Universidad Mayor de San Simon
Comunidad Haskell San Simon
1er OpenHouse Haskell Community

Serializador/Deserializador de Java Bytecode Classfile con la Biblioteca Binary

Est.: Carlos Gomez
carliros@comunidadhaskell.org

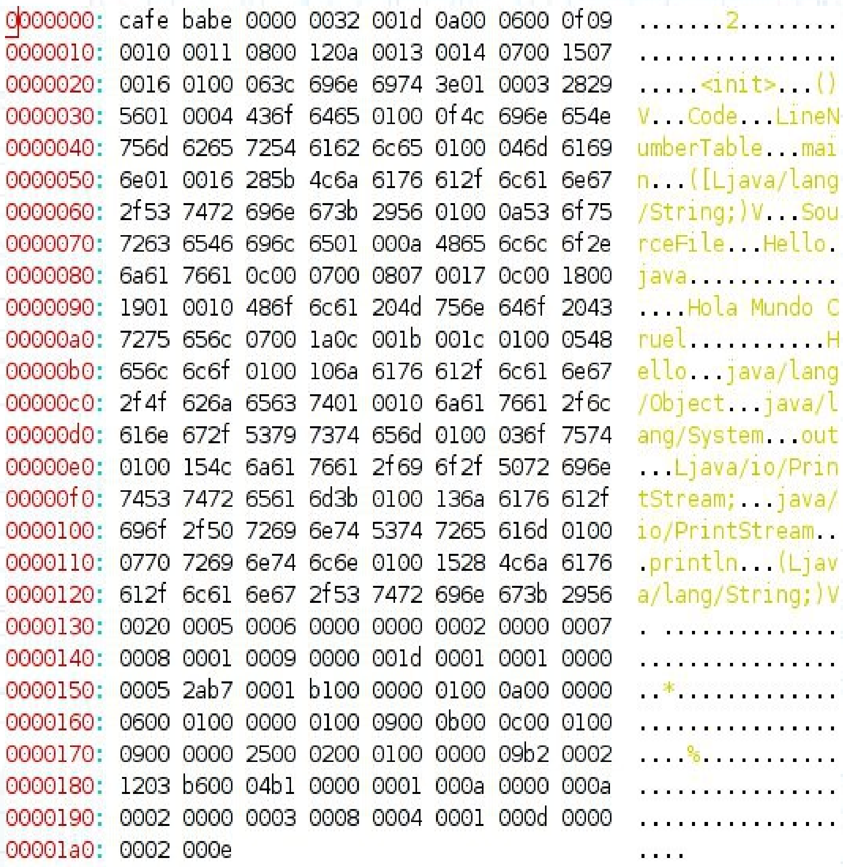
Cbba, Jueves 15 de Abrill del 2010

Contenido

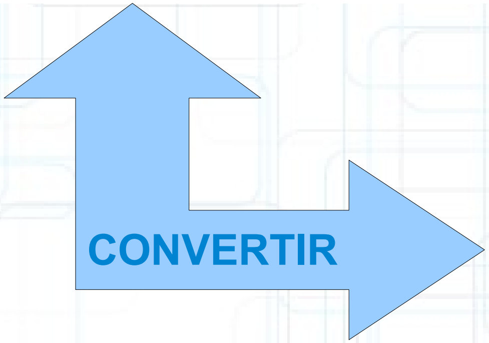
- Introduccion
  - Concepto Ser/Deser
  - Razon de Desarrollo
- Desarrollo
  - Estructura ClassFile
  - Representacion en Haskell
  - Serializador 1ra version
  - Ser/Deser 2da version
- Pruebas y otros problemas
- Preguntas

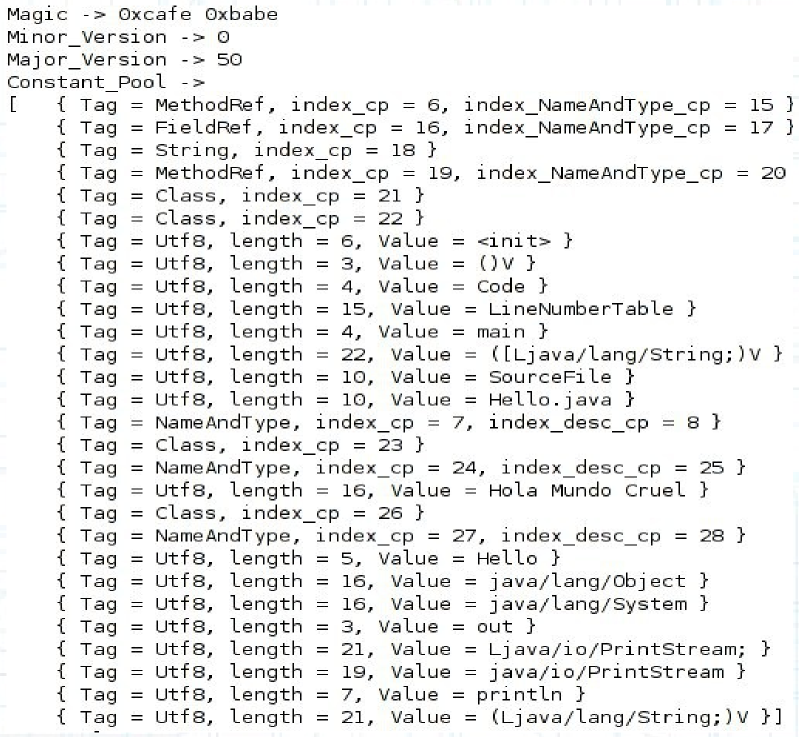
Introducción

Serializador/Deserializador



```
0000000: cafe babe 0000 0032 001d 0a00 0600 0f09  .......2........
0000010: 0010 0011 0800 120a 0013 0014 0700 1507  ................
0000020: 0016 0100 063c 696e 6974 3e01 0003 2829  .....<init>...()
0000030: 5601 0004 436f 6465 0100 0f4c 696e 654e  V...Code...LineN
0000040: 756d 6265 7254 6162 6c65 0100 046d 6169  umberTable...mai
0000050: 6e01 0016 285b 4c6a 6176 612f 6c61 6e67  n...([Ljava/lang
0000060: 2f53 7472 696e 673b 2956 0100 0a53 6f75  /String;)V...Sou
0000070: 7263 6546 696c 6501 000a 4865 6c6c 6f2e  rceFile...Hello.
0000080: 6a61 7661 0c00 0700 0807 0017 0c00 1800  java............
0000090: 1901 0010 486f 6c61 204d 756e 646f 2043  ....Hola Mundo C
00000a0: 7275 656c 0700 1a0c 001b 001c 0100 0548  ruel...........H
00000b0: 656c 6c6f 0100 106a 6176 612f 6c61 6e67  ello...java/lang
00000c0: 2f4f 626a 6563 7401 0010 6a61 7661 2f6c  /Object...java/l
00000d0: 616e 672f 5379 7374 656d 0100 036f 7574  ang/System...out
00000e0: 0100 154c 6a61 7661 2f69 6f2f 5072 696e  ...Ljava/io/Prin
00000f0: 7453 7472 6561 6d3b 0100 136a 6176 612f  tStream;...java/
0000100: 696f 2f50 7269 6e74 5374 7265 616d 0100  io/PrintStream..
0000110: 0770 7269 6e74 6c6e 0100 1528 4c6a 6176  .println...(Ljav
0000120: 612f 6c61 6e67 2f53 7472 696e 673b 2956  a/lang/String;)V
0000130: 0020 0005 0006 0000 0000 0002 0000 0007  . ..............
0000140: 0008 0001 0009 0000 001d 0001 0001 0000  ................
0000150: 0005 2ab7 0001 b100 0000 0100 0a00 0000  ..*.............
0000160: 0600 0100 0000 0100 0900 0b00 0c00 0100  ................
0000170: 0900 0000 2500 0200 0100 0000 09b2 0002  ....%...........
0000180: 1203 b600 04b1 0000 0001 000a 0000 000a  ................
0000190: 0002 0000 0003 0008 0004 0001 000d 0000  ................
00001a0: 0002 000e                                ....
```





```
Magic -> 0xcafe 0xbabe
Minor_Version -> 0
Major_Version -> 50
Constant_Pool ->
[   { Tag = MethodRef, index_cp = 6, index_NameAndType_cp = 15 }
    { Tag = FieldRef, index_cp = 16, index_NameAndType_cp = 17 }
    { Tag = String, index_cp = 18 }
    { Tag = MethodRef, index_cp = 19, index_NameAndType_cp = 20
    { Tag = Class, index_cp = 21 }
    { Tag = Class, index_cp = 22 }
    { Tag = Utf8, length = 6, Value = <init> }
    { Tag = Utf8, length = 3, Value = ()V }
    { Tag = Utf8, length = 4, Value = Code }
    { Tag = Utf8, length = 15, Value = LineNumberTable }
    { Tag = Utf8, length = 4, Value = main }
    { Tag = Utf8, length = 22, Value = ([Ljava/lang/String;)V }
    { Tag = Utf8, length = 10, Value = SourceFile }
    { Tag = Utf8, length = 10, Value = Hello.java }
    { Tag = NameAndType, index_cp = 7, index_desc_cp = 8 }
    { Tag = Class, index_cp = 23 }
    { Tag = NameAndType, index_cp = 24, index_desc_cp = 25 }
    { Tag = Utf8, length = 16, Value = Hola Mundo Cruel }
    { Tag = Class, index_cp = 26 }
    { Tag = NameAndType, index_cp = 27, index_desc_cp = 28 }
    { Tag = Utf8, length = 5, Value = Hello }
    { Tag = Utf8, length = 16, Value = java/lang/Object }
    { Tag = Utf8, length = 16, Value = java/lang/System }
    { Tag = Utf8, length = 3, Value = out }
    { Tag = Utf8, length = 21, Value = Ljava/io/PrintStream; }
    { Tag = Utf8, length = 19, Value = java/io/PrintStream }
    { Tag = Utf8, length = 7, Value = println }
    { Tag = Utf8, length = 21, Value = (Ljava/lang/String;)V }]
```

Porque desarrollar?

Desarrollo

Estructura 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];
  }
```

Estructura ClassFile --cont

- Constant_Pool
- Lista de Interfaces implementadas
- Lista de Fields
- Lista de Metodos

Estructura ClassFile --cont

- Constant_Pool

Constant Type	Value
==========================	=======
CONSTANT_Class	7
CONSTANT_Fieldref	9
CONSTANT_Methodref	10
CONSTANT_InterfaceMethodref	11
CONSTANT_String	8
CONSTANT_Integer	3
CONSTANT_Float	4
CONSTANT_Long	5
CONSTANT_Double	6
CONSTANT_NameAndType	12
CONSTANT_Utf8	1

```
CONSTANT_Class_info {
        u1 tag;
        u2 name_index;
}
```

```
CONSTANT_Methodref_info {
        u1 tag;
        u2 class_index;
        u2 name_and_type_index;
  }
```

```
CONSTANT_String_info {
        u1 tag;
        u2 string_index;
   }
```

```
CONSTANT_Utf8_info {
        u1 tag;
        u2 length;
        u1 bytes[length];
   }
```

```
CONSTANT_NameAndType_info {
        u1 tag;
        u2 name_index;
        u2 descriptor_index;
   }
```

```
CONSTANT_Integer_info {
        u1 tag;
        u4 bytes;
   }
```

Estructura ClassFile --cont

- Fields

Nombre de Flag	Valor
===============	======
ACC_PUBLIC	0x0001
ACC_PRIVATE	0x0002
ACC_PROTECTED	0x0004
ACC_STATIC	0x0008
ACC_FINAL	0x0010
ACC_SUPER	0x0020
ACC_VOLATILE	0x0040
ACC_TRANSIENT	0x0080
ACC_INTERFACE	0x0200
ACC_ABSTRACT	0x0400

```
field_info {
    u2 access_flags;
    u2 name_index;
    u2 descriptor_index;
    u2 attributes_count;
    attribute_info
        attributes[attributes_count];
  }
```

Estructura ClassFile --cont

- Methods

```
method_info {
    u2 access_flags;
    u2 name_index;
    u2 descriptor_index;
    u2 attributes_count;
    attribute_info
          attributes[attributes_count];
}
```

Lista de attibutos

Estructura ClassFile --cont

- Lista de atributos

```
attribute_info {
        u2 attribute_name_index;
        u4 attribute_length;
        u1 info[attribute_length];
    }
```

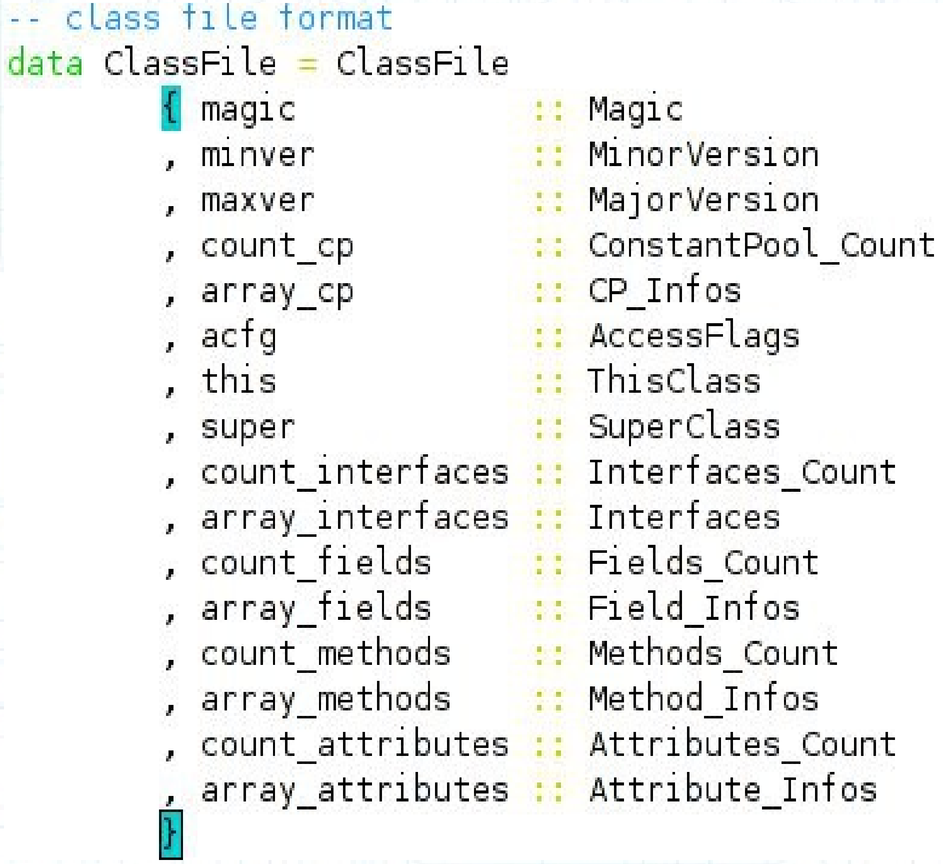
```
ConstantValue_attribute {
        u2 attribute_name_index;
        u4 attribute_length;
        u2 constantvalue_index;
    }
```

```
SourceFile_attribute {
    u2 attribute_name_index;
    u4 attribute_length;
    u2 sourcefile_index;
}
```

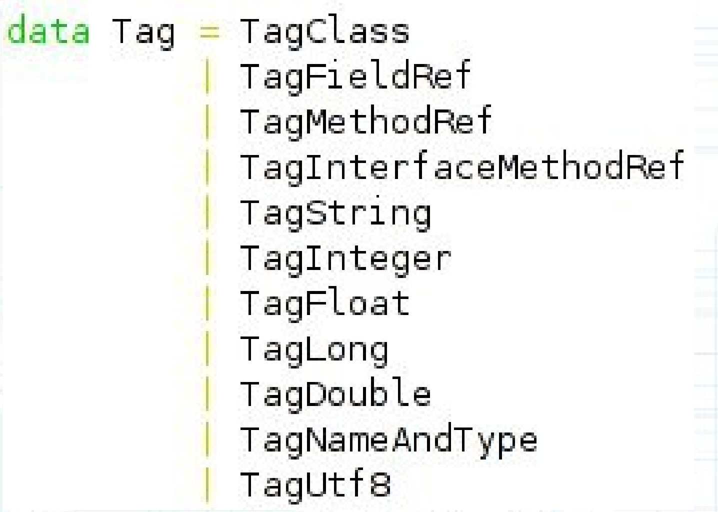
```
Code_attribute {
    u2 attribute_name_index;
    u4 attribute_length;
    u2 max_stack;
    u2 max_locals;
    u4 code_length;
    u1 code[code_length];
    u2 exception_table_length;
    {       u2 start_pc;
            u2 end_pc;
            u2 handler_pc;
            u2 catch_type;
    }   exception_table[exception_table_length];
    u2 attributes_count;
    attribute_info attributes[attributes_count];
}
```

```
Deprecated_attribute {
    u2 attribute_name_index;
    u4 attribute_length;
}
```

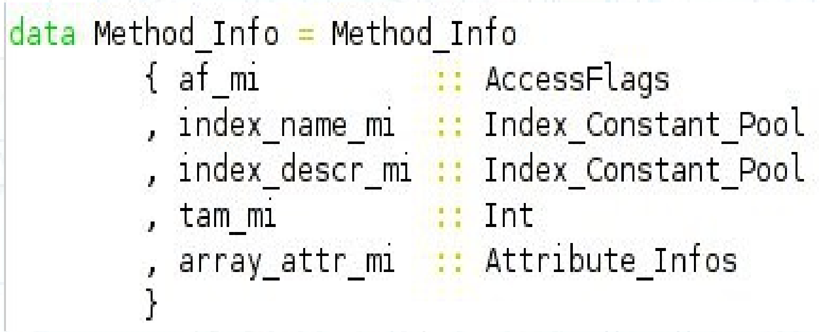
Representacion Haskell ClassFile



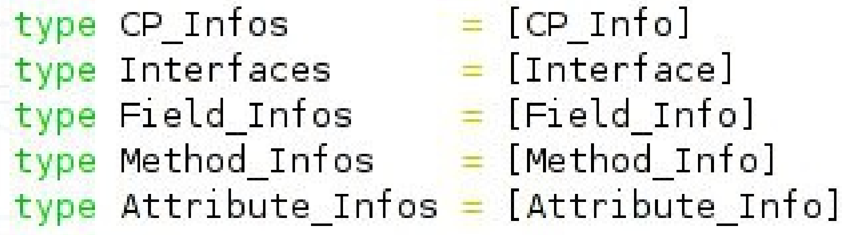
```haskell
-- class file format
data ClassFile = ClassFile
        { magic              :: Magic
        , minver             :: MinorVersion
        , maxver             :: MajorVersion
        , count_cp           :: ConstantPool_Count
        , array_cp           :: CP_Infos
        , acfg               :: AccessFlags
        , this               :: ThisClass
        , super              :: SuperClass
        , count_interfaces   :: Interfaces_Count
        , array_interfaces   :: Interfaces
        , count_fields       :: Fields_Count
        , array_fields       :: Field_Infos
        , count_methods      :: Methods_Count
        , array_methods      :: Method_Infos
        , count_attributes   :: Attributes_Count
        , array_attributes   :: Attribute_Infos
        }
```



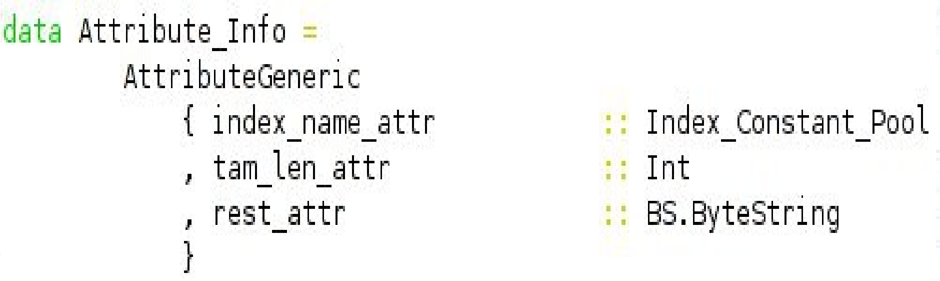
```haskell
data Tag = TagClass
         | TagFieldRef
         | TagMethodRef
         | TagInterfaceMethodRef
         | TagString
         | TagInteger
         | TagFloat
         | TagLong
         | TagDouble
         | TagNameAndType
         | TagUtf8
```



```haskell
data Method_Info = Method_Info
      { af_mi            :: AccessFlags
      , index_name_mi    :: Index_Constant_Pool
      , index_descr_mi   :: Index_Constant_Pool
      , tam_mi           :: Int
      , array_attr_mi    :: Attribute_Infos
      }
```



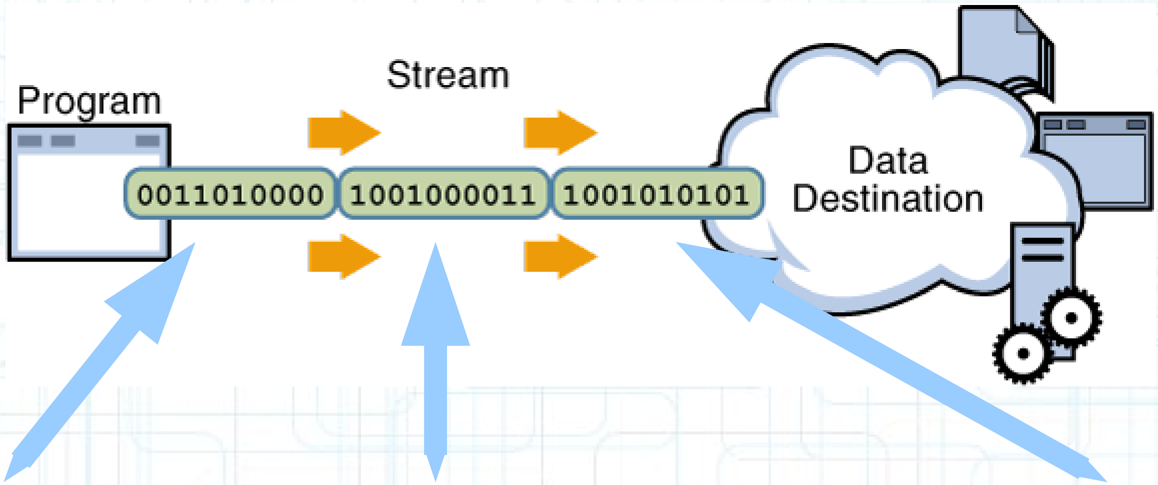
```haskell
type CP_Infos        = [CP_Info]
type Interfaces      = [Interface]
type Field_Infos     = [Field_Info]
type Method_Infos    = [Method_Info]
type Attribute_Infos = [Attribute_Info]
```



```haskell
data Attribute_Info =
     AttributeGeneric
        { index_name_attr            :: Index_Constant_Pool
        , tam_len_attr               :: Int
        , rest_attr                  :: BS.ByteString
        }
```

Serializador - 1ra version

- ByteString

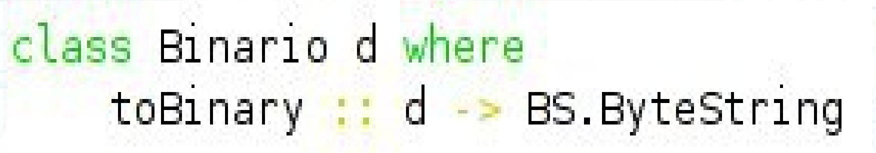


Constant_Pool **++** List_Fields **++** List_Methods

ByteString **append** **ByteString** **append** **ByteString**

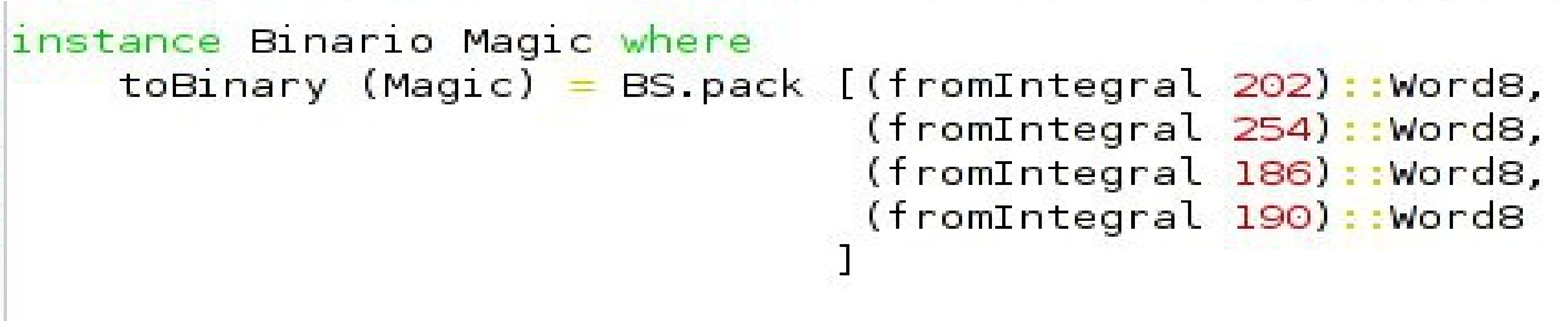
singleton pack

Serializador - 1ra version

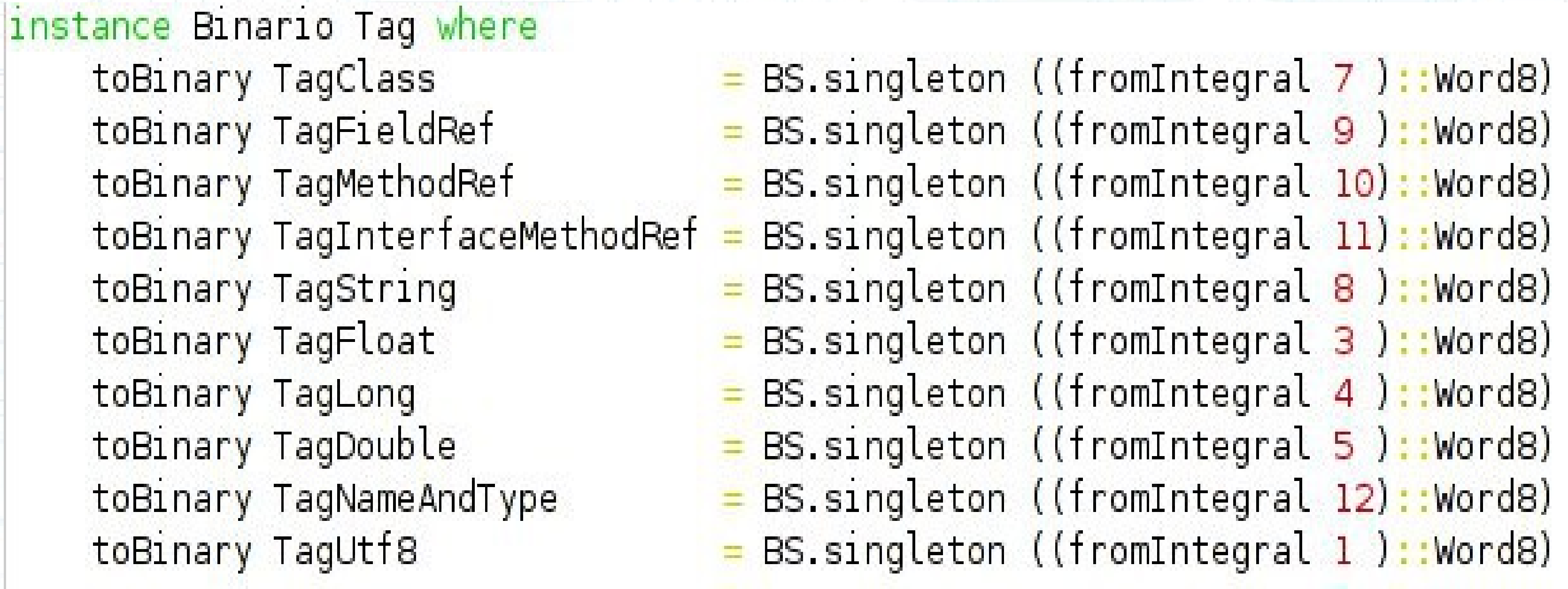


```haskell
class Binario d where
    toBinary :: d -> BS.ByteString
```

- Codigo?



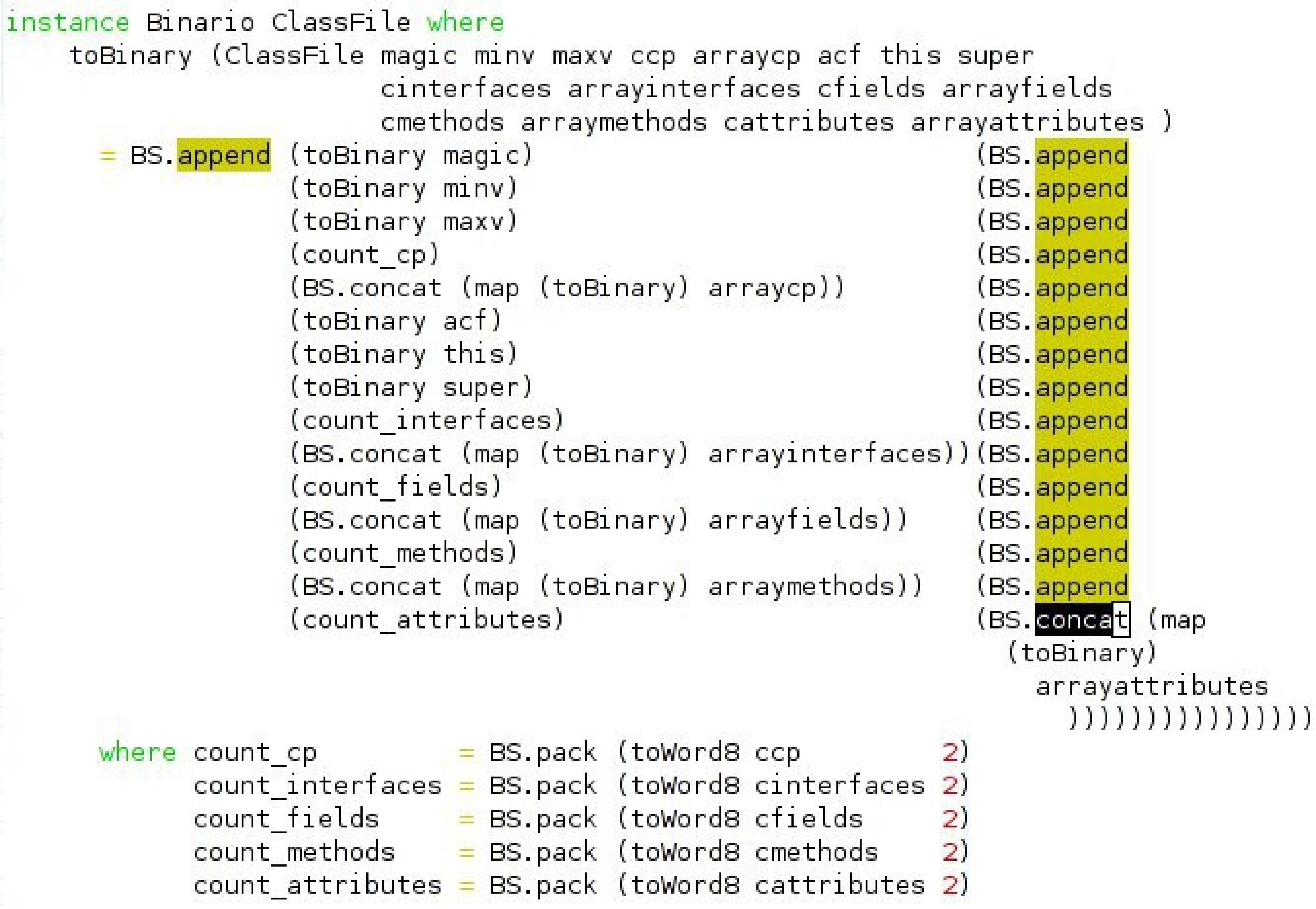
```haskell
instance Binario Magic where
    toBinary (Magic) = BS.pack [(fromIntegral 202)::Word8,
                                (fromIntegral 254)::Word8,
                                (fromIntegral 186)::Word8,
                                (fromIntegral 190)::Word8
                               ]
```



```haskell
instance Binario Tag where
    toBinary TagClass                = BS.singleton ((fromIntegral 7 )::Word8)
    toBinary TagFieldRef             = BS.singleton ((fromIntegral 9 )::Word8)
    toBinary TagMethodRef            = BS.singleton ((fromIntegral 10)::Word8)
    toBinary TagInterfaceMethodRef   = BS.singleton ((fromIntegral 11)::Word8)
    toBinary TagString               = BS.singleton ((fromIntegral 8 )::Word8)
    toBinary TagFloat                = BS.singleton ((fromIntegral 3 )::Word8)
    toBinary TagLong                 = BS.singleton ((fromIntegral 4 )::Word8)
    toBinary TagDouble               = BS.singleton ((fromIntegral 5 )::Word8)
    toBinary TagNameAndType          = BS.singleton ((fromIntegral 12)::Word8)
    toBinary TagUtf8                 = BS.singleton ((fromIntegral 1 )::Word8)
```

Serializador - 1ra version

- Mas codigo?

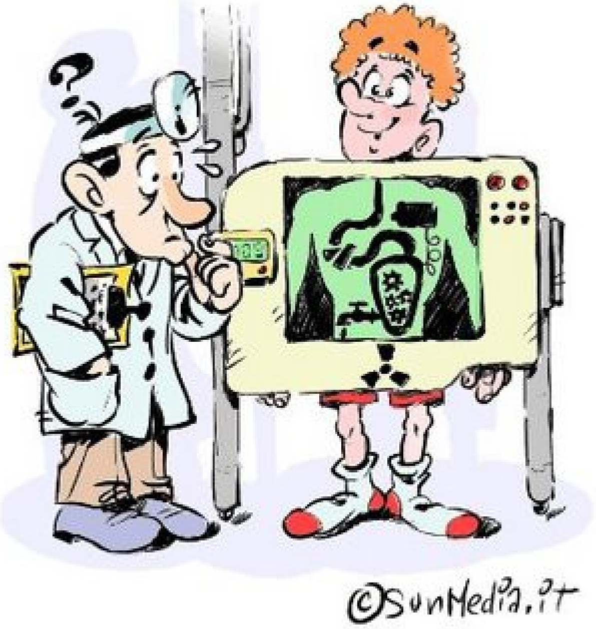


```haskell
instance Binario ClassFile where
    toBinary (ClassFile magic minv maxv ccp arraycp acf this super
                        cinterfaces arrayinterfaces cfields arrayfields
                        cmethods arraymethods cattributes arrayattributes )
      = BS.append (toBinary magic)                            (BS.append
                  (toBinary minv)                             (BS.append
                  (toBinary maxv)                             (BS.append
                  (count_cp)                                  (BS.append
                  (BS.concat (map (toBinary) arraycp))        (BS.append
                  (toBinary acf)                              (BS.append
                  (toBinary this)                             (BS.append
                  (toBinary super)                            (BS.append
                  (count_interfaces)                          (BS.append
                  (BS.concat (map (toBinary) arrayinterfaces))(BS.append
                  (count_fields)                              (BS.append
                  (BS.concat (map (toBinary) arrayfields))    (BS.append
                  (count_methods)                             (BS.append
                  (BS.concat (map (toBinary) arraymethods))   (BS.append
                  (count_attributes)                          (BS.concat (map
                                                                (toBinary)
                                                                  arrayattributes
                                                                    ))))))))))))))))

      where count_cp         = BS.pack (toWord8 ccp         2)
            count_interfaces = BS.pack (toWord8 cinterfaces 2)
            count_fields     = BS.pack (toWord8 cfields     2)
            count_methods    = BS.pack (toWord8 cmethods    2)
            count_attributes = BS.pack (toWord8 cattributes 2)
```

Problemas - 1ra version

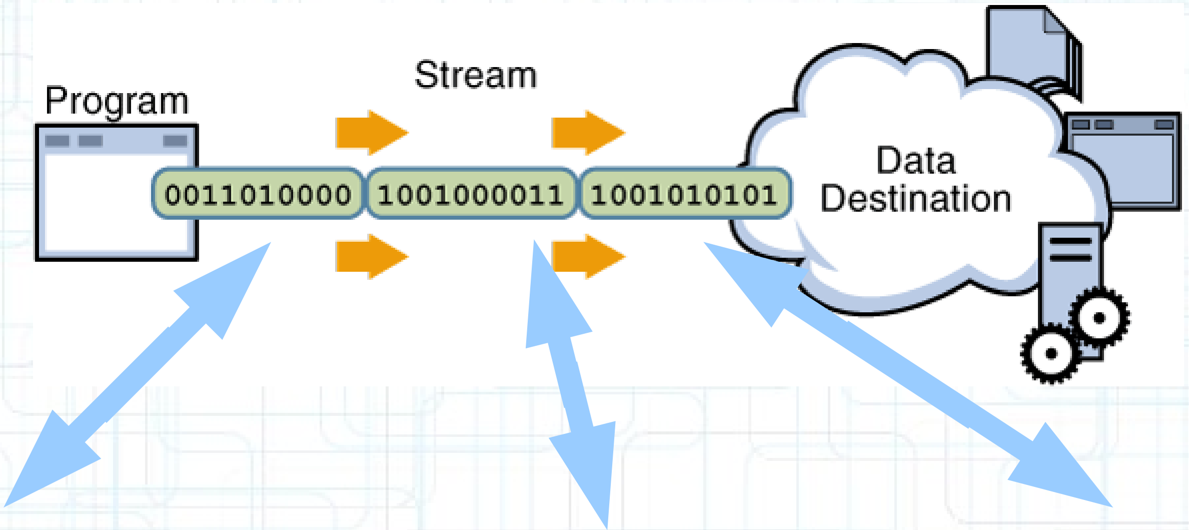
- Muchos append
- Se pierde la legibilidad del codigo
- Todo un choriso de codigo
- Es propenso a errores

Otra Solucion?



Serializador/Deserializador – 2da version

- Biblioteca Binary

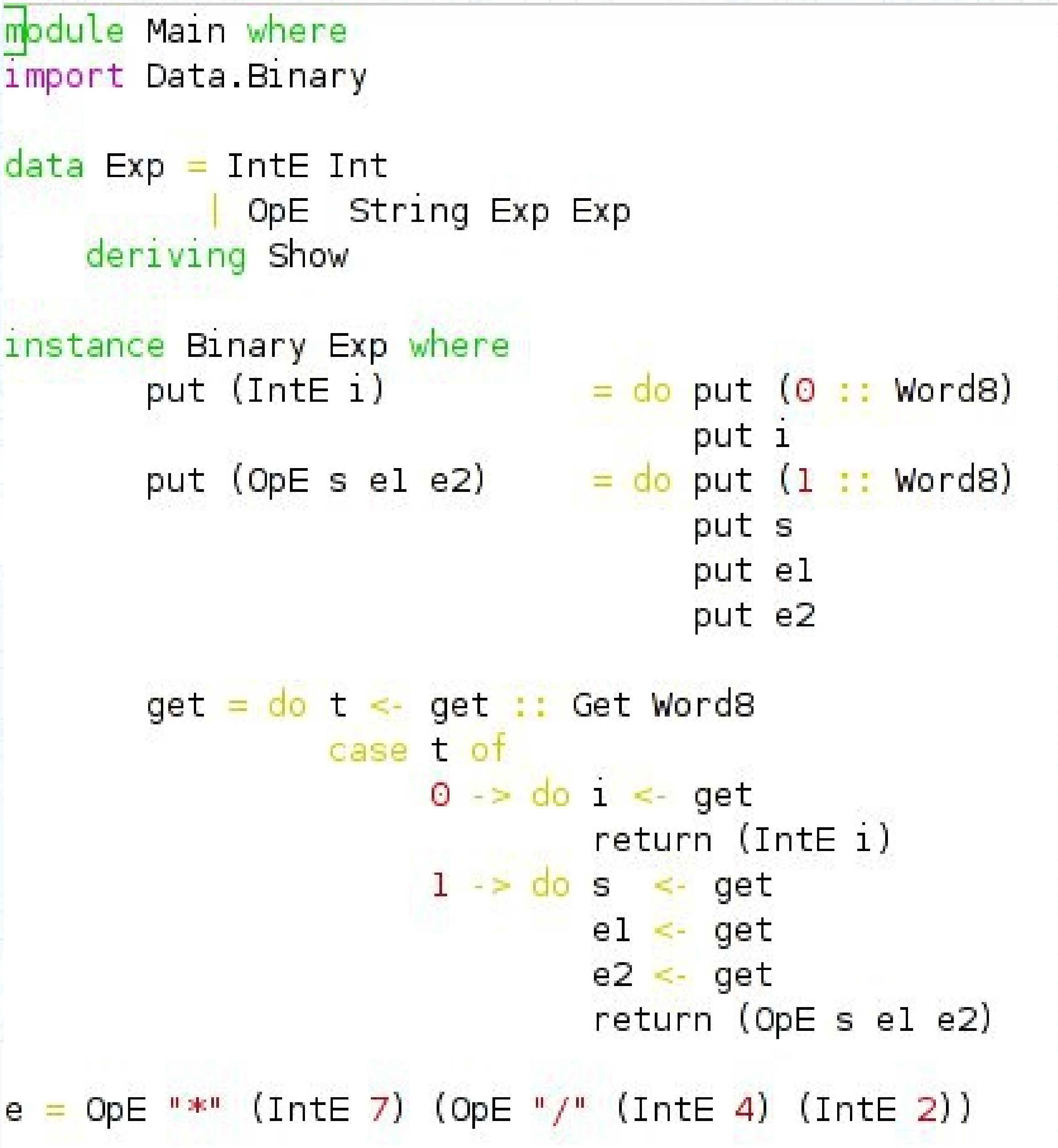


Constant_Pool ++ List_Fields ++ List_Methods

Put/Get Monad Put/Get Monad Put/Get

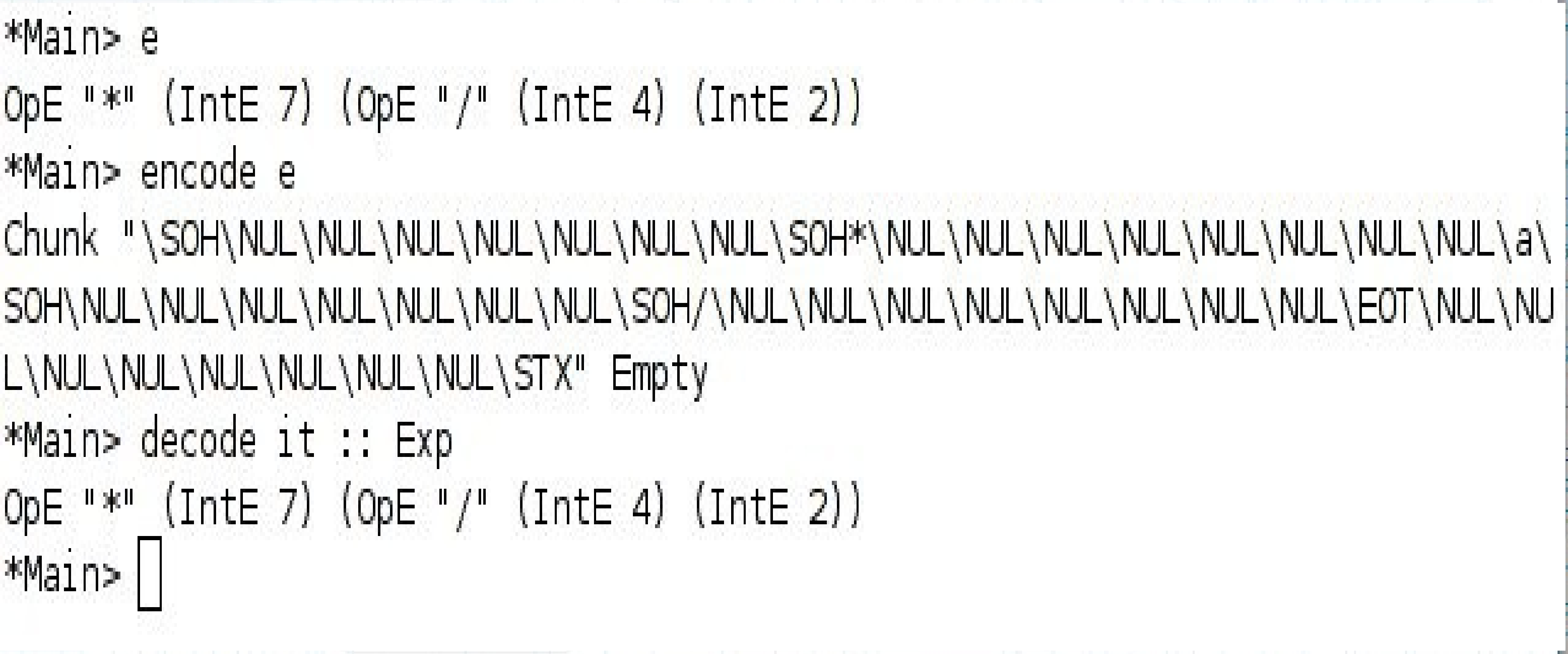
ByteString

Ejemplo uso Binary



```
module Main where
import Data.Binary

data Exp = IntE Int
          | OpE  String Exp Exp
    deriving Show

instance Binary Exp where
       put (IntE i)          = do put (0 :: Word8)
                                  put i
       put (OpE s e1 e2)     = do put (1 :: Word8)
                                  put s
                                  put e1
                                  put e2

       get = do t <- get :: Get Word8
                case t of
                     0 -> do i <- get
                             return (IntE i)
                     1 -> do s  <- get
                             e1 <- get
                             e2 <- get
                             return (OpE s e1 e2)

e = OpE "*" (IntE 7) (OpE "/" (IntE 4) (IntE 2))
```

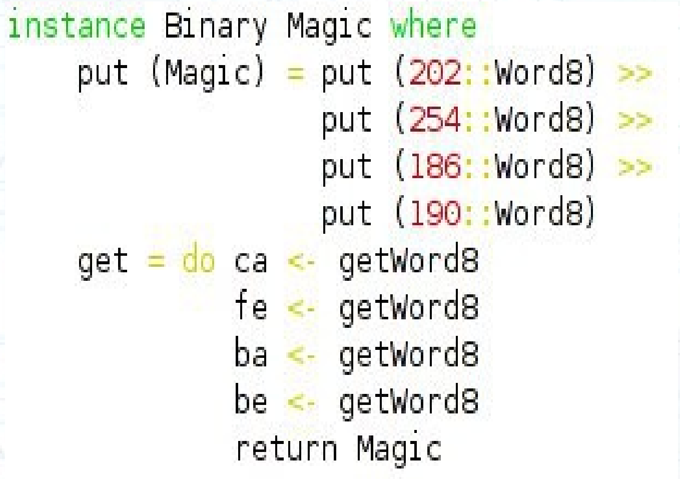
Ejemplo uso Binary - cont



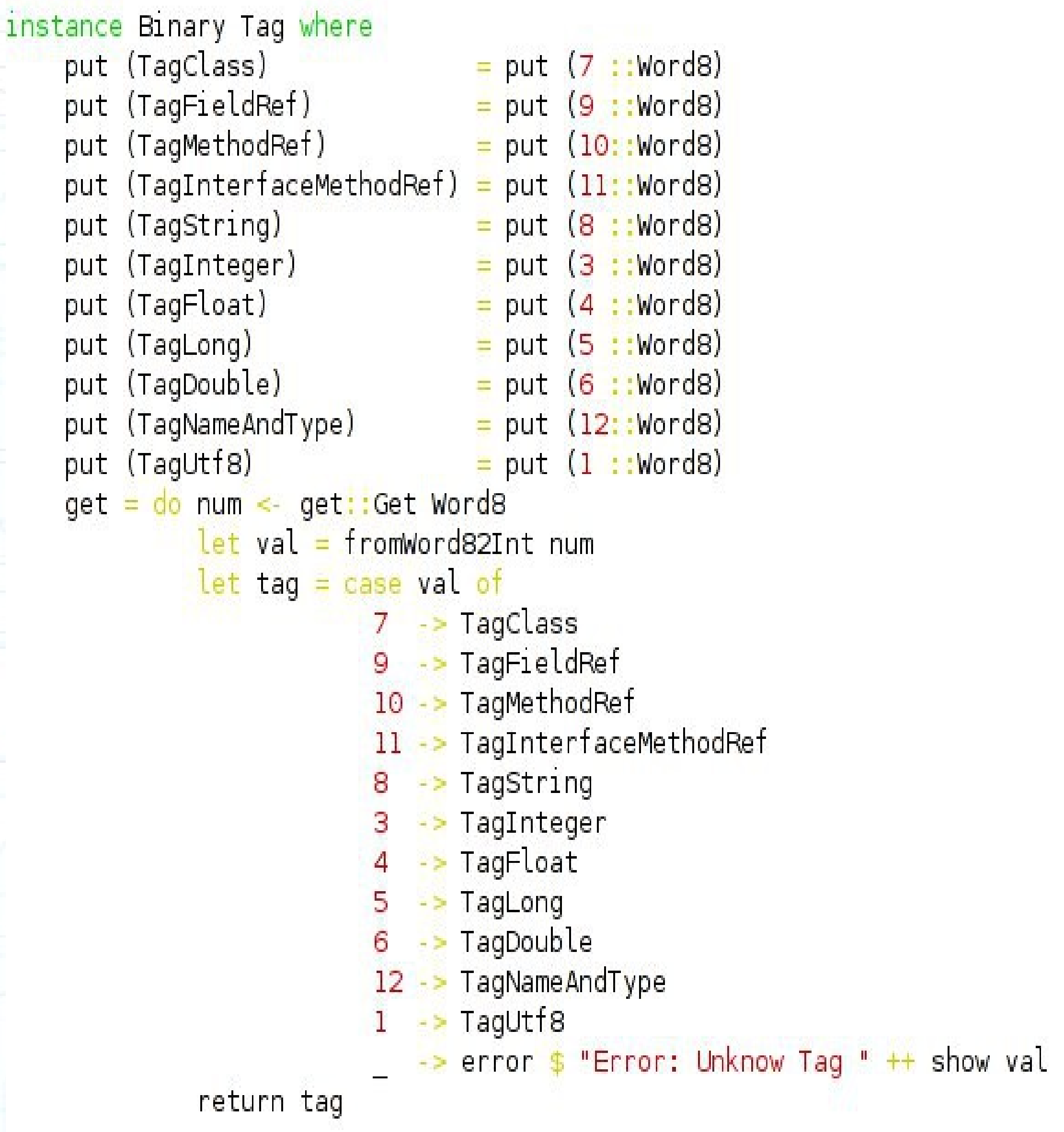
```
*Main> e
OpE "*" (IntE 7) (OpE "/" (IntE 4) (IntE 2))
*Main> encode e
Chunk "\SOH\NUL\NUL\NUL\NUL\NUL\NUL\NUL\SOH*\NUL\NUL\NUL\NUL\NUL\NUL\NUL\NUL\a\
SOH\NUL\NUL\NUL\NUL\NUL\NUL\NUL\SOH/\NUL\NUL\NUL\NUL\NUL\NUL\NUL\NUL\EOT\NUL\NU
L\NUL\NUL\NUL\NUL\NUL\NUL\STX" Empty
*Main> decode it :: Exp
OpE "*" (IntE 7) (OpE "/" (IntE 4) (IntE 2))
*Main> 
```

Serializador/Deserializador – 2da version

- Veamos Codigo



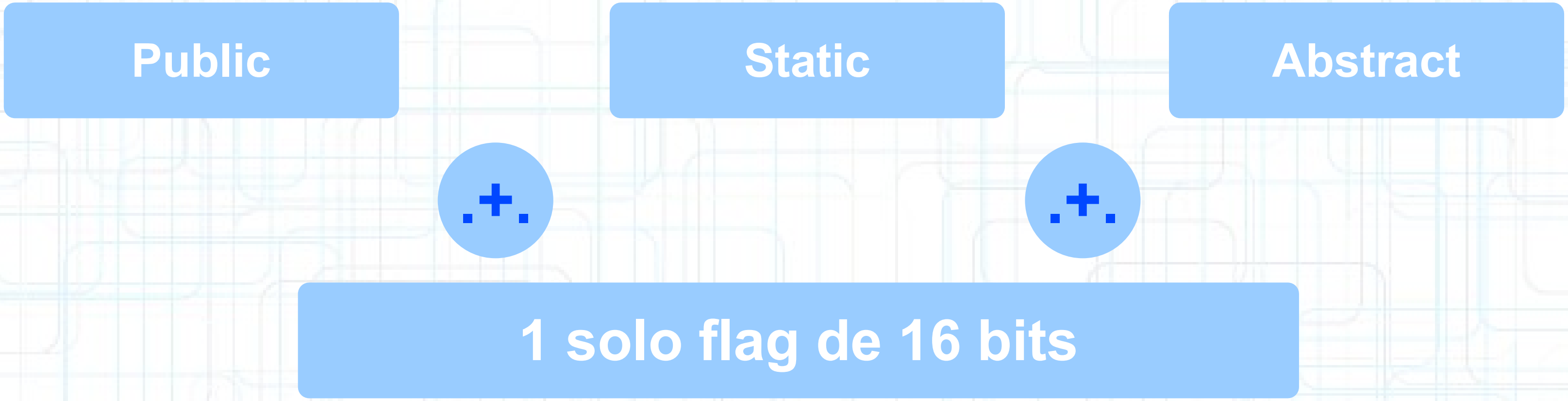
```
instance Binary Magic where
    put (Magic) = put (202::Word8) >>
                  put (254::Word8) >>
                  put (186::Word8) >>
                  put (190::Word8)
    get = do ca <- getWord8
             fe <- getWord8
             ba <- getWord8
             be <- getWord8
             return Magic
```

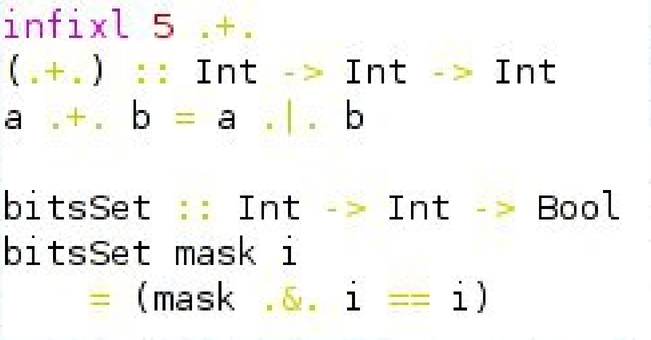


```
instance Binary Tag where
    put (TagClass)              = put (7 ::Word8)
    put (TagFieldRef)           = put (9 ::Word8)
    put (TagMethodRef)          = put (10::Word8)
    put (TagInterfaceMethodRef) = put (11::Word8)
    put (TagString)             = put (8 ::Word8)
    put (TagInteger)            = put (3 ::Word8)
    put (TagFloat)              = put (4 ::Word8)
    put (TagLong)               = put (5 ::Word8)
    put (TagDouble)             = put (6 ::Word8)
    put (TagNameAndType)        = put (12::Word8)
    put (TagUtf8)               = put (1 ::Word8)
    get = do num <- get::Get Word8
             let val = fromWord82Int num
             let tag = case val of
                         7  -> TagClass
                         9  -> TagFieldRef
                         10 -> TagMethodRef
                         11 -> TagInterfaceMethodRef
                         8  -> TagString
                         3  -> TagInteger
                         4  -> TagFloat
                         5  -> TagLong
                         6  -> TagDouble
                         12 -> TagNameAndType
                         1  -> TagUtf8
                         _  -> error $ "Error: Unknow Tag " ++ show val
             return tag
```

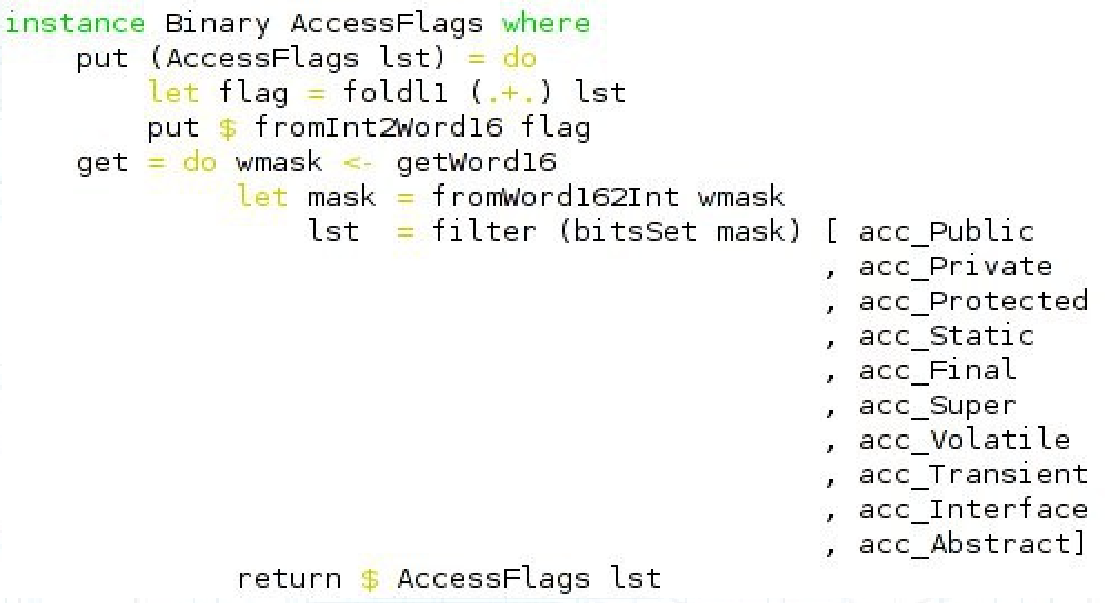
Serializador/Deserializador – 2da version

- Tratos Especiales, Flags





```haskell
infixl 5 .+.
(.+.) :: Int -> Int -> Int
a .+. b = a .|. b

bitsSet :: Int -> Int -> Bool
bitsSet mask i
    = (mask .&. i == i)
```



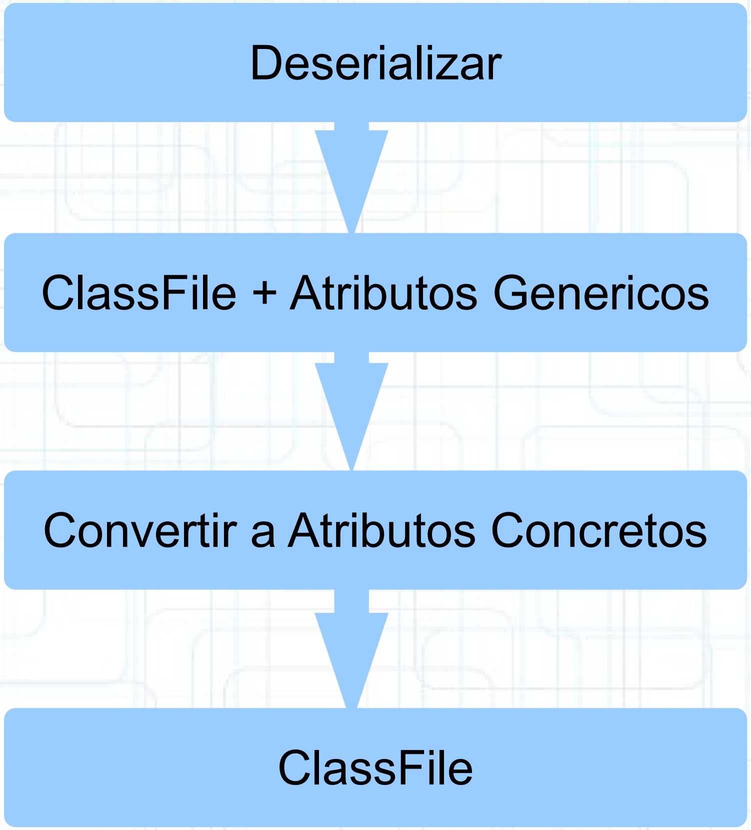
```haskell
instance Binary AccessFlags where
    put (AccessFlags lst) = do
        let flag = foldl1 (.+.) lst
        put $ fromInt2Word16 flag
    get = do wmask <- getWord16
             let mask = fromWord162Int wmask
                 lst  = filter (bitsSet mask) [ acc_Public
                                              , acc_Private
                                              , acc_Protected
                                              , acc_Static
                                              , acc_Final
                                              , acc_Super
                                              , acc_Volatile
                                              , acc_Transient
                                              , acc_Interface
                                              , acc_Abstract]
             return $ AccessFlags lst
```

Serializador/Deserializador – 2da version

- Tratos Especiales, Atributos

```
attribute_info {
        u2 attribute_name_index;
        u4 attribute_length;
        u1 info[attribute_length];
    }
```

Attribute_Generic



Deserializar

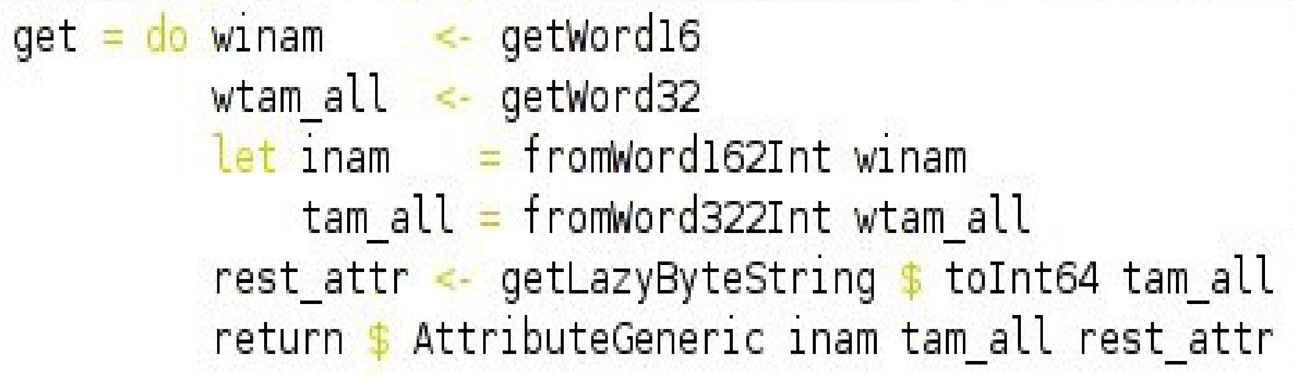
ClassFile + Atributos Genericos

Convertir a Atributos Concretos

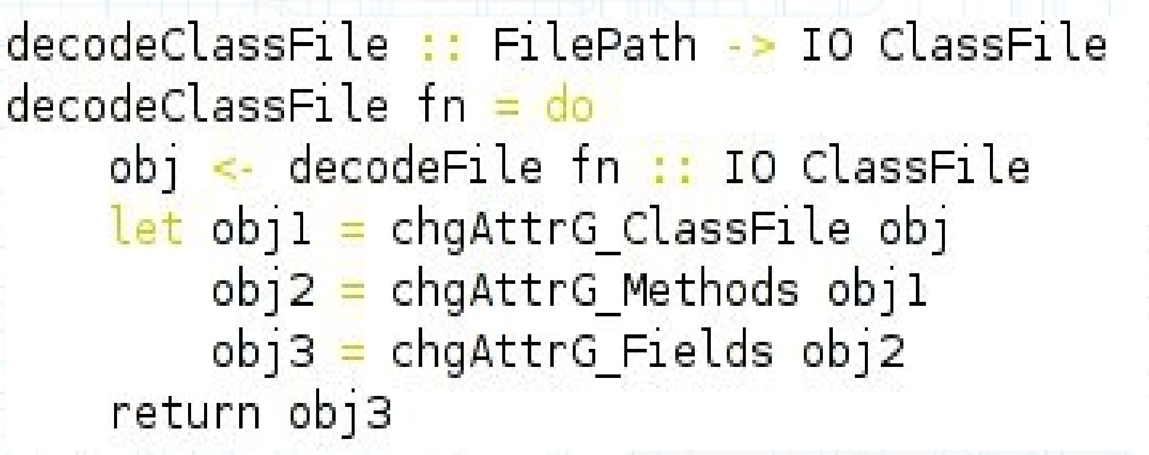
ClassFile

Serializador/Deserializador – 2da version

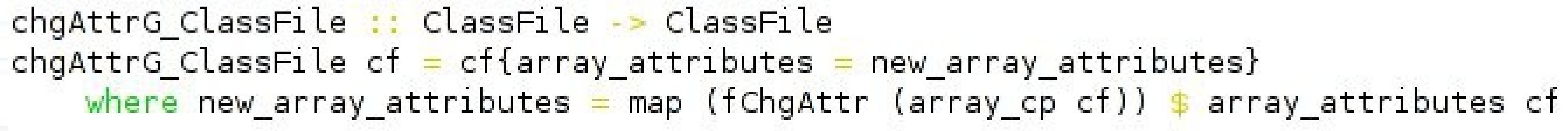
- Tratos Especiales, Atributos



```
get = do winam      <- getWord16
         wtam_all   <- getWord32
         let inam     = fromWord162Int winam
             tam_all  = fromWord322Int wtam_all
         rest_attr  <- getLazyByteString $ toInt64 tam_all
         return $ AttributeGeneric inam tam_all rest_attr
```



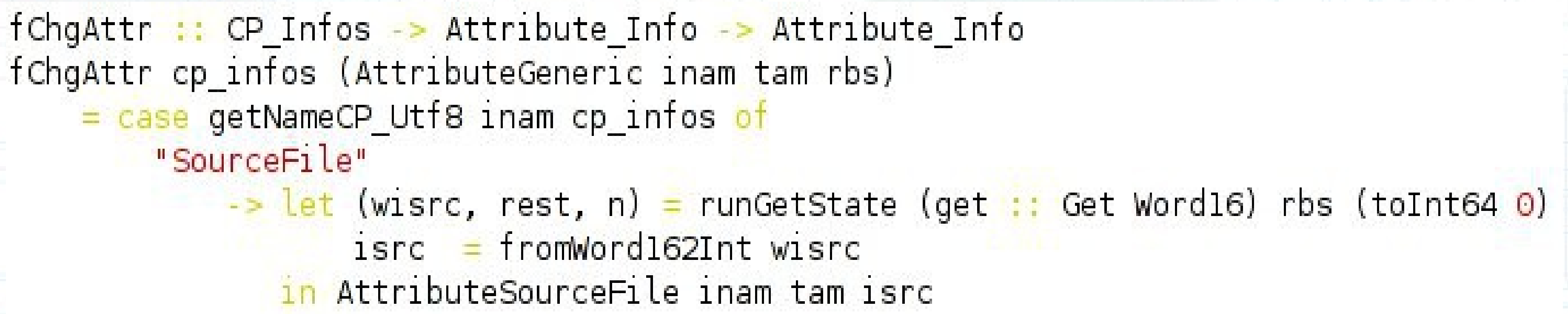
```
decodeClassFile :: FilePath -> IO ClassFile
decodeClassFile fn = do
    obj <- decodeFile fn :: IO ClassFile
    let obj1 = chgAttrG_ClassFile obj
        obj2 = chgAttrG_Methods obj1
        obj3 = chgAttrG_Fields obj2
    return obj3
```



```
chgAttrG_ClassFile :: ClassFile -> ClassFile
chgAttrG_ClassFile cf = cf{array_attributes = new_array_attributes}
   where new_array_attributes = map (fChgAttr (array_cp cf)) $ array_attributes cf
```

Serializador/Deserializador – 2da version

- Tratos Especiales, Atributos



```
fChgAttr :: CP_Infos -> Attribute_Info -> Attribute_Info
fChgAttr cp_infos (AttributeGeneric inam tam rbs)
    = case getNameCP_Utf8 inam cp_infos of
        "SourceFile"
            -> let (wisrc, rest, n) = runGetState (get :: Get Word16) rbs (toInt64 0)
                   isrc   = fromWord162Int wisrc
               in AttributeSourceFile inam tam isrc
```

Pruebas

- Vamos a la consola ...

Preguntas ?

Comunidad Haskell San Simon
===========================

Gracias por su Atencion!!!