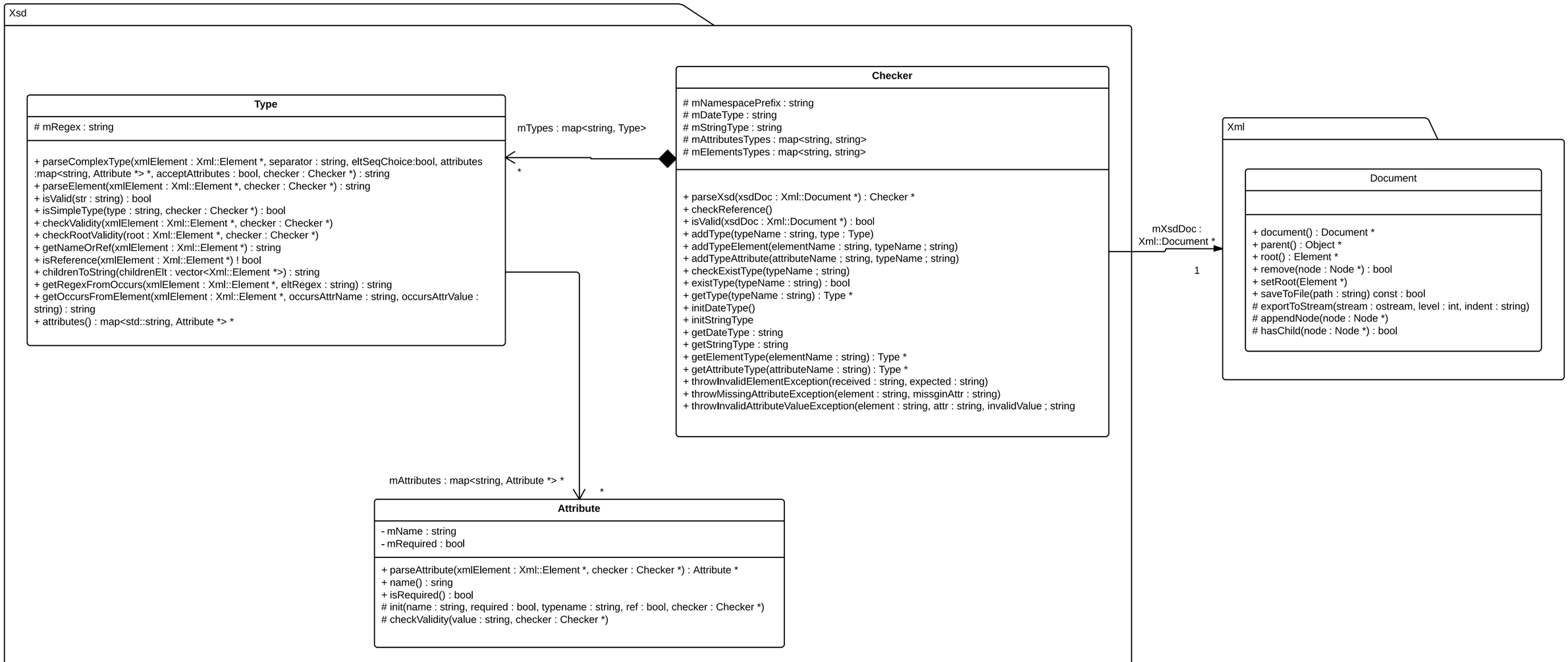
Xsd
Type
# mRegex : string
+ parseComplexType(xmlElement : Xml::Element *, separator : string, eltSeqChoice:bool, attributes :map<string, Attribute *> *, acceptAttributes : bool, checker : Checker *) : string
+ parseElement(xmlElement : Xml::Element *, checker : Checker *) : string
+ isValid(str : string) : bool
+ isSimpleType(type : string, checker : Checker *) : bool
+ checkValidity(xmlElement : Xml::Element *, checker : Checker *)
+ checkRootValidity(root : Xml::Element *, checker : Checker *)
+ getNameOrRef(xmlElement : Xml::Element *) : string
+ isReference(xmlElement : Xml::Element *) ! bool
+ childrenToString(childrenElt : vector<Xml::Element *>) : string
+ getRegexFromOccurs(xmlElement : Xml::Element *, eltRegex : string) : string
+ getOccursFromElement(xmlElement : Xml::Element *, occursAttrName : string, occursAttrValue : string) : string
+ attributes() : map<std::string, Attribute *> *
mTypes : map<string, Type>
*
Checker
# mNamespacePrefix : string
# mDateType : string
# mStringType : string
# mAttributesTypes : map<string, string>
# mElementsTypes : map<string, string>
+ parseXsd(xsdDoc : Xml::Document *) : Checker *
+ checkReference()
+ isValid(xsdDoc : Xml::Document *) : bool
+ addType(typeName : string, type : Type)
+ addTypeElement(elementName : string, typeName ; string)
+ addTypeAttribute(attributeName ; string, typeName ; string)
+ checkExistType(typeName ; string)
+ existType(typeName : string) : bool
+ getType(typeName : string) : Type *
+ initDateType()
+ initStringType
+ getDateType : string
+ getStringType : string
+ getElementType(elementName : string) : Type *
+ getAttributeType(attributeName : string) : Type *
+ throwInvalidElementException(received : string, expected : string)
+ throwMissingAttributeException(element : string, missginAttr : string)
+ throwInvalidAttributeValueException(element : string, attr : string, invalidValue ; string
mXsdDoc : Xml::Document *
1
Xml
Document
+ document() : Document *
+ parent() : Object *
+ root() : Element *
+ remove(node : Node *) : bool
+ setRoot(Element *)
+ saveToFile(path : string) const : bool
# exportToStream(stream : ostream, level : int, indent : string)
# appendNode(node : Node *)
# hasChild(node : Node *) : bool
mAttributes : map<string, Attribute *> *
*
Attribute
- mName : string
- mRequired : bool
+ parseAttribute(xmlElement : Xml::Element *, checker : Checker *) : Attribute *
+ name() : sring
+ isRequired() : bool
# init(name : string, required : bool, typename : string, ref : bool, checker : Checker *)
# checkValidity(value : string, checker : Checker *)