Leitor-Exibidor de Bytecode

Gabriel Nazareno Halabi - 15/0010290
Mariana Borges de Sampaio - 18/0046926
Paulo Mauricio Costa Lopes - 18/0112520
Pedro Henriques Nogueira - 14/0065032

Introdução

A JVM precisa verificar para si mesma que as restrições desejadas são satisfeitas pelo arquivo .class que está sendo incorporado ao seu leitor. Logo, a implementação da JVM verifica se cada classe do arquivo satisfaz as restrições necessárias. Para que se tenha a verificação do arquivo .class é necessário o leitor-exibidor de bytecodes.

Ler

O leitor tem como função ler o arquivo .class que foi enviado e ele precisa verificar inicialmente se esse arquivo é adequado. Para essa verificação ele analisa a estrutura do arquivo .class

ESTRUTURA INTERNA DE ARQUIVO .CLASS
ESTRUTURA CLASSFILE

```
ClassFile {
u4                  magic;
u2                  minor_version;
u2                  major_version;
u2                  constant_pool_count;
cp_info             constant_pool [constant_pool_count-1];
u2                  access_flags;
u2                  this_class;
u2                  super_class;
u2                  interfaces_count;
u2                  interfaces [interfaces_count];
u2                  fields_count;
field_info  fields [fields_count];
u2                  methods_count;
method_info         methods [methods_count];
u2                  attributes_count;
attribute_info      attributes [attributes_count];
}
```

Exibir

O exibidor realiza a decodificação das instruções que foram enviadas pelo arquivo .class , dessa forma ele imprime o arquivo .class na tela do usuário.

- Imprime a constante pool;
- Imprime Fields;
- Imprime Métodos;
- Imprime os Atributos;
- Decodifica o código.

Exemplos

- Hello world
- Operações com inteiros

Execução completa do programa (sem warnings)

```
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$ make limpa && make debug
rm -rdf dep
rm -rdf bin
rm -f main.exe
g++ -Iinclude src/classFile.cpp -M -MT dep/classFile.d -MT bin/classFile.o -MP -MF dep/classFile.d -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG
g++ -Iinclude src/classFile.cpp -c -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG -o bin/classFile.o
g++ -Iinclude src/readClassByteCode.cpp -M -MT dep/readClassByteCode.d -MT bin/readClassByteCode.o -MP -MF dep/readClassByteCode.d -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG
g++ -Iinclude src/readClassByteCode.cpp -c -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG -o bin/readClassByteCode.o
g++ -Iinclude src/_main.cpp -M -MT dep/_main.d -MT bin/_main.o -MP -MF dep/_main.d -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG
g++ -Iinclude src/_main.cpp -c -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG -o bin/_main.o
g++ -o main.exe bin/classFile.o bin/readClassByteCode.o bin/_main.o   -std=c++17 -Wall -pedantic -Wextra -Werror=init-self -g -ggdb -O0 -DDEBUG
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$
```

Cppcheck funcionando sem problemas

```
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$ make cppcheck
cppcheck --enable=all . -I./include --suppress=missingIncludeSystem --suppress=unusedFunction
Checking src/_main.cpp ...
1/3 files checked 4% done
Checking src/classFile.cpp ...
2/3 files checked 39% done
Checking src/readClassByteCode.cpp ...
3/3 files checked 100% done
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$
```

Exemplo do HelloWorld

```java
public class helloworld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}

```

```
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$ make executa
./main.exe
Using default.class....
--------------------------------
| Magic #         |  3405691582
| Minor v.        |  0
| Major v.        |  52
| Interfaces #    |  0
| Fields #        |  0
| Methods #       |  2
--------------------------------
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$
```

Operações básicas com inteiros

```java
public class teste02 {

    private int a;
    private int b;

    public teste02(int a, int b) {
        this.a = a;
        this.b = b;
    }

    public static void main(String[] args) {
        teste02 objeto1 = new teste02(200, 550);
        teste02 objeto2 = new teste02(400, 4);
        System.out.print("A soma eh:");
        System.out.println(objeto1.soma(objeto1.a, objeto2.b));
        System.out.print("A multiplicação eh:");
        System.out.println(objeto1.multiplica(objeto1.a, objeto1.b));
        System.out.print("A divisão eh:");
        System.out.println(teste02.divisao(objeto1.a, objeto2.b));
    }

    public int soma(int a, int b) { return a + b; }
    public int multiplica(int a, int b) { return a * b; }
    public static int divisao(int a, int b) { return a / b; }

    public int getA() { return this.a; }
    public int getB() { return this.b; }
    public void setA(int a) { this.a = a; }
    public void setB(int b) { this.b = b; }
}
```

Resultado do código de operações

```
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$ ./main.exe defaultsomamul
tiplicadivide.class
--------------------------------
| Magic #          |  3405691582
| Minor v.         |  0
| Major v.         |  52
| Interfaces #     |  0
| Fields #         |  2
| Methods #        |  9
--------------------------------
bj@BJ114:/mnt/c/Users/Pedro Nogueira/Documents/projects/unb-2021_1/JVM-Software-Basico/fonte$
```