



Universidade Federal de Campina Grande – UFCG  
Centro de Engenharia Elétrica e Informática - CEEI  
Departamento de Sistemas e Computação – DSC

**Disciplina:** Compiladores

**Professora:** Franklin Ramalho

## Proposta de Projeto para Compilador

**Linguagem Fonte:** 

**Linguagem Destino:** *Assembly*

```
push    ebp
mov     ebp, esp
movzx   ecx, [ebp+arg_0]
pop     ebp
movzx   dx, cl
lea     eax, [edx+edx]
```

### Equipe

Adalberto Teixeira

Andrey Menezes

Augusto Queiroz

Campina Grande, Agosto – 2011

## Exemplo de Funções e procedimentos

```
class ExemploFuncoesProcedimentos {

    static volatile int x = 0;
    static transient int[] y = { 1, 2, 3 };

    static int funcao1() {
        return 10;
    }

    static String funcao2() {
        return "string";
    }

    static int funcao3(int a) {
        return a;
    }

    static String funcao4(String b) {
        return b;
    }

    static double funcao5() {
        return 5.5;
    }

    static boolean funcao6(boolean a) {
        return true;
    }

    static int[] funcao7() {
        return y;
    }

    static void procedure1() {
        int x = 5 + 4;
    }

    static void procedure2() {
        y[1] = 55;
    }

    public static void main(String[] args) {
        ExemploFuncoesProcedimentos.funcao1();
        ExemploFuncoesProcedimentos.funcao2();
        ExemploFuncoesProcedimentos.funcao3(999);
        ExemploFuncoesProcedimentos.funcao4("Ola!");
        ExemploFuncoesProcedimentos.funcao5();
        ExemploFuncoesProcedimentos.funcao6(false);
        ExemploFuncoesProcedimentos.funcao7();
        ExemploFuncoesProcedimentos.procedure1();
        ExemploFuncoesProcedimentos.procedure2();
    }
}
```

## Exemplo de Comandos condicionais

```

class ExemploComandosCondicionais {

    static final int[] notas = { 8, 7, 7, 6, 10, 3 };

    /* If-Else */
    static int mediaAprovados() {
        int media = 0;
        for (int y = 0; y < notas.length; y++) {
            if (notas[y] >= 7) {
                media += notas[y];
            } else {
                media += (notas[y] * (60 / 100));
            }
        }
        return media;
    }

    /* If-Elseif-Else */
    static String aprovado(int aluno) {
        if (notas[aluno] >= 7) {
            return "Aprovado";
        } else if (notas[aluno] >= 5) {
            return "Final";
        } else {
            return "Reprovado";
        }
    }

    /* Switch */
    static String diaSemana(int diaSemana) {
        switch (diaSemana) {
            case 1:
                return "Domingo";
            case 2:
                return "Segunda-feira";
            case 3:
                return "Terca-feira";
            case 4:
                return "Quarta-feira";
            case 5:
                return "Quinta-feira";
            case 6:
                return "Sexta-feira";
            case 7:
                return "Sabado";
            default:
                return "Este dia nao existe!";
        }
    }

    public static void main(String[] args) {
        ExemploComandosCondicionais.mediaAprovados();
        ExemploComandosCondicionais.aprovado(1);
        ExemploComandosCondicionais.diaSemana(2);
    }
}

```

## Exemplo de Comandos iterativos

```

class ExemploComandosIterativos {

    /* FOR */
    static int sum(int initialValue, int finalValue) {
        int totalSum = 0;
        for (int i = initialValue; i <= finalValue; i++) {
            totalSum += i;
        }
        return totalSum;
    }

    /* FOREACH */
    static int sumArray(int[] values) {
        int totalSum = 0;
        for (int value=values.length(); value >= 0; value--) {
            totalSum += value;
        }
        return totalSum;
    }

    /* WHILE */
    static int sumLastTwo(int v1, int v2, int valueMax) {
        int sum = 0;
        int last1 = v1;
        int last2 = v2;
        while (sum <= valueMax) {
            sum = last1 + last2;
            last1 = last2;
            last2 = sum;
        }
        return last1;
    }

    /* DO-WHILE */
    static int exponential(int value, int exp) {
        int i = 1;
        int valueExp = 1;
        do {
            valueExp = value * valueExp;
            i++;
        } while (i <= exp);
        return valueExp;
    }

    public static void main(String[] args) {
        ExemploComandosIterativos.sum(4, 6);
        int[] array = { 6, 4, 5 };
        ExemploComandosIterativos.sumArray(array);
        ExemploComandosIterativos.sumLastTwo(1, 7, 40);
        ExemploComandosIterativos.exponential(2, 3);
    }
}

```

## Exemplo de Comando de atribuição

```

class ExemploComandosAtribuicao {

    /* += ++ */
    static int sumCresc(int initialValue, int finalValue) {
        int sum = 0;
        int i = initialValue;
        while (i <= finalValue) {
            sum += i;
            i++;
        }
        return sum;
    }

    /* -- */
    static int sumDecresc(int initialValue, int finalValue) {
        int sum = 0;
        int i = initialValue;
        while (i >= finalValue) {
            sum += i;
            i--;
        }
        return sum;
    }

    /* -= */
    static int decrement(int initialValue, int dec, int times) {
        int total = initialValue;
        for (int i = 0; i < times; i++)
            total -= dec;
        return total;
    }

    /* *= */
    static int exp(int value, int exp) {
        int total = 1;
        for (int i = 0; i < exp; i++)
            total *= value;
        return total;
    }

    /* /= */
    static int div(int value, int div, int times) {
        int finalValue = value;
        for (int i = 0; i < times; i++) {
            finalValue /= div;
        }
        return finalValue;
    }

    /* %= */
    static int mod(int value, int div) {
        value %= div;
        return value;
    }

    public static void main(String[] args) {
        ExemploComandosAtribuicao.sumCresc(2, 6);
        ExemploComandosAtribuicao.sumDecresc(8, 4);
        ExemploComandosAtribuicao.decrement(10, 2, 3);
        ExemploComandosAtribuicao.exp(2, 5);
        ExemploComandosAtribuicao.div(16, 2, 3);
        ExemploComandosAtribuicao.mod(10, 4);
    }
}

```

```
}
```

## Exemplo de Expressões aritméticas e booleanas

```
class ExemploExpressoesAritmeticasBooleanas {

    /* +, -, /, *, uso de parenteses */
    static void expAritmeticas() {
        int a = 1 + 3;
        int b = a - 4;
        int c = 3 * (-4);
        int d = c / 2;
        int e = 14 % 3;
        int f = e + 5 - 1 + b;
        a = (b + c);
        b = (1 + 3) - c;
        a = (b + 4) * (5 - c);
        f = ((a + b) / 2);
    }

    /* |, &, !, &&, || */
    static void expBooleanas() {
        boolean a = true;
        boolean b = false;
        boolean c = (a | b);
        boolean d = c & true;
        d = a && !b;
        boolean e = d || (b & c);
    }

    /* &, |, ^, ~ */
    static void expBooleanasBitWise() {
        int a = 10;
        int b = 2;
        int c = a & b;
        int d = a | b;
        int e = a ^ b;
        int f = ~e;
    }

    public static void main(String[] args) {
        ExemploExpressoesAritmeticasBooleanas.expAritmeticas();
        ExemploExpressoesAritmeticasBooleanas.expBooleanas();
        ExemploExpressoesAritmeticasBooleanas.expBooleanasBitWise();
    }
}
```

## Exemplo de Expressões relacionais

```
class ExemploExpressoesRelacionais {

    static final int[] notas = { 8, 7, 7, 6, 10, 3 };

    /* >=, <= */
    static int mediaAprovados() {
        int media = 0;
        for (int y = 0; y < notas.length; y++) {
            if (notas[y] >= 7) {

```

```

        media += notas[y];
    } else if (notas[y] <= 5) {
        media += (notas[y] * (60 / 100));
    }
}
return media;
}

/* !=, ==, >, < */
static void atualizarNota(int aluno, int nota) {
    int k = notas[aluno];
    if (k != nota) {
        if (k < nota) {
            notas[aluno] = nota;
        }
        if (k > nota) {
            notas[aluno] = k + 1;
        }
    }
    if (k == nota) {
        notas[aluno] = k + 2;
    }
}

public static void main(String[] args) {
    ExemploExpressoesRelacionais.mediaAprovados();
    ExemploExpressoesRelacionais.atualizarNota(1, 10);
}
}

```

## Exemplo de Literais

```

class ExemploLiterais {

    /* char, string, boolean, int, long, float e double */
    public static void main(String[] args) {

        char c1 = 'a', c2 = 'z';
        char c3 = '1', c4 = '0';
        char c5 = 'n', c6 = 't', c7 = 'b';
        char c8 = 'r', c9 = 'f', c10 = '\\';
        String s1 = "Compiladores", s2 = "X";
        String s3 = " Compiladores Java ";
        boolean b1 = true, b2 = false;
        int i1 = 190, i2 = 1800000;
        long l1 = -9019090910909190L, l2 = 18000009019019019L;
        float f1 = 1.0f, f2 = -6.09090f, f3 = 6.5E32f;
        float f4 = -3333.5E-12f, f5 = 908E18f, f6 = 7.010101010101f;
        double d1 = 1.0, d2 = -6.09090, d3 = 6.5E32;
        double d4 = -3333.5E-12, d5 = 908E188, d6 = 7.010101010101;

        if (true) {
            if (!false) {
                if ("Professores".length() > 10 && 'd' > 'a'
                    && "Compilar".charAt(0) == 'C') {
                    String compiler = "JavaCompiler \\o/ Java Compiler";
                    if (d1 > d2 || c2 > c3) {
                        String concat = s1 + " " + s2 + " " + s3;
                    }
                }
            }
        }
    }
}

```

```

    }
}
}

```

## Exemplo de Arranjos

```

class ExemploArranjos {

    static final String[] nomes = { "Carlitos Tevez", "Ronaldo", "Romario", "Silvio
Santos" };
    static final int[] notas = { 8, 7, 7, 6 };
    static final int[] notas2 = { 2, 10, 9, 10 };
    static transient double[] medias = { 0.0, 0.0, 0.0, 0.0 };

    public static void main(String[] args) {
        for (int y = 0; y < notas.length; y++) {
            medias[y] = (notas[y] + notas2[y]) / 2;
        }

        for (int i = 0; i < nomes.length; i++) {
            String a = "Estudante: " + nomes[i];
            String b = "Media: " + medias[i];
        }
    }
}

```

## Exemplo de Chamada de funções e procedimentos (Ver outros exemplos)

## Exemplo de Seqüenciadores de escape

```

class ExemploSequenciadoresEscape {

    /* BREAK */
    static void calculateTable(int value) {
        int mult = 1;
        while (true) {
            if (mult > 10)
                break;
            else {
                String test = value + " * " + mult + " = "
                    + (value * mult);
                mult++;
            }
        }
    }

    /* GOTO */
    static void increaseGrade(int value) {

        validation: {
            if (value < 0) {
                break validation;
            }
            validation2: {
                int mult = 1;
                while (true) {
                    if (mult > 10) {

```



```

        break validation2;
    } else {
        String pk = value + " * " + mult + " = "
            + (value * mult);
        mult++;
    }
}

}

}

/* RETURN */
static int calculateSquare(int x) {
    return x * x;
}

/* CONTINUE */
static void calculateSquares(int initialValue, int finalValue) {
    for (int i = initialValue; i <= finalValue; i++) {
        if (i == 0)
            continue;
        else {
            String method = "Square (" + i + ") : " + (i * i);
        }
    }
}

public static void main(String[] args) {

    ExemploSequenciadoresEscape.calculateTable(4);
    ExemploSequenciadoresEscape.increaseGrade(-1);
    ExemploSequenciadoresEscape.calculateSquare(7);
    ExemploSequenciadoresEscape.calculateSquares(-4, 7);
}
}

```

## Exemplo Completo

```

/** ESSE EXEMPLO CONTEM: */

/* FUNCOES */
/* PROCEDURES */
/* COMANDOS: */
/* - CONDICIONAIS (if/else; if/else-if/else e switch) */
/* - ITERATIVOS (for, foreach, while e do-while) */
/* - ATRIBUICAO (=, +=, /=, ++) */
/* EXPRESSOES: */
/* - ARITMETICAS (+, -, * e /) */
/* - BOOLEANAS (||, !) */
/* - RELACIONAIS (<, ==, >=, <=, >, !=) */
/* LITERAIS */
/* ARRANJOS (String e int) */
/* CHAMADA DE FUNCOES */
/* CHAMADA DE PROCEDURES */
/* SEQUENCIADORES DE ESCAPE (break e return) */

/* AINDA FALTARAM ALGUNS NESSE EXEMPLO: */
/* COMANDOS: */
/* - ATRIBUICAO (--, %=, --) */
/* EXPRESSOES: */
/* - ARITMETICAS (% , >>, <<, ~) */
/* - BOOLEANAS (&, &&, |) */
/* SEQUENCIADORES DE ESCAPE (continue e goto) */

```

```

class ExemploCompleto {

    static final int MAX_STUDENTS = 25;
    static final int CLASS_A = 1;
    static final int CLASS_B = 2;

    static final String[] studentNamesClassA = new String[MAX_STUDENTS];
    static transient double[] studentGradesClassA = new double[MAX_STUDENTS];
    static volatile int classASize = 0;

    static final String[] studentNamesClassB = new String[MAX_STUDENTS];
    static transient double[] studentGradesClassB = new double[MAX_STUDENTS];
    static volatile int classBSize = 0;

    static int getStudentIndexClassA(String name) {
        int index = 0;
        while (index < classASize) {
            if (name.equals(name)) {
                break;
            }
            index++;
        }
        return index;
    }

    static int getStudentIndexClassB(String name) {
        int index = 0;
        while (index < classBSize) {
            if (name.equals(name)) {
                break;
            }
            index++;
        }
        return index;
    }

    static void addStudentName(String newName, int studentClass) {
        switch (studentClass) {
            case (CLASS_A):
                if (classASize < MAX_STUDENTS) {
                    studentNamesClassA[classASize++] = newName;
                }
                break;
            case (CLASS_B):
                if (classBSize < MAX_STUDENTS) {
                    studentNamesClassB[classBSize++] = newName;
                }
                break;
            default:
        }
    }

    static boolean setStudentGrade(String name, int studentClass, double grade) {
        boolean result = false;
        switch (studentClass) {
            case (CLASS_A):
                int indexA = getStudentIndexClassA(name);
                if (indexA < classASize || indexA < MAX_STUDENTS) {
                    studentGradesClassA[indexA] = grade;
                    result = !result;
                }
        }
    }
}

```

```

        break;
    case (CLASS_B):
        int indexB = getStudentIndexClassA(name);
        if (classBSize < MAX_STUDENTS) {
            studentGradesClassB[indexB] = grade;
            result = !result;
        }
        break;
    default:
        }
    return result;
}

static boolean hasFailed(int index, int studentClass) {
    if (index >= 0) {
        if (studentClass == CLASS_A) {
            if (index <= classASize - 1)
                return (studentGradesClassA[index] < 5);
        } else if (studentClass == CLASS_B) {
            if (index <= classBSize - 1)
                return (studentGradesClassB[index] < 5);
        } else {
            return false;
        }
    }
    return false;
}

static double calculateMeanGradeByClass(int myClass) {
    double meanGrade, tmp = 0.0;
    switch (myClass) {
        case (CLASS_A):
            for (int i = 0; i < studentGradesClassA.length; i++) {
                tmp += studentGradesClassA[i];
            }
            meanGrade = tmp / studentGradesClassA.length;
            break;
        case (CLASS_B):
            for (double grade = 0; grade <= studentGradesClassB; grade++) {
                tmp = tmp + grade;
            }
            meanGrade = tmp / studentGradesClassB.length;
            break;
        default:
            meanGrade = 0.0;
    }
    return meanGrade;
}

static double getDiffMaxMinGrades() {
    return getHighestGrade() - getLowestGrade();
}

static double getHighestGrade() {
    double high = 0.0;
    for (double grade = 0; grade <= studentGradesClassA; grade++) {
        high = (grade > high) ? grade : high;
    }
    for (double grade = 0; grade <= studentGradesClassB; grade++) {
        high = (grade > high) ? grade : high;
    }
    return high;
}

```

```

}

static double getLowestGrade() {
    double low = 10;
    for (double grade = 0; grade <= studentGradesClassA; grade++) {
        low = (grade < low) ? grade : low;
    }
    for (double grade = 0; grade <= studentGradesClassB; grade++) {
        low = (grade > low) ? grade : low;
    }
    return low;
}

static double getStandardDeviation(int myClass) {

    double result = 0.0;
    double mean = calculateMeanGradeByClass(myClass);
    switch (myClass) {
        case (CLASS_A):
            for (double grade = 0; grade <= studentGradesClassA; grade++) {
                result += (grade - mean) * (grade - mean);
            }
            result = result / studentGradesClassA.length;
            break;
        case (CLASS_B):
            for (double grade = 0; grade <= studentGradesClassB; grade++) {
                result += (grade - mean) * (grade - mean);
            }
            result /= studentGradesClassA.length();
            break;
        default:
    }
    return result;
}

static void printNamesClassA() {
    int i = 0;
    while (true) {
        if (i >= classASize) {
            break;
        }
        String teste = "\t" + studentNamesClassA[i];
        i++;
    }
}

static void printNamesClassB() {
    int i = 0;
    do {
        if (studentNamesClassB[i] != null) {
            String teste = "\t" + studentNamesClassB[i];
        }
        i++;
    } while (i < classBSize);
}

public static void main(String[] args) {
    ExemploCompleto.addStudentName("Jose Almeida", ExemploCompleto.CLASS_A);
    ExemploCompleto.addStudentName("Joaquim Barros", ExemploCompleto.CLASS_A);
    ExemploCompleto.addStudentName("Maria da Silva", ExemploCompleto.CLASS_A);
    ExemploCompleto.addStudentName("Maria das Dores", ExemploCompleto.CLASS_B);
    ExemploCompleto.addStudentName("Carlos Siqueira", ExemploCompleto.CLASS_B);
    ExemploCompleto.addStudentName("Zeze di Camargo", ExemploCompleto.CLASS_B);
}

```

```
        ExemploCompleto.printNamesClassA();  
        ExemploCompleto.printNamesClassB();  
    }  
}
```