

Departamento de Ciência da Computação – Universidade de Brasília
Programação Sistemática – 1º/2015
Data de entrega: 17/06/2015, 23:59h.

Trabalho Prático – Parte 2

Introdução:

A especificação para o trabalho prático da disciplina de PS em 1/2015 consiste em implementar a segunda etapa do sistema para auxiliar na coleta das informações necessárias para o Sucupira a partir dos currículos Lattes dos professores da Pós Graduação em Informática (PPGI) do CIC. O programa deve ser implementado em dois componentes: um que filtra as informações dos currículos Lattes e outro que processa essas informações em relatórios relativas a projetos e publicações dos professores em questão.

Parte 1 – Componente que processa os currículos Lattes

Funcionamento do componente

Essa parte do sistema deverá receber como entrada os currículos lattes de todos os professores do PPGI em XML e usando a biblioteca JDOM da mesma forma como adotado na primeira etapa do trabalho prático.

Além do processamento dos nomes dos docentes e seus respectivos projetos, o objetivo dessa parte deverá ser a extração das informações relativas às publicações em periódicos e eventos conforme descrito logo abaixo:

No currículo Lattes em XML das *publicações em eventos*:

```
<TRABALHO-EM-EVENTOS SEQUENCIA-PRODUCAO="81"><DADOS-BASICOS-DO-TRABALHO NATUREZA="COMPLETO" TITULO-DO-TRABALHO="Um Estudo Sobre o Impacto da Atenção Visual nas Redes Sociais" ANO-DO-TRABALHO="2013" PAIS-DO-EVENTO="Brasil" IDIOMA="Português" HOME-PAGE-DO-TRABALHO="" ...</TRABALHO-EM-EVENTOS>
```

Produzindo como resultado as seguintes informações no arquivo em formato txt:

PUBLICAÇÃO EM EVENTOS:

NATUREZA = "RESUMO_EXPANDIDO"

TITULO = "Reliability Analysis of Concurrent Systems using LTSA"

PAIS = "Estados Unidos"

IDIOMA = "Inglês"

NOME DO EVENTO = "International Conference on Software Engineering"

ANO = 2007

PROCEEDINGS = "Proceedings of 29th ICSE - Companion Volume"

VOLUME = ""

PAGINA INICIAL = "63"

PAGINA-FINAL = "64"

AUTORES:

1. "David S. Rosenblum"
 - a. Nome para citação = "ROSENBLUM, D. S."
 - b. Ordem de autoria = 2
2. "Jonas Wolf"
 - a. Nome para citação = "WOLF, J."
 - b. Ordem de autoria = 3
3. "Genaina Nunes Rodrigues"
 - a. Nome para citação = "RODRIGUES, G. N.;Rodrigues, Genaina Nunes;Rodrigues, Genaina Nunes;RODRIGUES, GENAÏNA N.;RODRIGUES, GENAINA;RODRIGUES, GENAINA N."
 - b. Ordem de autoria = 1

E das publicações em periódicos:

No currículo Lattes em XML das *publicações em periódicos*:

```
<ARTIGO-PUBLICADO SEQUENCIA-PRODUCAO="93" ORDEM-IMPORTANCIA=""><DADOS-BASICOS-DO-ARTIGO NATUREZA="COMPLETO" TITULO-DO-ARTIGO="Border Effect Analysis for Reliability Assurance and Continuous Connectivity of Wireless Sensor Networks in the Presence of Sensor Failures" ANO-DO-ARTIGO="2014" PAIS-DE-PUBLICACAO="" IDIOMA="Inglês" ...</ARTIGO-PUBLICADO>
```

Produzindo como resultado as seguintes informações no arquivo em formato txt:

PUBLICAÇÃO EM PERIODICO:

NATUREZA = "COMPLETO"

TITULO do ARTIGO = "Border Effect Analysis for Reliability Assurance and Continuous Connectivity of Wireless Sensor Networks in the Presence of Sensor Failures"

ANO = "2014"

IDIOMA = "Inglês"

TITULO do PERIODICO = "IEEE Transactions on Wireless Communications"

VOLUME = "13"

SERIE = ""

PAGINA INICIAL = "4232"

PAGINA FINAL = "4246"

AUTORES:

1. "Luiz Laranjeira"
 - a. Nome para citação = "LARANJEIRA, L."
 - b. Ordem de autoria = 1
2. "Genaina Nunes Rodrigues"
 - a. Nome para citação = "RODRIGUES, G. N."
 - b. Ordem de autoria = 2

Modularização

A implementação dessas novas funcionalidades devem aderir à implementação dos módulos criados na etapa 1 (parte 1) do trabalho prático relativos a: leitura dos XMLs, tratamento dos XMLs e de persistência dos dados.

Para cada um dos módulos desse componente, deverá ser criado um módulo de definição (*Interface*) e um módulo de implementação (*Class*). Além de serem aplicadas as devidas regras de encapsulamento do código. Lembrando que cada módulo deverá ser possível a compilação independente.

Documentação

Reorganize o seu programa adotando padrões conforme capítulo 4 e apêndice 5 do material bibliográfico da disciplina. Ao final da tarefa, garanta que seu programa esteja executando corretamente conforme testes realizados no trabalho prático 1.

Elabore um documento “.doc” ou “.pdf” que contenha os padrões de documentação que seu grupo adotou. Produza a documentação de todos os módulos utilizando o DoxyGen ou JavaDoc.

Controle de qualidade das funcionalidades

A corretude do código deve ser assegurada utilizando as ferramentas de análise estática e dinâmica. Para a análise estática, utilize a ferramenta **FindBugs**, seguindo a análise para faltas de controle, armazenamento, interface e entrada e saída.

Para o teste, continue utilizando o **JUnit**. Lembrem-se de criar um módulo controlador de teste (disciplinado) usando o JUnit para testar se as principais funcionalidades e restrições dos módulos de armazenamento, tratamento e persistência dos dados atendem a especificação. O teste disciplinado deve seguir os seguintes passos:

- 1) Antes de testar: produzir um roteiro de teste
- 2) Antes de iniciar o teste: estabelecer o cenário do teste
- 3) Criar um módulo controlador de teste, usando a ferramenta JUnit para testar as principais funcionalidades de cada módulo.
- 4) Ao testar: produzir um laudo em que todas as discrepâncias encontradas são registradas. Esse laudo pode ser uma saída da execução do JUnit. Somente termine o teste antes de completar o roteiro, caso observe que não vale mais a pena continuar executando o roteiro, uma vez que o contexto para o resto está danificado

Após a correção: repetir o teste a partir de 2 até o roteiro passar sem encontrar falhas.

Parte 2 Componente que processa as informações sobre os projetos.

Funcionamento do componente

A implementação dessa funcionalidade deve continuar a implementação dos módulos na etapa 1 do trabalho prático, sendo, no entanto, o foco na extração dos dados relativos às publicações em eventos e em periódicos.

Da mesma forma que os projetos, nessa etapa deverá ser feito o tratamento dos dados do arquivo de texto e feita a verificação de quais professores do departamento do CIC estão participando do artigo de periódico ou de um evento. Lembrando que o nome constado no arquivo de entrada pode ser qualquer um dos nomes presentes na lista de nomes para citação de cada professor. Deve-se gerar (em memória) dois módulos para manipular as seguintes listas: uma relativa às publicações em eventos e outra para publicações em periódicos.

Em particular, o campo **categoria** de cada publicação deve identificar se os integrantes são **docentes**, **discentes** do PPGI ou **participante externo**. Caso o nome do

autor não conste na lista de docentes ou discentes do PPGI, esse participante deve ser classificado como participante externo. Vide exemplo logo abaixo. Observem que a ordem de impressão dos autores deve seguir o campo **ordem de autoria** do arquivo de entrada.

Um segundo requisito importante para esse componente é ler os arquivos de entrada a partir de um diretório. Dessa forma, cada professor tem um arquivo que contém todas as publicações eventos em diretório. De forma análoga, esse procedimento se aplica às publicações em periódicos. Para tanto, utilizem a biblioteca `dirent.h` nativa ao GCC. A documentação dessa biblioteca pode ser encontrada em: <http://pubs.opengroup.org/onlinepubs/007908775/xsh/dirent.h.html>

Por fim, deve-se completar o grafo de relacionamentos dos projetos e publicações entre docentes do PPGI a partir dos integrantes de cada projeto e do autores de artigos.

Logo abaixo segue um exemplo do arquivo de saída para publicações em periódicos e em eventos respectivamente:

NOME COMPLETO = GENAINA NUNES RODRIGUES

PUBLICAÇÃO EM PERIÓDICO

TÍTULO = Border Effect Analysis for Reliability Assurance and Continuous Connectivity of Wireless Sensor Networks in the Presence of Sensor Failures

ANO DE PUBLICAÇÃO

ORDEM	AUTORES	CATEGORIA:
1	Luiz Laranjeira	Participante Externo
2	Genaina Nunes Rodrigues	Docente

NATUREZA: Completo

NOME DA EDITORA: IEEE Transactions on Wireless Communications

VOLUME: 13

SÉRIE:

NUMERO DA PÁGINA INICIAL: 4232

NUMERO DA PÁGINA FINAL: 4246

PUBLICAÇÃO EM EVENTOS

TITULO = Reliability Analysis of Concurrent Systems using LTSA

ANO = 2007

ORDEM	AUTORES	CATEGORIA
1	Genaina Nunes Rodrigues	Docente
2	Jonas Wolf	Participante Externo
3	David S. Rosenblum	Participante Externo

NATUREZA = RESUMO EXPANDIDO

PAIS = Estados Unidos

IDIOMA = Inglês

NOME DO EVENTO = International Conference on Software Engineering

PROCEEDINGS = Proceedings of 29th ICSE - Companion Volume

Volume: 8

Numero da página inicial: 63

Número da página final: 64

Modularização

A implementação dessas novas funcionalidades deve aderir à implementação dos módulos criados na etapa 1 (parte 2 em C) do trabalho prático relativos a: leitura dos dados dos arquivos de entrada, tratamento da informação, de persistência dos dados e de criação do grafo. Além de estendê-los, criar também os devidos módulos para publicações em eventos e periódicos, conforme solicitado anteriormente. Lembre-se de manter os princípios de modularidade, criando os devidos módulos de definição e implementação para cada módulo.

Documentação

Reorganize o seu programa adotando padrões conforme capítulo 4 e apêndice 5 do material bibliográfico da disciplina. Ao final da tarefa, garanta que seu programa esteja executando corretamente conforme testes realizados no trabalho prático 1.

Elabore um documento “.doc” ou “.pdf” que contenha os padrões de documentação que seu grupo adotou. Produza a documentação de todos os módulos utilizando o DoxyGen ou JavaDoc.

Controle de qualidade das funcionalidades

A corretude do código deve ser assegurada utilizando as ferramentas de análise **estática e dinâmica**. Para a análise estática, utilize a ferramenta Splint (C) seguindo a análise para faltas de controle, armazenamento, interface e entrada e saída.

Para o teste, continue utilizando o CUnit. Depois de prontos os casos de teste, deve-se criar um módulo controlador de teste (disciplinado) usando o CUnit para testar se as principais funcionalidades e restrições dos módulos de armazenamento, tratamento e persistência dos dados atendem à especificação. O teste disciplinado deve seguir os seguintes passos:

- 1) Antes de testar: produzir um roteiro de teste
- 2) Antes de iniciar o teste: estabelecer o cenário do teste
- 3) Criar um módulo controlador de teste, usando a ferramenta CUnit para testar as principais funcionalidades de cada módulo.
- 4) Ao testar: produzir um laudo em que todas as discrepâncias encontradas são registradas. Esse laudo pode ser uma saída da execução do CUnit. Somente termine o teste antes de completar o roteiro, caso observe que não vale mais a pena continuar executando o roteiro, uma vez que o contexto para o resto está danificado

Após a correção: repetir o teste a partir de 2 até o roteiro passar sem encontrar falhas.

Parte 2 - Escrita

Deverão ser entregues em conjunto com os códigos fonte:

- Um arquivo **LEIAME.TXT** contendo a explicação de como utilizar o(s) programa(s).
- Tantos arquivos **RELATORIO-nome.TXT** quantos forem os membros do grupo. O tema **nome** deve identificar o membro do grupo ao qual se refere o relatório. Estes arquivos devem conter uma tabela de registro de trabalho organizada como a seguir:

Data | Horas Trabalhadas | Tipo Tarefa | Descrição da Tarefa Realizada

Na descrição da tarefa redija uma explicação breve sobre o que o componente do grupo fez. Esta descrição deve estar de acordo com o Tipo Tarefa. Cada Tipo Tarefa identifica uma natureza de atividade que deverá ser discriminada explicitamente, mesmo que, durante uma mesma sessão de trabalho tenham sido realizadas diversas tarefas. Os tipos de tarefa são:

- Estudar
- Especificar os módulos
- Especificar as funções
- Revisar especificações
- Projetar
- Revisar projetos
- Codificar módulo
- Revisar código do módulo
- Redigir casos de teste
- Revisar casos de teste
- Realizar os testes
- Diagnosticar e corrigir os problemas encontrados

Observações:

Dica: Preencha esta tabela de atividades ao longo do processo. NÃO DEIXE PARA ÚLTIMA HORA, POIS VOCÊ NÃO SE LEMBRARÁ DO QUE FEZ TAL DIA, TAL HORA. Com relatórios similares a esse você aprende a planejar o seu trabalho.