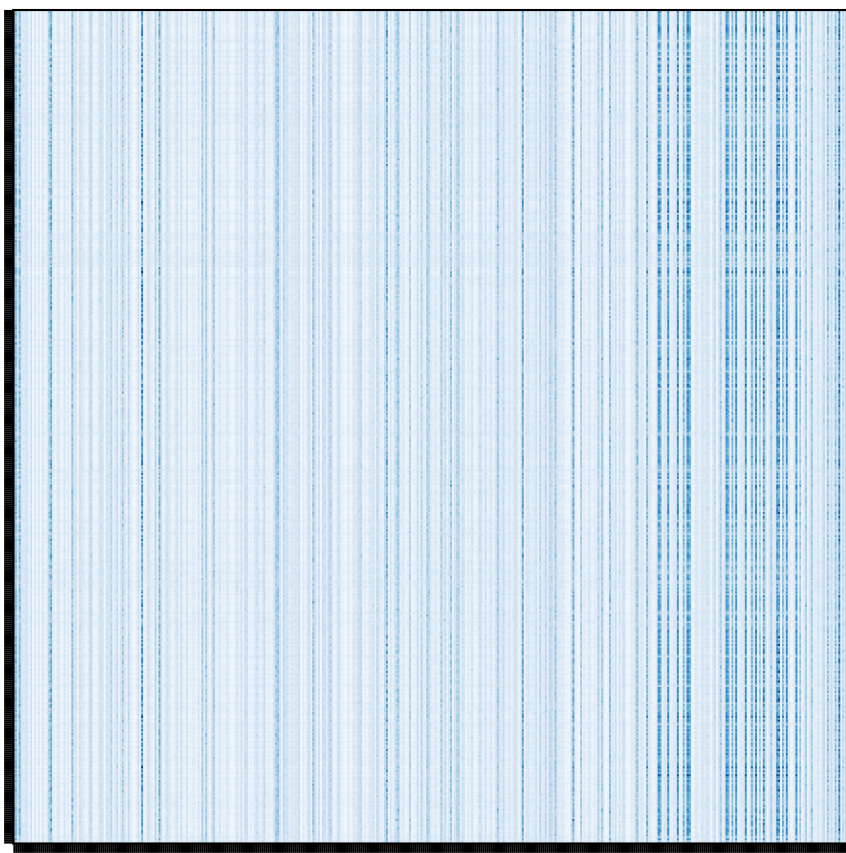


```

        parseInputStream(doc)
        ConstraintProcessor()
        true;input.markReadReferences()
        detectCharsetFromMetaCharset
        Charset.forName(defaultCharset).decode(firstBytes.toByteArray())
        getCharsetFromMetaCharset(meta.hasAttr("charset"))
        doc.childNodes().XMLDeclaration
        (decl.name().equals("xml:decl:"))
        !foundCharset.trim().isEmpty()
        Validate.notEmpty(charsetName)
        defaultCharset
        Charset.forName(charsetName);doc.outputSettings().charset(charsetName)
        readToValidate.isValid()
        ConstraintProcessor()
        getCharsetFromMetaCharset
        charsetPattern.matcher(content).validates()
        cs.toUpperCase(charsetName)
        StringBuilders.append(charsetName)
        {mime.append(mimeBoundaryChars[rand.nextInt(mimeBoundaryChars.length)])}
        {byteData.get(bom),buffer}
        BomCharset("U")
        BomCharset("U")
        BomCharset("U")
        BomCharset("U")
        charset,charset,j,j,j

```



```

        parseInputStream(doc)
        ConstraintProcessor()
        true;input.markReadReferences()
        detectCharsetFromMetaCharset
        Charset.forName(defaultCharset).decode(firstBytes.toByteArray())
        getCharsetFromMetaCharset(meta.hasAttr("charset"))
        doc.childNodes().XMLDeclaration
        (decl.name().equals("xml:decl:"))
        !foundCharset.trim().isEmpty()
        Validate.notEmpty(charsetName)
        defaultCharset
        Charset.forName(charsetName);doc.outputSettings().charset(charsetName)
        readToValidate.isValid()
        ConstraintProcessor()
        getCharsetFromMetaCharset
        charsetPattern.matcher(content).validates()
        cs.toUpperCase(charsetName)
        StringBuilders.append(charsetName)
        {mime.append(mimeBoundaryChars[rand.nextInt(mimeBoundaryChars.length)])}
        {byteData.get(bom),buffer}
        BomCharset("U")
        BomCharset("U")
        BomCharset("U")
        BomCharset("U")
        charset,charset,j,j,j

```