PUCRS/FACIN
Algoritmos e Programação II
Trabalho Final
2014/1

Simuladores Discretos

1. Contexto

A empresa Simula&Emula desenvolveu um simulador para estudar a configuração ideal para supermercados. O sistema é um sucesso e a empresa conseguiu novos clientes interessados em simuladores discretos. Entre os novos clientes estão:

a) filas em caixas bancários;
b) *clusters* de servidores Web;
c) filas em farmácias;
d) filas de entradas em estádios de futebol;
e) filas em auto-escolas;
e) diversos outros sistemas com simulações similares.

Para atender tantos novos clientes a empresa deseja desenvolver um segundo simulador, com base no código do simulador de supermercados já existente. A implementação do primeiro e do segundo simulador deve ser fatorada, de forma a reduzir o custo de manutenção dos diversos simuladores que serão desenvolvidos em seguida.

Além das funções disponíveis no simulador do supermercado, todos os novos simuladores devem apresentar as mesmas funções. O simulador original deve também conter as novas funções, ou seja, as melhorias feitas no segundo simulador devem ser compatíveis com o primeiro simulador e todos os demais simuladores da empresa. Por exemplo, se o segundo simulador gerar relatórios detalhados de simulação, o primeiro simulador deve ser alterado para fazer o mesmo.

2. Descrição Geral do Trabalho

Este trabalho, que deverá ser feito individualmente, duplas, ou trios, consiste em modelar e implementar e manter dois sistemas que realizam simulações discretas. Leia atentamente o enunciado do problema, estude as classes existentes e projete as que deverão ser criadas.

O objetivo do trabalho é projetar, implementar e manter dois simuladores, que permitem determinar parâmetros, executar e coletar dados sobre a simulação. O trabalho deve destacar a aplicação dos conceitos estudados na disciplina.

O trabalho foi dividido em duas partes, de forma a favorecer e estimular uma prática adequada para a construção de sistemas. A seguir detalharemos cada uma dessas partes. Um relatório escrito de cada parte deve ser entregue pelos alunos.

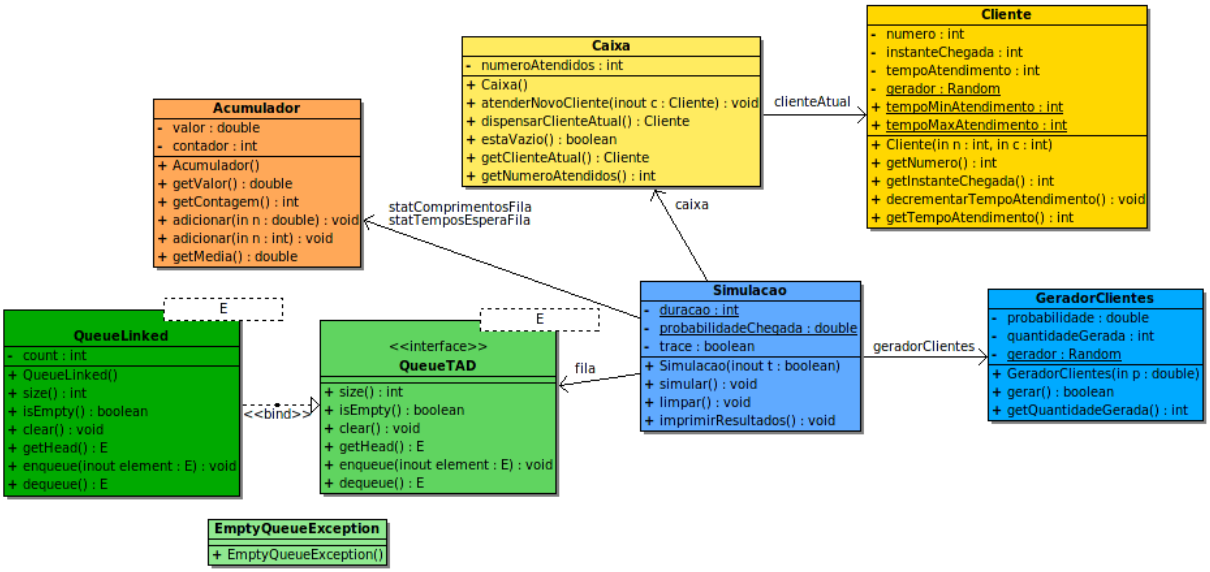

Figura 1 - Projeto de classes do atual simulador (Fonte: Cohen, 2012).

O enunciado do trabalho tenta recriar as condições de manutenção de sistemas, modelagem e reutilização de software no paradigma orientado a objetos. Além disso, o enunciado incentiva o desenvolvimento de algoritmos de maior complexidade, envolvendo listas e demais estruturas de dados estudadas na disciplina. A expectativa é que o desenvolvimento do simulador ocorra de forma sistemática, planejada e que as decisões de modelagem e programação sejam justificáveis.

2.1. Parte 1: Modelagem das classes e leitura do arquivo

A **primeira tarefa** é estudar a implementação atual do simulador (ver Figura 1 e código fonte fornecido).

A segunda tarefa é escolher o novo cliente para o qual será desenvolvido o novo simulador. O novo cliente pode ser uma proposta original ou um cliente da lista apresentada na seção 1. A nova simulação deve apresentar obrigatoriamente mais de um estágio de atendimento, ou seja, existem atendimentos que obrigam passar por mais de uma fila. Para exemplos, veja o apêndice.

A **terceira tarefa** é desenvolver um conjunto de classes e interfaces que permitam representar os dois sistemas.

A **quarta tarefa** é realizar a leitura do arquivo de dados da simulação. Escolher um ou mais formatos de arquivo: linguagem de simulação (ver GPSS), propriedade, JSON ou XML.

O arquivo deve ser compatível com um editor de textos simples. O formato do arquivo pode ser alterado por demanda de novos modelos de simulação. Alterações devem ser descritas nos relatórios.

Resumo da Parte 1:
Tarefa 1.1 - Estudar simulador do supermercado.
Tarefa 1.2 - Escolher cliente para o novo simulador.
Tarefa 1.3 - Criar classes e interfaces para o novo simulador.
Tarefa 1.4 - Escolher formato de arquivo e ler arquivo de entrada.

Entregas:
A. Artigo sobre a Parte 1 do trabalho com até 3 páginas em formato da SBC (SBC, 2010).
B. Diagrama com classes e interfaces propostas (pode estar no artigo)
C. Código fonte da versão preliminar dos sistemas.

2.2. Parte 2: Simulação e Estatísticas

O objetivo do simulador é permitir a descrição de uma configuração de simulação e a coleta de estatísticas.

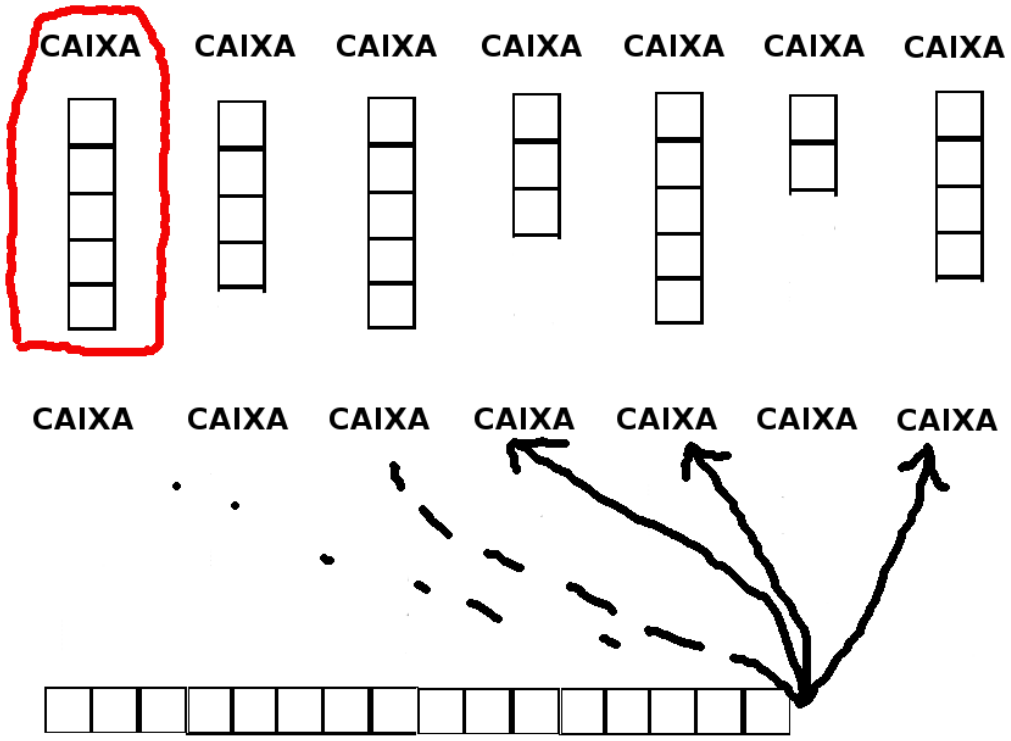

Figura 2 - Configurações de filas individuais ou fila única na simulação do supermercado.

A configuração mais simples é a conexão de uma ou mais filas a um ou mais atendentes.

Exemplo de "melhoria: pilha de documentos

Quando o atendimento ocorre em mais de uma etapa, um atendente passa um formulário para outro atendente. Porém, pode acontecer que os formulários sejam **empilhados** pelo segundo atendente. Nesse caso, a ordem de atendimento será **invertida**, pois o último cliente será atendido primeiro.

Resumo da Parte 2:
Tarefa 2.1 - Implementar simulador.
Tarefa 2.2 - Realizar testes com números variados de atendentes e filas.
Tarefa 2.3 - Propor e implementar melhorias como a pilha de documentos ou outra função específica do cliente.

Entregas:
D. Artigo sobre a Parte 1, Parte 2 e Extras com até 5 páginas em formato da SBC.
E. Diagrama com classes e interfaces implementadas.
F. Código fonte final.

2.3. Extras

A programação profissional adota práticas de desenvolvimento de software. Entre elas, o teste unitário, a gerência de configurações, qualidade de projeto de classes e documentação.

Resumo dos Extras:
Tarefa 3.1 - Escolher práticas que desejam adotar

Entregas:
G. Artefatos característicos da primeira prática
H. Artefatos característicos da segunda prática

3. Critérios de Avaliação

Leia com atenção os critérios de avaliação:

3.1 Pontuação

A pontuação obtida é limitada em 10 pontos, incluindo os pontos extras.

Parte 1 (Limite de 4 pontos)

Modelagem das classes: 2 pontos
Refatoração de classes. Uso de interfaces e superclasses. Uso de pacotes.

Parametrização da simulação: 2 pontos
Leitura de parâmetros de simulação a partir de arquivo de dados. Alteração da simulação por instanciação de objetos. Parâmetros: tempo de atendimento, tempo de espera na fila, tempo de geração.

Parte 2 (Limite de 5 pontos)

Estatísticas de atendentes: 1 ponto
Número de atendimentos por atendente, tempo médio de atendimento, utilização percentual.

Estatísticas de filas: 1 ponto
Tamanho máximo para cada fila, tempo médio de espera (com e sem atendimentos sem espera), atendimentos sem espera, tempo com fila vazia.

Estatísticas Avançadas para atendentes e filas: 1 ponto
Mediana, mínimo, máximo e desvio padrão. Ordenar dados para obter valores.

Pilha de documentos ou outra estratégia: 1 ponto

Interface de usuário gráfica: 1 ponto
Representar graficamente filas, atendentes e estados.

Gráficos das estatísticas: 1 ponto
Gerar gráficos das estatísticas de filas ao longo do tempo

Extras (Limite de 2 pontos)

Regras de projeto de classe (PMD): 1 ponto
Utilizar conjunto de regras completo do PMD. Justificar regras que não foram atendidas. Adotar: DRY, SOLID, KISS, YAGNI (ver Wikipedia).

Testes automáticos (JUnit): 1 ponto
Cobertura de mais de 70% do código-fonte com testes automáticos em ambos simuladores.

JavaDoc e Convenções de Código: 1 ponto
Atendimento do Java Code Conventions em mais de 70% das classes e interfaces.

Repositório (git ou hg): 1 ponto
Registro de mais de três semanas de trabalho em repositório no GitHub, BitBucket ou outros.

3.2 Código

A implementação deve seguir o Java Code Conventions para nomes de identificadores e estrutura das classes. Não serão aceitos trabalhos com erros de compilação. Programas que não compilarem corretamente terão nota ZERO automaticamente.

3.3 Testes

Serão utilizados dados de teste nos dois simuladores para verificar se o sistema funciona corretamente.

3.4 Entregas e Apresentações

Os trabalhos são individuais, em duplas ou trios. Se o trabalho for realizado por **duplas** ou **trios**, a comprovação da **contribuição individual** deverá ser **obrigatoriamente** feita através de controle de versão, com um repositório remoto.

O diretório contendo o projeto completo do Eclipse, NetBeans ou IntelliJ deve ser compactado no formato .zip e submetido via Moodle até as datas e horas especificadas no cronograma de sua turma.

O projeto deve conter um arquivo LEIAME com os nomes completos dos integrantes do grupo. Somente um integrante do grupo deve enviar o trabalho.

Datas de entrega das partes do trabalho:

Inscrição: 30/05/2014
Informar cliente e repositório

Parte 1: 09/06/2014

Esta parte deverá apenas ser entregue pelo Moodle, sem apresentação.

Parte 2: 07/07/2014

Esta parte deverá ser entregue até o horário de início da aula (impreterivelmente, não haverá adiamentos), e será apresentada ao professor(a) durante a aula.

Trabalhos entregues, mas não apresentados, terão sua nota anulada automaticamente. Durante a apresentação será avaliado o domínio da resolução do problema, podendo inclusive ser possível invalidar o trabalho quando constatada a falta de conhecimento sobre o código implementado.

A cópia parcial ou completa do trabalho terá como conseqüência a atribuição de nota ZERO ao trabalho dos alunos envolvidos. A verificação de cópias é feita inclusive entre turmas.

Referências

COHEN, M. (2012) Tipos Abstratos de Dados - Filas e Pilhas.
http://www.inf.pucrs.br/flash/alpro2/present/U09_TAD/08-TAD-filapilha/#22

COHEN, M. (2012b) Simulador de supermercado.
http://www.inf.pucrs.br/flash/alpro2/present/U09_TAD/08-TAD-filapilha/supermercado.zip

ORACLE (2014) Code Conventions for the Java Programming Language.
http://www.oracle.com/technetwork/java/index-135089.html

PMD (2014). PMD Source Code Analyzer.
http://pmd.sourceforge.net/

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Templates SBC.
http://www.sbc.org.br/en/index.php?option=com_jdownloads&Itemid=195&task=finish&cid=38&catid=32

WIKIPEDIA (2014) GPSS.
http://en.wikipedia.org/wiki/GPSS

Apêndices

A Exemplos de arquivos

Formato GPSS-like

```
=-=
SIMULATE

*
* Atendentes
*
        CLERKS 1,STAGE1   ; apenas um atendente, estágio 1
        TIME 10,5         ; atendimento dura 10 +/- 5 minutos

        CLERKS 3,STAGE2   ; 3 atendentes, estágio 2
        TIME 13,6         ; atendimento dura 13 +/- 6 minutos
*
* Filas
*
```

```
        QUEUES 1,STAGE1  ; apenas uma fila, estágio 1
        GENERATE 18, 6   ; chegadas na fila a cada 18 +/- 6 minutos

        QUEUES 2,STAGE2  ; duas filas, estágio 2

END
=-=
```

Formato Properties

```
=-=
#
# Atendentes
#
        clerks=1
        stage=1
# apenas um atendente
        clerksgenmean= 10
        clerksgendvt= 5
#atendimento dura 10 +/- 5 minutos
#
# Filas
#
        queues= 1
        stage=1
# apenas uma fila
        queuesgenmean=18
        queuesgendvt=6
# chegadas na fila a cada 18 +/- 6 minutos

        queues= 2
        stage=2
# duas filas
=-=
```

B Exemplos de simulações

Cliente: agência bancária

Existe uma fila na entrada para distribuição de senhas. Ao solicitar a senha, o usuário pode ser selecionado para dois setores diferentes: caixa normal e atendimento por gerente. Cada setor, por sua vez, tem duas filas: normal e preferencial. Portanto, para ser atendido o usuário deverá entrar em duas filas (dois estágios).

Cliente: cluster de servidores

A requisição web chega no cluster em uma fila única, e de lá a requisição é direcionada para um dos servidores web que fazem o atendimento (há 4 servidores). Cada servidor web gera solicitações de acesso a banco de dados, e há dois servidores de banco de dados disponíveis. Portanto, há uma fila geral, uma fila para cada servidor web e uma fila para cada servidor de banco de dados (três estágios).

Cliente: auto-escola

Todos os usuários retiram a senha em uma fila única, o usuário é chamado para atendimento (2 guichês). O usuário entrega a documentação, e aguarda ser chamado novamente. Ele é chamado quando o documento solicitado estiver disponível (o que leva de 2 a 15 minutos). Os documentos são empilhados, e um guichê adicional faz a devolução.