING11= "" (defrule A-sayer-solfeh-no ""
(selectedindex 4)
(ifYesNoSolfeh no)
=>
(printout t " moteasefane nmitavanam bimari shoma ra tashkhis daham" crlf crlf " Thank you for using my Program... "crlf crlf)
)""
environment.build(DEFRULE_STRING11)
environment.run()

```

```
mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid
```

```
1.) shekam.
```

```
2.) galo.
```

```
3.) sineh.
```

```
4.) sayer.
```

```
5.) EXIT OF SYSTEM..
```

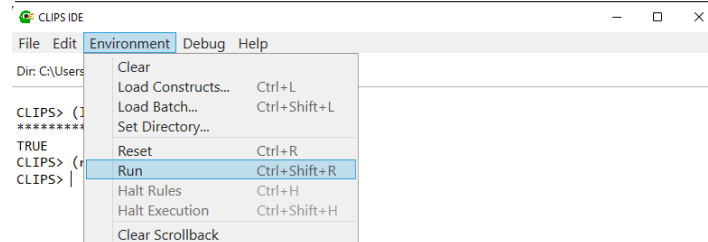
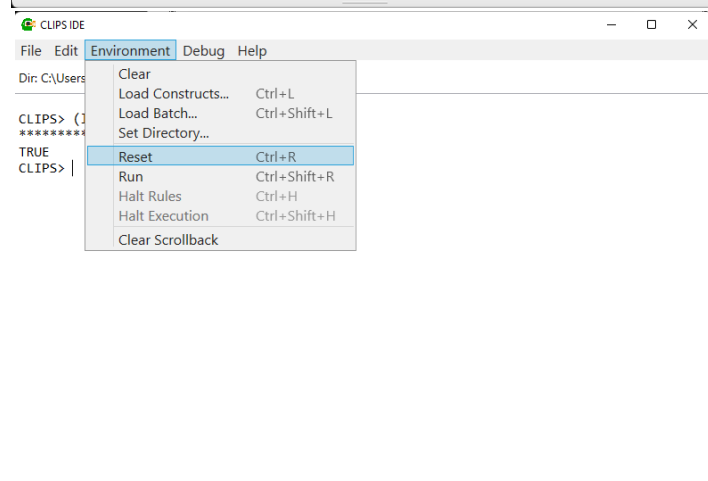
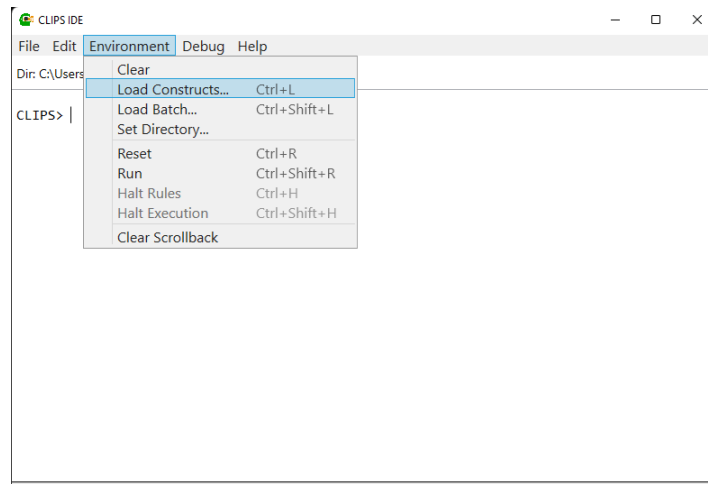
```
Your answer: 1
```

```
█
```

```
shoma bimary Apandis darid
```

```
Thank you for using my Program...
```

برای اجرای برنامه در کلیپ مانند شکل عمل می کنیم



```
CLIPS IDE
File Edit Environment Debug Help
Dir: C:\Users\iscpt\Desktop\clipspy

CLIPS (6.4.1 4/8/23)
CLIPS> (load "final.clp")
*****
TRUE
CLIPS> (reset)
CLIPS> (run)

mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid

1.) shekam.
2.) galo.
3.) sineh.
4.) sayer.
5.) EXIT OF SYSTEM..

Your answer:
```

```
CLIPS IDE
File Edit Environment Debug Help
Dir: C:\Users\iscpt\Desktop\clipspy

mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid

1.) shekam.
2.) galo.
3.) sineh.
4.) sayer.
5.) EXIT OF SYSTEM..

Your answer: 3

shoma sekteh kardeid

Thank you for using my Program...

CLIPS>
```


<https://www.clipsrules.net>

<https://clipspy.readthedocs.io/en/latest/>

<http://Parsbook.org>

<http://tehranit.net/آموزش-سریع-کلیپس-سیستم-خبره-۴-ساعت/>



Computer Engineering Department

Bachelor's final project

Design and implementation of graphic user interface for Clips software with
adaptability feature

Ali Asadi

Supervisor

Dr. Kangavari