#include <iostream>

#include <cstdint>

#include "pugixml.hpp"

static const int32\_t nBufSize = 128;

static char szBuf[nBufSize] = { 0 };

// 写操作

void Write(const char \*szXmlFileName)

{

pugi::xml\_document xmlDoc;

pugi::xml\_node nodeRoot = xmlDoc.append\_child("root");

// 声明

pugi::xml\_node pre = xmlDoc.prepend\_child(pugi::node\_declaration);

pre.append\_attribute("version") = "1.0";

pre.append\_attribute("encoding") = "utf-8";

// 注释节点1

pugi::xml\_node nodeCommentStudents = nodeRoot.append\_child(pugi::node\_comment);

nodeCommentStudents.set\_value("all students info");

// 普通节点1

pugi::xml\_node nodeStudents = nodeRoot.append\_child("students");

for(int32\_t i = 0; i < 10; ++i)

{

sprintf\_s(szBuf, nBufSize, "student\_%02d", i);

pugi::xml\_node nodeStudent = nodeStudents.append\_child("student");

// 增加属性

nodeStudent.append\_attribute("name").set\_value(szBuf);

nodeStudent.append\_attribute("score").set\_value(100 - i);

}

// 注释节点2

pugi::xml\_node nodeCommentBooks = nodeRoot.append\_child(pugi::node\_comment);

nodeCommentBooks.set\_value("all books info");

// 普通结点2

pugi::xml\_node nodeBooks = nodeRoot.append\_child("books");

for(int32\_t i = 0; i < 10; ++i)

{

sprintf\_s(szBuf, nBufSize, "book\_%02d", i);

pugi::xml\_node nodeBook = nodeBooks.append\_child("book");

// 增加属性

nodeBook.append\_attribute("book").set\_value(szBuf);

nodeBook.append\_attribute("price").set\_value(50 - i);

}

xmlDoc.save\_file(szXmlFileName, "\t", 1U, pugi::encoding\_utf8);

}

// 读操作

void Read(const char \*szXmlFileName)

{

pugi::xml\_document xmlDoc;

if(!xmlDoc.load\_file(szXmlFileName, pugi::parse\_default, pugi::encoding\_utf8))

{

std::cout << "read " << szXmlFileName << " failed" << std::endl;

return;

}

xmlDoc.load\_file(szXmlFileName, pugi::parse\_default, pugi::encoding\_utf8);

pugi::xml\_node nodeRoot = xmlDoc.child("root");

// 读取第一个节点

for(pugi::xml\_node node = nodeRoot.child("students").first\_child(); node; node = node.next\_sibling())

{

std::cout << "\t" << node.attribute("name").value() << "," << node.attribute("score").value() << std::endl;

}

std::cout << std::endl;

// 读取第二个节点

for(pugi::xml\_node node = nodeRoot.child("books").first\_child(); node; node = node.next\_sibling())

{

std::cout << "\t" << node.attribute("book").value() << "," << node.attribute("price").value() << std::endl;

}

}

int32\_t main()

{

const char \*szXmlFileName = "info.xml";

Write(szXmlFileName);

Read(szXmlFileName);

std::cout << "finish" << std::endl;

getchar();

return 0;

}

<?xml version="1.0" encoding="utf-8"?>

<root>

<!--all students info-->

<students>

<student name="student\_00" score="100" />

<student name="student\_01" score="99" />

<student name="student\_02" score="98" />

<student name="student\_03" score="97" />

<student name="student\_04" score="96" />

<student name="student\_05" score="95" />

<student name="student\_06" score="94" />

<student name="student\_07" score="93" />

<student name="student\_08" score="92" />

<student name="student\_09" score="91" />

</students>

<!--all books info-->

<books>

<book book="book\_00" price="50" />

<book book="book\_01" price="49" />

<book book="book\_02" price="48" />

<book book="book\_03" price="47" />

<book book="book\_04" price="46" />

<book book="book\_05" price="45" />

<book book="book\_06" price="44" />

<book book="book\_07" price="43" />

<book book="book\_08" price="42" />

<book book="book\_09" price="41" />

</books>

</root>