Trabalho Final (2015/2)

Disciplina de Técnicas de Programação

A) Horário de entrega: 21:15 de 01/12/2015
Apresentações: 01/12/2015

B) Objetivo:

O objetivo deste trabalho é consolidar o conhecimento sobre conceitos e construção de sistemas empresariais orientados a objetos em arquiteturas multicamadas através da exploração dos tópicos discutidos na disciplina de Técnicas de Programação.

C) Enunciado do problema:

Uma empresa está interessada no desenvolvimento de um sistema de operação e controle de parquímetros. Esse sistema é composto de dois módulos:

- Módulo Operacional – oferece toda a funcionalidade para a operação de parquímetros; é distribuído de forma embarcada nos parquímetros, sendo assim, não é de sua responsabilidade a implementação de uma interface gráfica;
- Módulo Gerencial – oferece funcionalidade de gerência sobre os parquímetros; é fornecido através de um programa desktop.

Módulo Operacional:

Ao estacionar seu veículo nas vagas gerenciadas por parquímetros, o usuário deverá adquirir um tíquete de estacionamento de acordo com o tempo escolhido de permanência do veículo (o tempo mínimo, incremento, tempo máximo e valores tarifados são customizáveis no sistema). Um tíquete de estacionamento deve possuir as seguintes informações impressas:

- Número de identificação do parquímetro (5 dígitos de tamanho fixo);
- Endereço do parquímetro;
- Número serial do tíquete emitido (5 dígitos de tamanho fixo);
- Data e hora de emissão do tíquete;
- Data e hora de validade do tíquete.

Atualmente, os parquímetros estão atuando com as seguintes configurações (o sistema deve suportar os valores de forma totalmente configurável):

- Início do horário de tarifação: 8h30min;
- Final do horário de tarifação: 18h30min;
- Tempo mínimo: 30 minutos (valor atual R$ 0,75);
- Tempo máximo: 120 minutos (valor atual R$ 3,00);
- Incremento: 10 minutos;
- Tarifa por fator de incremento: R$ 0,25.

O pagamento do tíquete de estacionamento pode ser realizado de maneiras flexíveis e configuráveis. Atualmente o sistema suporta os seguintes modos:

- Pagamento com moedas – o tipo de moeda aceito é configurável no sistema; o parquímetro deve possuir a capacidade de devolver o troco em moedas no valor correspondente desde que possua a quantidade e tipo

de moedas necessária em seu repositório, caso contrário, deverá retornar o maior valor possível menor que o troco devido.
- Pagamento com cartão recarregável – o valor do tíquete deve ser debitado do cartão recarregável e caso não exista saldo suficiente, o tíquete não deve ser emitido. Todo cartão recarregável possui um código único representado por uma sequência de 128 caracteres.
- Pagamento com cartão residente – o uso do cartão isenta o pagamento da tarifa. Todo cartão recarregável possui um código único representado por uma sequência de 128 caracteres.

Todas as operações realizadas pelo parquímetro eletrônico são registradas e transformadas em relatórios no momento da emissão do tíquete. A fim de otimizar o gerenciamento, estas informações podem ser obtidas a qualquer momento por meio dos coletores de dados portáteis. Sendo assim, o sistema deve possuir uma forma de exportar os dados operacionais em um arquivo textual.

Fisicamente, o parquímetro apresenta dois mostradores digitais: um com o dia e horário atual e outro de propósito geral que permite observar informações como o horário de validade do tíquete, o valor total a ser pago pelo tíquete e o valor total de moedas inseridas. Para interação com o usuário, o parquímetro possui os seguintes botões:
- Botão +: para incrementar o período de validade do tíquete;
- Botão -: para decrementar o período de validade do tíquete;
- Botão verde: para concluir a operação e imprimir um tíquete;
- Botão vermelho: para cancelar a operação e devolver as moedas (se for o caso).

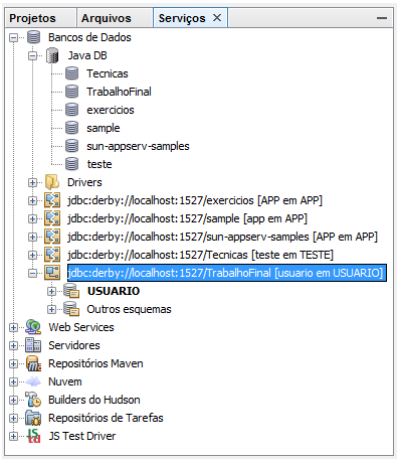
Módulo Gerencial:

O módulo gerencial é o responsável pela geração de relatórios e estatísticas dos dados de arrecadação dos parquímetros. Isso permite que a empresa responda de forma ágil e precisa a todas as solicitações de relatórios informativos feitas pela administração municipal.
O módulo apresenta uma interface gráfica desktop que permite realizar as seguintes operações:
- Importar arquivos de *logging* dos parquímetros;
- Inserir dados importados na base de dados;
- Gerar relatório com todos os dados do *log* dos parquímetros filtrados por dia específico ou mês;
- Gerar estatísticas de total de valor arrecadado, total de valor isento, filtrados por número do parquímetro e agrupados por mês ou ano;
- Gerar gráfico estatístico (sugere-se o uso do http://www.jfree.org/jfreechart/) de barras de total de valor arrecadado por um parquímetro dentro de um período anual;
- Gerar gráfico estatístico de pizza de percentual de total de valor arrecadado e total de valor isento de todos parquímetros dentro de um período de mês de início e fim.

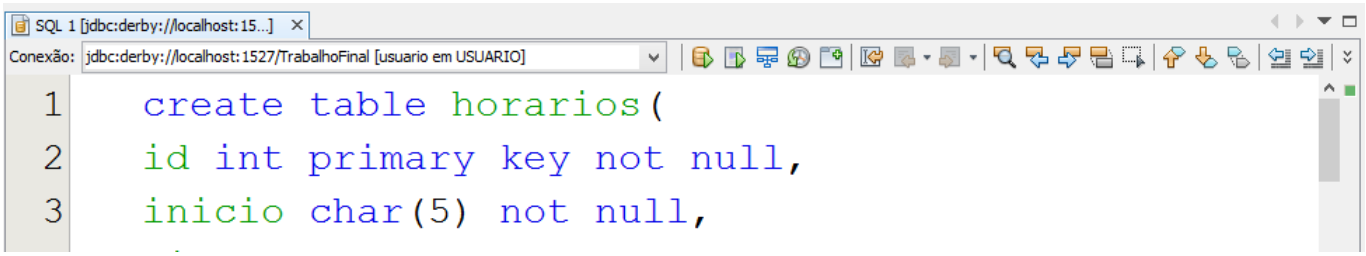
D) Base de dados:

O trabalho, em especial o modo gerencial, deve fazer uso de uma base de dados externa, ou seja, não embutida, de modo a disponibilizar o requisito cliente-servidor do sistema. Importante: o driver JDBC a ser utilizado se encontra no pacote “derbyclient.jar”.

Siga os seguintes passos para configurar um servidor externo de banco de dados JavaDB (Derby).
- Inicie o Netbeans.
- Clique na aba “Serviços”. Localize os nodos “Bancos de Dados - JavaDB” e com o botão direito do mouse selecione a opção “Iniciar Servidor” para inicializar o servidor de banco de dados. Depois de inicializado, clique com o botão direito e selecione a opção “Criar Banco de Dados...” para criar uma nova base de dados com o nome “TrabalhoFinal”. Configure um usuário (“usuario”) com senha (“senha”). Como resultado, será criado um link de conexão com o banco de dados. Dê um duplo clique sobre ele para estabelecer uma conexão e verificar se está tudo correto.



- Com a conexão estabelecida, clique com o botão direito sobre a mesma e selecione o menu “Executar Comando...”. Na janela que se abriu, copie e cole o script de criação das tabelas. Para executar o script, clique no primeiro botão ao lado do menu drop-down (é o botão com uma seta verde sobre o ícone de base de dados).



E) Requisitos:

Os seguintes itens são obrigatórios na implementação do sistema:

- Arquitetura multicamada (pelo menos 3);
- Uso dos padrões de projeto explorados em sala de aula, sendo obrigatoriamente:
  - Uso do padrão “MVC” na camada de interface gráfica;
  - Uso do padrão “Facade” para isolar a camada de domínio da camada de interface gráfica;
  - Uso do padrão arquitetural “Domain Model” na camada de domínio;
  - Uso do padrão “DAO” na camada de persistência.
    - A camada de persistência deve ser implementada sem a utilização de frameworks mapeadores objeto/relacional (como JPA, Hibernate, etc);
- Implementação em Java;
- Tratamento correto do encapsulamento de exceções entre as camadas.

Para o módulo operacional:

- Módulo JAR independente do módulo gerencial;
- Comprovação da funcionalidade via teste unitário;
- A utilização de uma interface gráfica simulando o parquímetro físico é opcional.

Para o módulo gerencial:

- Módulo JAR independente do módulo operacional;
- Interface gráfica de usuário (interface textual de console não será aceita);
- Persistência em banco de dados relacional;
- O banco de dados deverá ter sido previamente populado (um arquivo contendo os scripts para geração do BD devem ser entregues juntamente com o código fonte) com, no mínimo, os valores necessários para uma boa cobertura de casos de teste.

Ponto Extra:

- Monitoramento dos equipamentos através de mapa geo-referenciado no módulo gerencial.

F) Desenvolvimento, apresentação e avaliação do trabalho:

- O trabalho pode ser realizado individualmente ou em grupos de, no máximo, 3 alunos.
- Os trabalhos serão apresentados no laboratório. Durante a apresentação, TODOS os alunos devem estar presentes e aptos a responder às perguntas. Respostas insatisfatórias por um aluno ou a sua ausência acarretarão descontos na nota final.
- A apresentação do trabalho é de inteira responsabilidade dos alunos (configuração da máquina, do ambiente de software, banco de dados, etc.) e o código-fonte utilizado deverá ser o mesmo entregue ao professor. É tarefa do grupo garantir que o sistema esteja apto a ser executado no dia da apresentação.
- Sistemas que não consigam ser executados ou apresentados no dia da apresentação receberão nota zero.
- Mensagens de erro apresentadas durante a execução do programa, mesmo que a aplicação não pare de executar, serão consideradas como erros de execução, e acarretarão descontos na nota do trabalho.
- Em caso de erro de sintaxe (compilação), o peso final do trabalho será valorado em zero.
- Em caso de erro de semântica (conteúdo), o peso final do trabalho sofrerá uma redução.
- Os trabalhos serão avaliados de acordo com critérios a serem estabelecidos pelo professor da disciplina, considerando o que é pedido no enunciado e o que foi realizado com sucesso pelo sistema. Também serão avaliadas a modelagem do sistema (correta criação das classes necessárias, com seus atributos e métodos, encapsulamento, e correto estabelecimento de relações entre as classes) e sua implementação de acordo com os conceitos de orientação a objetos e arquitetura multicamada.
- A comprovação do uso de teste unitário, padrões de projeto e de programação por contratos será levada em conta na avaliação do trabalho.
- ***Trabalhos copiados resultarão em nota zero para todos os alunos envolvidos.***

G) Entrega do trabalho:

- Todos os arquivos necessários a execução do sistema, bem como os arquivos-fonte, scripts de banco de dados e os arquivos de documentação, deverão ser empacotados em um único arquivo (.zip) e submetidos através do sistema Moodle até a data de entrega.
- Devem fazer parte da documentação pelo menos:
    - Diagrama de classes do sistema. O diagrama de classes deverá ser entregue junto com documentação em texto salientando os pontos onde foram utilizados padrões de projeto na implementação da solução. É importante que as classes estejam agrupadas em pacotes de acordo com a arquitetura de camadas do sistema. Diagramas gerados diretamente via engenharia reversa de código, sem qualquer organização lógica, serão desconsiderados na avaliação.
    - Diagrama relacional da base de dados.
    - Os diagramas devem estar disponíveis em imagens com resolução suficiente e de fácil visualização. Não serão aceitos diagramas que estejam em formato original da ferramenta de desenho (como Visio, Astah, e outros).
    - Casos de teste utilizados no teste unitário (se for o caso).
- Não serão aceitos trabalhos enviados por correio eletrônico.
- Não serão aceitos trabalhos enviados fora do prazo estabelecido.