Notes from 11/16

```haskell
{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE NoMonomorphismRestriction #-}
```

stack install parsec

```haskell
import Text.Parsec
```

```haskell
delims =
  many space >>
  many (many space >>
        char '[' >> delims >> char ']' >>
        many space) >> many space
```

public static volatile synchronized final void main

```haskell
data JAttribute
  = JPublic
  | JPrivate
  | JStatic
  | JFinal
  deriving (Show, Eq)
```

```haskell
jattribute =
  many space >>
  (try (string "public" >> return JPublic) <|>
    (string "private" >> return JPrivate) <|>
   (string "static" >> return JStatic) <|>
   (string "final" >> return JFinal))
```

```haskell
jattributes = many jattribute
```

---

visibility: public, or private, or protected, or default

transient or volatile, or default

static : yes or no final : yes or no synchronized : yes or no

```haskell
data JAttributes = JAttributes
  { isStatic :: Bool
  , isFinal :: Bool
  , isSynchronized :: Bool
  , visibility :: Maybe JVisibility
  , volatility :: Maybe JVolatility
  }
  deriving (Eq, Show)
data JVisibility
  = JVPublic
  | JVProtected
  | JVPrivate
  deriving (Eq, Show)
data JVolatility
  = JVTransient
  | JVVolatile
  deriving (Eq, Show)

defaultAttributes :: JAttributes
defaultAttributes =
  JAttributes{ visibility = Nothing
             , volatility = Nothing
             , isStatic = False
             , isFinal = False
             , isSynchronized = False
             }

parseFinal = do
  string "final"
  attrs <- getState
  if isFinal attrs then fail "duplicate final"
  else putState attrs{isFinal=True}

parseTransient = do
  string "transient"
  attrs <- getState
  case volatility attrs of
    Nothing -> putState attrs{volatility=Just JVTransient}
    Just JVTransient -> fail "duplicate 'transient'"
    Just JVVolatile -> fail "'transient' conflicts with 'volatile'"

parseVolatile = do
  string "volatile"
  attrs <- getState
  case volatility attrs of
```

```
      Nothing -> putState attrs{volatility=Just JVVolatile}
      Just JVVolatile -> fail "duplicate 'volatile'"
      Just JVTransient -> fail "'volatile' conflicts with 'transient'"


parseAnyAttribute =
  many space >>
  (parseFinal <|> parseTransient <|> parseVolatile)


parseAttributes =
  many space >> many parseAnyAttribute >> getState


runAttributes =
  runParser parseAttributes defaultAttributes ""
```

Parsing numbers

```
parseInteger = do
  neg <- optionMaybe (char '-')
  digits <- many1 (oneOf "0123456789")
  case neg of
    Nothing -> return digits
    Just _ -> return ("-" ++ digits)
```

.

```
main = return ()
```