

Oprogramowanie Systemów Medycznych

Zadanie 2

Temat: Tworzenie aplikacji z połączeniem do bazy danych

Termin oddania: 23.59 12.05.2013

Cel

Celem zadania jest rozwinięcie aplikacji zaprojektowanej w ramach zadania 1, poprzez dodanie bazy danych, która będzie przetrzymywała wprowadzone do rejestru dane. Zadanie 2 ma również sprawdzić umiejętność tworzenia własnych komponentów i wykorzystania prostych funkcji java2d.

Działanie

Na rysunku poniżej przedstawiono schemat interfejsu użytkownika. Składa się on z dwóch części:

1. okna głównego aplikacji *Rejestr Pacjentów*, której działanie zostało omówione w instrukcji do zadania 1. Interfejs graficzny został wzbogacony o dodatkową funkcję - filtrowania tablicy pacjentów. Wpisanie części imienia lub nazwiska w pole tekstowe i naciśnięcie przycisku *Szukaj* powinno spowodować ograniczenie listy do elementów spełniających kryteria wyszukiwania. W przypadku pozostawienia pola tekstowego pustego i naciśnięciu przycisku *Szukaj*, użytkownik powinien zobaczyć pełną listę pacjentów. Lista powinna być sortowana w sposób alfabetyczny. Dodatkowo, interfejs posiada przycisk *Karta Pacjenta*, wywołujący okno dialogowe opisane poniżej.
2. okna dialogowego będącego *Kartą Pacjenta*, zawierającą pomiary temperatury. Okno dialogowe powinno być oknem modalnym względem głównego okna aplikacji (tj. wyświetlenie karty pacjenta blokuje okno *Rejestru Pacjentów*). *Karta Pacjenta* składa się z tabeli prezentującej wprowadzone pomiary, wykresu będącego graficzną reprezentacją zmienności temperatury pacjenta w czasie. Tabela powinna być posortowana wg dat wprowadzania kolejnych pomiarów. *Karta Pacjenta* zawiera również panel z kontrolkami pozwalającymi na wprowadzanie nowych rekordów. Wypełnienie pól formularza i naciśnięcie przycisku *Dodaj*, powinno aktualizować zarówno tabelę jak i wykres.

Dane wprowadzane do aplikacji mają charakter trwały, tzn. stan aplikacji jest zapamiętywany pomiędzy kolejnymi uruchomieniami. Za równo *Rejestr Pacjentów* jak i *Karta Pacjenta*, korzystają z bazy danych, która jest zintegrowana z aplikacją.

Rejestracja pacjentów

Aplikacja ...

Zamknij CTRL+X

Dane pacjenta

Imię:

Nazwisko:

Płeć

☐ Kobieta
☐ Mężczyzna

Ubezpieczony

☐

Zawód:

Student

Pracownik fizyczny

Pracownik umysłowy

Imię i Nazwisko:

Szukaj

Imię i Nazwisko	Płeć	Ubezpieczony	Zawód
Jan Kowalski	M	<input checked="" type="checkbox"/>	Student
Magdalena Kwiatkowska	K	<input checked="" type="checkbox"/>	Pracownik

Zapisz

Anuluj

Karta pacjenta

Dodaj

Usuń

Naciśnięcie przycisku szukaj filtruje listę

Karta Pacjenta

Pacjent: Jan Kowalski

Data	Temperatura
01/04/2012	36.6
02/04/2012	36.7
03/04/2012	36.9
04/04/2012	36.6

Data: / /

Temp.: 36.6

Dodaj

Okno dialogowe po naciśnięciu przycisku

created with Balsamiq Mockups - www.balsamiq.com

Szczegóły implementacji

Jako RDBMS danych należy użyć silnik Java DB [1] podłączony w trybie „embedded” (zintegrowana z aplikacją baza danych – tak jak pokazane w przykładzie w materiałach wykładowych). Baza powinna posiadać co najmniej dwie tabele trzymające dane z pacjentów oraz powiązanych z nimi pomiarów temperatury.

Wprowadzanie pomiarów temperatury wymaga podania daty. Możliwe jest skorzystanie z gotowego komponentu JCalendar [2] .

Wykres temperatur powinien zostać stworzony jako własny komponent (specjalizacja klasy JComponent) i wykorzystywać możliwości pakietu java2d do odwzorowania danych pomiarowych. Oś odciętych powinna przedstawiać kolejne

dni miesiąca, oś rzędnych – skalę temperatur z zakresu 32-45 stopni. Za równo wykres jak i tabela powinny korzystać z tego samego źródła danych. Skala czasu na wykresie powinna obejmować jeden miesiąc.

Środowisko pracy

Projekt należy wykonać z wykorzystaniem biblioteki Java Swing. Preferowanym środowiskiem pracy jest IDE Eclipse.

Oddawanie projektów

Projekt należy udostępnić poprzez system kontroli wersji SVN do dnia określonego przez termin realizacji. Nazwa projektu przesłanego na serwer powinna brzmieć **zadanie2**.

Kryteria oceny

Ocenie będzie podlegać :

1. wykonanie GUI zgodnie z projektem graficznym oraz opisem działania – uwaga GUI należy wykonać samemu, bez użycia kreatora typu przesun i puść
2. projekt aplikacji – zastosowanie projektowania obiektowego, wykorzystanie wzorców projektowych (w szczególności MVC)
3. projekt bazy danych – zastosowanie modelu relacyjnego do przechowywania tablic
4. jakość kodu (przejrzystość, formatowanie, czytelne nazewnictwo klas, metod oraz zmiennych)
5. termin oddania – projekty oddane po terminie nie będą przyjmowane

Odnosińki

[1] ORACLE, “Java DB.” [Online]. Available:

<http://www.oracle.com/technetwork/java/javadb/overview/index.html>.

[2] K. Toedter, “JCalendar.” [Online]. Available: <http://www.toedter.com/en/jcalendar/index.html>.