

SOOP

Neste curso iremos escrever um compilador para uma linguagem orientada a objetos simples chamada SOOP (Simple Object Oriented Programming language). SOOP permite definições e declarações de funções, bem como comandos tradicionais de atribuição e expressões, loops usando comandos **while**, comandos **if** e comandos **print**. Os tipos de dados suportados são **int**, **boolean**, **double**, **arrays**, e **class** (com herança). Escopos locais e globais também são suportados.

A linguagem de máquina a ser gerada ainda será definida, mas a princípio deve ser baseado em um conjunto de instruções similar ao MIPS.

Definições mais detalhadas de SOOP serão entregues ao longo do semestre, por hora alguns exemplos de programas simples escritos em SOOP são fornecidos.

1) Imprimindo “hello world” em SOOP:

```
void main(void) {  
    Print("hello world");           // a built-in Print function  
}
```

2) Cálculos usando SOOP:

```
int a;                               // a global  
  
void main(void) {  
    int b;  
    int a;  
    double d;  
  
    b = 3;  
    a = b + 2;  
    d = 0.3;  
    Print(a, b, d);  
}
```

3) Um loop simples em SOOP:

```
void main(void) {  
    int a;  
    a = 0;  
  
    while (a != 10) {  
        Print(a, " ");  
        a = a + 1;  
    }  
}
```

4) Uma chamada simples de função em SOOP:

```
int foo(int a, double b, bool c);    // we have a boolean type built in

int foo(int a, double b, bool c) {
    return a + 2;
}

void main(void) {
    int b;
    int a;
    double d;

    a = 3;
    d = 0.3;
    foo(a, d, TRUE);
}
```

5) Uma classe simples em SOOP:

```
class Cow {
    int height;
    int weight;
    void Moo(void) {
        Print ( this.height, " ", this.weight, "\n" );
    }
}

void main(void) {
    int x[8];

    class Cow betsy;
    betsy = New(Cow);
    betsy.weight = 100;
    betsy.height = 122;
    betsy.Moo();
}
```

6) Mais classes (com herança)

```
class Animal {
    int height;
    class Animal mother;
    void InitAnimal(int h, class Animal mom) {
        this.height = h;
        this.mother = mom;
    }

    int GetHeight(void) {
        return this.height;
    }

    class Animal GetMom(void) {
        return this.mother;
    }
}

class Cow : Animal {
    int Spots;
    void InitCow(int h, class Animal m, int spot) {
        this.Spots = spot;
        this.InitAnimal(h,m);
    }
}
```

```

    }
    int GetSpots (void) {
        return this.Spots;
    }
}

```

```

void main(void) {
    class Cow betsy;
    class Animal b;
    betsy = New(Cow);

    betsy.InitCow(5, NULL, 1223);
    b = betsy;
    b.GetMom();
    Print("spots: ", betsy.GetSpots(), "    height: ", b.GetHeight());
}

```

OBS: Baseado em projeto desenvolvido por Maggie Johnson no curso de Compiladores da Stanford University.