

**MAC0316/5754**  
**Exercício Programa 3**  
Data de entrega: 10/12/2017

**Instruções:**

1. Você deve entregar seu programa pelo PACA em um **único arquivo .rkt** contendo as definições do interpretador.
2. **Programas atrasados terão uma penalidade de 20% na pontuação POR DIA.**

**Interpretador (10 pontos)**

Nesta parte vamos implementar uma versão da linguagem **Smalltalk**. As grandes modificações serão na **linguagem de entrada** e no **processo de resolução de métodos**. Utilizaremos resolução *dinâmica* de métodos mas usaremos um padrão “híbrido” onde inteiros e suas operações são primitivos.

Sintaxe da linguagem (descrita como uma gramática):

```
<input>      --> <expression> | <class-def>

<class-def>   --> (class <class-id> <class-id> <inst-var>
                  <method-def> <method-def>)
<class-id>    --> um identificador (para nome da classe)
<inst-var>    --> um identificador (para nome da variável de instância)

<method-def> --> (method <method-id> <method-param> <expression>)
<method-id>  --> um identificador (para nome do método)
<method-arg> --> um identificador (para nome do parâmetro do método)

<expression> --> <arith-exp> | <if-exp> | <new-exp> | <send-exp> |
                  <set-exp> | <value>
<arith-exp>   --> (<arith-op> <expression> <expression>)
<arith-op>    --> + | - | * | /
<if-exp>      --> (if <expression> <expression> <expression>)
<new-exp>     --> (new <class-id>)
<send-exp>    --> (send <expression> <method-id> <value>)
<set-exp>     --> (set! <inst-var> <value>) | (set! <method-arg> <value>)

<value>      --> <integer> | <object>
```

- a) **(2 pontos):** Acrescente o valor `classV` no interpretador. Implemente o comando `class` que possui 5 parâmetros: nome da classe, nome da classe pai, variável de instância, definição de método 1, e definição de método 2. Note que `class` deve registrar o nome do `classV` no ambiente para posterior criação de objetos.

- b) (2 pontos): Substitua o valor `closV` pelo valor `methodV`, que é semelhante aos fechamentos mas que possui um nome (para envio de mensagens). Note que `methodV` será componente de `classV` quando o usuário definir uma classe. Use esse fato para garantir que todos os métodos de uma dada classe compartilhem o mesmo fechamento (que inclui o atributo de instância da classe).
- c) (2 pontos): Acrescente o valor `objetcV` no interpretador e adicione uma classe pai-de-todos `Object` no ambiente global. Ou seja, quando o usuário for rodar `interpS`, ele deve poder usar `Object` em seu código. Isso é necessário porque o comando `class` exige que toda classe feita pelo usuário tenha uma classe pai. Assim todas as classes herdarão de `Object`.
- d) (2 pontos): Implemente o operador primitivo `new` que recebe como parâmetro uma `classV` e cria um `objetcV` daquela classe. Lembre-se que a criação de um objeto da classe filha leva a criação de objetos das suas classes ancestrais. Além disso, lembre que o objeto filho usará esses objetos para acessar atributos de instância e métodos das classes ancestrais. Finalmente, adicione um caso especial quando a classe analisada for `Object`, que não terá classe pai.
- e) (2 pontos): Substitua o operador primitivo `call` pelo `send`, que recebe como parâmetro um `objetcV`, um nome de método e um argumento e que executa o método correspondente com o argumento. Lembre-se de que, se o objeto da classe filha não conhecer a mensagem, ele deve repassá-lo para a classe pai. Além disso, lembre que se a classe filha tiver um método de mesmo nome que um da classe pai, o método da classe filha deve ter preferência. Finalmente, adicione um caso especial quando a classe analisada for `Object`, lançando um erro de “mensagem desconhecida”.

Note que toda classe possui 2 métodos. Isso permitirá testar a redefinição de métodos da classe pai.

**IMPORTANTE:** Não será dado **crédito parcial** aos itens acima. Apenas aqueles que funcionarem receberão pontos.

**IMPORTANTE:** Você deve usar a versão `plai` (**não** `plai-typed`) do interpretador com `closureTyped.rkt` (disponibilizado com este enunciado).