

به نام خدا



# دانشگاه صنعتی شیراز

## پروژه درس طراحی کامپایلر ها

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

بهار ۱۴۰۰

مدرس: دکتر رضا اکبری

تهیه کنندگان: عباس مهربانیان، سروش احراری

## فهرست محتوا

۳	مقدمه و شرح پروژه
۴	فاز اول: طراحی تحلیلگر لغوی
۶	فاز دوم: طراحی تحلیلگر نحوی
۷	تحویل و ارائه پروژه
۷	تحویل فاز اول
۷	تحویل فاز دوم
۸	نمره دهی و ارزیابی پروژه

## مقدمه و شرح پروژه

قبل از شروع، لطفا در نظر داشته باشید که حجم این مستندات به معنی دشوار تر شدن و بیشتر شدن حجم پروژه نسبت به سال های پیشین نمیشود. این مستندات صرفا برای رفع ابهام شما و ارائه توضیحات دقیقتر درباره پروژه برای شما دانشجویان عزیز تدارک داده شده است.

هدف از این پروژه آشنایی بیشتر شما با مفاهیم درس طراحی کامپایلر میباشد.

این پروژه شامل دو فاز خواهد بود:

- طراحی اسکنر یا تحلیلگر لغوی (Lexical Analyzer)
- طراحی ساختاریاب یا تحلیلگر نحوی (Parser)

در این پروژه قصد داریم یک زبان ساده مشابه به جاوا (MiniJava) طراحی کنیم.

برای تست پروژه شما یک سری فایل های ورودی در پوشه inputs در نظر گرفته شده است که بتوانید به وسیله آنها عملکرد کامپایلر خود را بررسی نمایید. توجه داشته باشید همه نمونه فایل های ورودی موجود لزوما یک ورودی صحیح نیستند.

در توضیحات فازهای پروژه که در ادامه مطالعه خواهید کرد، منظور از ماژول، کلاس و یا تابعی است که کار مورد نظر (مثل: خواندن فایل ورودی) را در برنامه شما انجام میدهد.

این پروژه را میتوانید بصورت فردی و یا در قالب گروه های دو نفره انجام دهید. یک نام برای گروه خود انتخاب کرده و نام گروه به همراه لیست اعضا و شماره دانشجویی آنان را به ایمیل زیر ارسال کنید. در صورتی که بصورت فردی پروژه را انجام میدهید، نام گروه شما میتواند نام خانوادگی شما باشد. عنوان ایمیل خود را SutechCompiler1400-GName قرار دهید.

[abbas.mrbn@gmail.com](mailto:abbas.mrbn@gmail.com)

برای انجام این پروژه نیاز است که یک اکانت گیت هاب ایجاد کرده و پروژه خود را در یک repository به نام sutech-compiler-1400 آپلود و بروزرسانی نمایید.

هرگونه ابهام و یا مشکلات خود را میتوانید در قسمت مشکلات (issues) در repository پروژه درس به آدرس زیر مطرح کنید. مشارکت دانشجویان برای پاسخ گویی به سوالات دیگران مانعی ندارد. ترجیحا سوالات خود را در زبان انگلیسی مطرح کنید.

✓ <https://github.com/mehrbanian/sutech-compiler-1400>

مشابه بودن بیش از حد کدهای دو تیم تقلب محسوب میشود و در صورت شناسایی از نمرات هر دو تیم کسر خواهد شد. تیم ها بایستی پروژه ها را در تاریخی که اعلام خواهد شد بصورت شفاهی ارائه دهند. توجه داشته باشید تاریخچه repository گیت هاب شما به عنوان یک سند از فعالیت های اعضای گروه محسوب خواهد شد (**حجم بالای فعالیت مهم نیست**).

## فاز اول: طراحی تحلیلگر لغوی

**هدف:** پیاده سازی تحلیلگر لغوی (Scanner)

### توضیحات:

اسکنر باید با استفاده از ماژول خواندن فایل ورودی، فایل ورودی را بلوک به بلوک خوانده و سپس از داخل بافر کاراکتر به کاراکتر دستورات را خوانده و آنها را به عنوان ورودی به ماژول تحلیل لغوی دهید تا این ماژول توکن ها را تشخیص دهد.

در صورتی که توکن تشخیص داده شده یک شناسه باشد باید این شناسه به ماژول مدیریت جدول نماد داده شود تا به جدول نماد اضافه شود. در صورتی که توکن تشخیص داده نشود باید ماژول مدیریت خطا فراخوانی شود تا در صورت امکان خطا را رفع کند یا پیغام خطای مناسب را تولید کند.

برای پیاده سازی تحلیلگر لغوی می توانید از روش دستی یا خودکار استفاده کنید. همچنین استفاده از روش دستی در کنار روش خودکار برای برخی قسمت ها مجاز است.

در روش خودکار میتوانید از نرم افزارهایی مثل Lex، Jflex یا ... استفاده نمایید و با استفاده از عبارات منظم ابتدا فرم کلی توکن های خود را تعریف نمایید. استفاده از Regex نیز مجاز است.

در روش دستی میتوانید از امکانات زبان برنامه نویسی که خصوصیات یک دیاگرام انتقال (Transition Diagram) را به شما میدهند استفاده کنید. مثل: دستورات شرطی if ... else، switch case و ...

برای کامپایلر کوچک شما در این فاز بایستی موارد زیر در نظر گرفته شوند:

- تعریف متغیر به همراه نوع آنها
- حلقه های for و while
- دستور شرطی if-then-else
- عملگر انتساب
- عملگرهای پایه ریاضی
- جمع (+) و تفریق (-)
- ضرب (\*) و تقسیم (/)
- باقی مانده تقسیم (%)
- AND و OR منطقی (&& و ||)
- عملگرهای مقایسه
- مساوی(=) و نامساوی(!=)
- کوچکتر(<)، کوچکتر و مساوی(<=)، بزرگتر(>)، بزرگتر و مساوی(>=)
- عملگرهای حسابی
- عملگر افزایش(++) و کاهش(--)
- کامنت (بصورت تک خط و چند خطی)

## موارد اختیاری:

- تعریف کلاس (اختیاری)
- دستور switch-case (اختیاری)

کلمات کلیدی زبان شما شامل موارد زیر خواهد بود:

"for", "while", "if", "else", "then", "switch", "case", "break", "else if", "elseif", "static", "void", "public", "default", "continue", "return", "long", "int", "String", "char", "boolean", "double", "float", "class"

عبارات true و false را میتوانید به عنوان Boolean literal در نظر بگیرید.

در صورتی که نتوانستید نوع یک متغیر در برنامه ورودی را تشخیص دهید، میتوانید نوع آن را در جدول نماد ها، undefined قرار دهید.

یک نمونه از جدول نمادها:

Id	name	type	value
1	x	int	N/A
2	y	boolean	true

## مواردی که باید تحویل داده شوند:

فاز اول پروژه شما بایستی موارد زیر را شامل شود:

- ✓ مازول خواندن کد ورودی
- ✓ مازول تحلیل لغوی
- ✓ مازول مدیریت جدول نماد

**ورودی:** فایل حاوی دستورات برنامه

**خروجی:**

- لیست توکن ها
- جدول نماد حاوی شناسه ها و مشخصات آنها

## فاز دوم: طراحی تحلیلگر نحوی

**هدف:** پیاده سازی تحلیلگر نحوی (Parser)

### توضیحات:

در این فاز کامپایلر باید خروجی های مرحله تحلیل لغوی را دریافت نموده و دستورات برنامه را از نظر نحوی بررسی نماید. برای تحلیل نحوی باید از یکی از پارسرهای SLR ، CLR یا LALR استفاده کنید. در این مرحله در صورتیکه خطایی تشخیص داده شود باید مازول مدیریت خطا فراخوانی شود تا عمل مناسب را انجام دهد. از هر کدام از روشهای موجود برای مدیریت خطا می توانید استفاده کنید.

در این فاز بایستی مازول مدیریت جدول نماد (Symbol Table) نیز در صورت نیاز بطوری تغییر دهید که مازول مدیریت خطا از ورود مقادیر غیرمجاز به این جدول جلوگیری نماید.

استفاده از کتابخانه های پارسرهای موجود، برای مثال در زبان جاوا: YACC ، JavaCC و...، مانعی ندارد.

تحلیلگر نحوی شما بایستی طبق یک گرامر عمل parse را انجام دهد. گرامر مورد نظر گرامر زبان MiniJava میباشد که نمونه هایی از آن در سطح اینترنت موجود است. یک نمونه از این گرامر در فایل MiniJava.htm و همچنین در قسمت README، در ریو پروژه قرار گرفته است. میتوانید به عنوان نمونه از این گرامر برای پارسر خود استفاده کنید.

### مواردی که باید تحویل داده شوند:

فاز دوم پروژه شما بایستی موارد زیر را شامل شود:

- ✓ مازول مدیریت خطا
- ✓ مازول تحلیلگر نحوی
- ✓ مازول مدیریت جدول نماد

**ورودی:** خروجی های به دست آمده از تحلیلگر لغوی (Scanner)

### خروجی:

- مراحل پارس برای قطعه کد ورودی
- محتوای پشته در طی مراحل باید نشان داده شود
- شرح خطاها در صورت وجود
- جدول نماد

## تحويل و ارائه پروژه

همانطور که در قسمت مقدمه گفته شد پروژه خود را بایستی بر روی github قرار دهید و فایل نهایی خود را در سامانه مدل دانشگاه اپلود کنید.

دقت داشته باشید برای جلوگیری از به اشتراک گذاشته شدن کد، repository گیت هاب شما بایستی تا پایان ترم بصورت private باشد و آی دی های گیت هاب زیر را برای دسترسی TA ها به پروژه اضافه کنید:

- @merhbanian
- @soroushahrari

تاریخ هایی برای ارائه هر یک از فاز ها بر روی سامانه اعلام خواهد شد که میتوانید روزی که برای شما مناسب است را در نظرسنجی سامانه انتخاب کنید. (ظرفیت هر روز محدود خواهد بود)

### تحويل فاز اول

**مهلت تحويل:** ۱۵ خرداد ۱۴۰۰ ساعت ۲۳:۵۹

**تاریخ ارائه:** در سامانه اعلام خواهد شد

**نام و فرمت فایل:**

فایل نهایی این فاز بایستی یک فایل با فرمت rar یا zip حاوی کد پروژه + لینک repository پروژه شما بر روی گیت هاب باشد. نام فایل نهایی شما باید با فرمت زیر باشد:

[نام گروه][P1][SutechCompiler1400]

### تحويل فاز دوم

**مهلت تحويل:** ۱۸ تیر ۱۴۰۰ ساعت ۲۳:۵۹

**تاریخ ارائه:** در سامانه اعلام خواهد شد

**نام و فرمت فایل:**

فایل نهایی این فاز بایستی یک فایل با فرمت rar یا zip حاوی کد پروژه + لینک repository پروژه شما بر روی گیت هاب باشد. نام فایل نهایی شما باید با فرمت زیر باشد:

[نام گروه][P2][SutechCompiler1400]

توجه داشته باشید با توجه به مهلت های تحويل پروژه معین شده، امکان تمدید محدود میباشد.

## نمره‌دهی و ارزیابی پروژه

نمره دهی پروژه شما بصورت زیر محاسبه خواهد شد:

فاز اول	فاز دوم	
۱۵	۱۵	استفاده از گیت/گیت هاب
۴۰	۴۰	ارائه شفایی
۱۰	۱۰	رعایت نکات (ارائه و تحویل پروژه)
۳۵	۳۵	انجام پروژه (کد)
۱۰۰	۱۰۰	مجموع
۱	۱	ضریب تاثیر در نمره نهایی

نمرات گروه های تک نفره در ۱/۲۵ ضرب خواهد شد.

موارد زیر میتوانند به عنوان نمره مثبت محسوب شوند:

- ایجاد یک فایل Readme مناسب برای repository پروژه خود بر روی گیت هاب
- در صورتی که بصورت گروهی پروژه را انجام میدهید و تشخیص داده شود که در طول روند انجام پروژه مشارک گروهی خوبی داشته اید. (نمره کل ضرب در ۱/۲۰)

موفق و پیروز باشید.