

Programowanie komponentowe

Sprawozdanie z pracy projektowej

Kalendarz

Piotr Wasiak 210346

Filip Florczyk 210175

Informatyka, sem. IV

2017/ 2018

1. Opis projektu

„**Kalendarz**” to program umożliwiający przeglądanie standardowego kalendarza ściennego oraz dodawanie wydarzeń wraz z miejscem, opisem oraz datą. Program obsługuje operowanie na repozytorium reprezentowanym przez plik **.xml** oraz serwer zewnętrzny **MySQL**. Aplikacja wspiera również eksport do formatu **.ics** dzięki któremu można przeglądać wydarzenia poza aplikacją oraz alarm dźwiękowy przypominający o wydarzeniu.

2. Realizacja

2.1 Klasy warstwy danych

Event - jest klasą reprezentującą wydarzenie w warstwie danych. Ta klasa jest wykorzystywana jedynie lokalnie, nie istnieje globalna lista wydarzeń w aplikacji oprócz repozytoriów. Zawiera w sobie informacje takie jak: nazwa wydarzenia, miejsce wydarzenia, opis oraz data tego wydarzenia. Zawiera także tzw. gettery i settery, z których korzysta klasa **DataRepository**.

Constructors:

Constructor and Description

Event()

Konstruktor klasy **Event** tworzący nowe wydarzenie z ustawieniem null

Event(java.lang.String name, java.lang.String description, java.util.Date date, java.lang.String place)

Konstruktor klasy **Event** tworzący nowe wydarzenie z podanych danych

Method Summary:

Modifier and Type	Method and Description
java.util.Date	getDate() Metoda zwracająca date wydarzenia
java.lang.Stri	getDescription()

ng Metoda zwracajaca opis wydarzenia

java.lang.String
ng **getMySQLDate()**

java.lang.String
ng **getName()**
Metoda zwracajaca nazwe
wydarzenia

java.lang.String
ng **getPlace()**
Metoda zwracajaca miejsce
wydarzenia

void **setDate**(java.util.Date newDate)
Metoda ustawiajaca date wydarzenia

void **setDescription**(java.lang.String new
Description)
Metoda ustawiajaca opis wydarzenia

void **setName**(java.lang.String newName)
Metoda ustawiajaca nowa nazwe
wydarzenia

void **setPlace**(java.lang.String newPlace)
Metoda ustawiajaca miejsce
wydarzenia

java.lang.String
ng **toString()**

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

Event

```
public Event(java.lang.String name,  
             java.lang.String description,  
             java.util.Date date,  
             java.lang.String place)
```

Konstruktor klasy Event tworzący nowe wydarzenie z podanych danych

Parameters:

name - nazwa wydarzenia

description - opis wydarzenia

date - data wydarzenia

place - miejsce wydarzenia

Event

public Event()

Konstruktor klasy Event tworzący nowe wydarzenie z ustawieniem null

Method Detail

getName

public java.lang.String getName()

Metoda zwracająca nazwę wydarzenia

Returns:

zwraca nazwę

setName

public void setName(java.lang.String newName)

Metoda ustawiająca nową nazwę wydarzenia

Parameters:

newName - nowa nazwa wydarzenia

getDescription

public java.lang.String getDescription()

Metoda zwracająca opis wydarzenia

Returns:

zwraca opis wydarzenia

setDescription

public void setDescription(java.lang.String newDescription)

Metoda ustawiająca opis wydarzenia

Parameters:

newDescription - opis wydarzenia

getDate

```
public java.util.Date getDate()
```

Metoda zwracajaca date wydarzenia

Returns:

zwraca date wydarzenia

getMySqlDate

```
public java.lang.String getMySqlDate()
```

setDate

```
public void setDate(java.util.Date newDate)
```

Metoda ustawiajaca date wydarzenia

Parameters:

newDate - data wydarzenia

getPlace

```
public java.lang.String getPlace()
```

Metoda zwracajaca miejsce wydarzenia

Returns:

zwraca miejsce wydarzenia

setPlace

```
public void setPlace(java.lang.String newPlace)
```

Metoda ustawiajaca miejsce wydarzenia

Parameters:

newPlace - miejsce wydarzenia

toString

```
public java.lang.String toString()
```

Overrides:

toString in class java.lang.Object

EventList - jest klasą pomocniczą, zawierającą w sobie listę wydarzeń. Jest wykorzystywana głównie przy operacjach związanych z plikiem .xml, takich jak serializacja i deserializacja.

DataRepository – jest to interfejs opisujący działanie repozytorium na obiektach wydarzeń. Definiuje on takie operacje jak: zwracanie wydarzenia, zwracanie listy wydarzeń, usuwanie wydarzenia oraz aktualizacja wydarzenia.

XMLRepository - jest to klasa implementująca interfejs DataRepository, zapewniająca zapis, odczyt, usuwanie oraz aktualizację wydarzeń w pliku *Events.xml*.

MySQLRepository – jest to druga klasa implementująca interfejs DataRepository, zapewniająca zapis, odczyt, usuwanie oraz aktualizację wydarzeń na serwerze zewnętrznym MySQL, zlokalizowanym na platformie db4free.net

2.2 Klasy warstwy logiki

DataService – jest to klasa odpowiedzialna za logikę obsługi wydarzeń, wybranego repozytorium oraz filtrowanie danych.

ICalExporter – jest to klasa odpowiedzialna za obsługę eksportu do formatu *.ics.

2.3 Klasy warstwy prezentacji

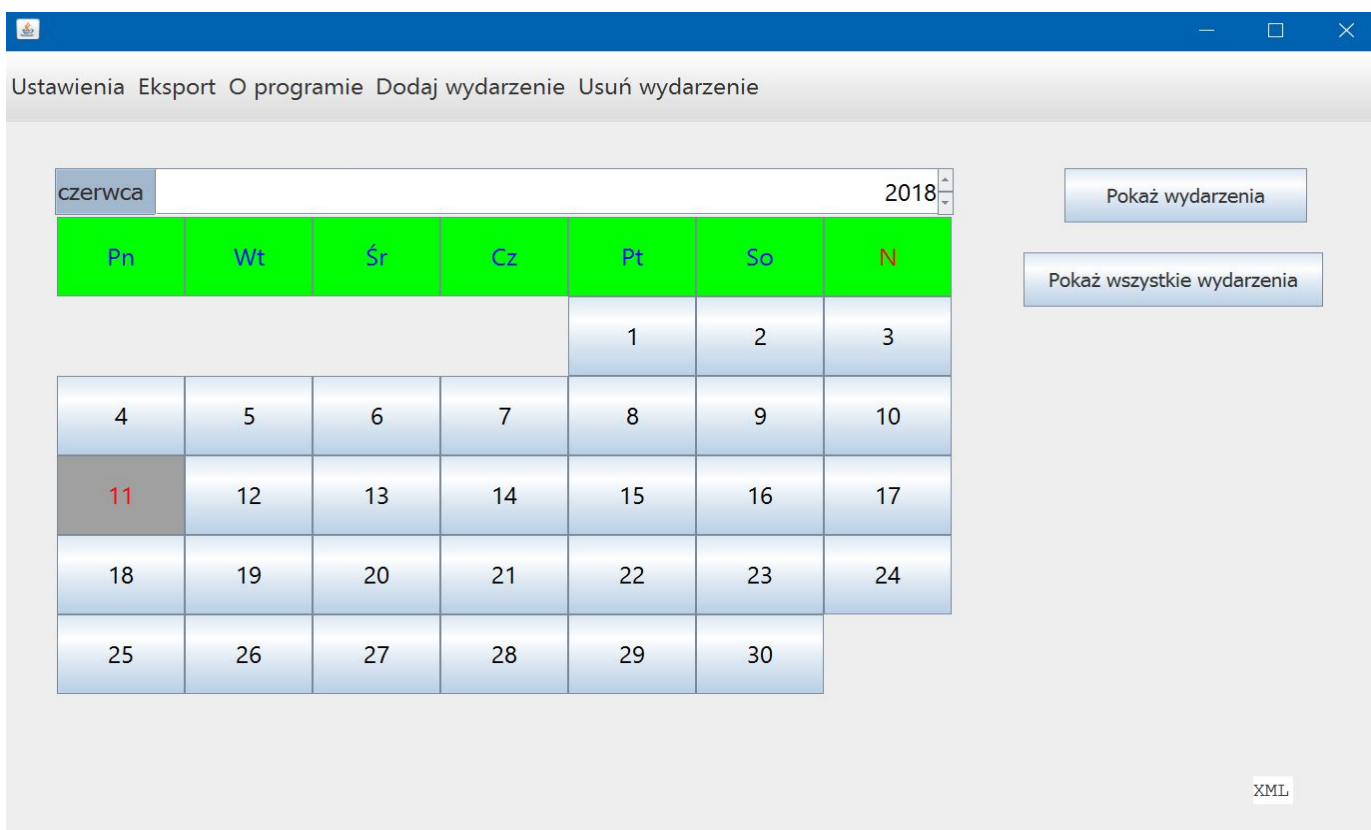
UserInterface – jest klasa reprezentująca główne okno aplikacji, zawierające wygląd kalendarza ściennego, menu główne oraz przyciski wyświetlania wydarzeń.

ButtonListener – jest to klasa odpowiedzialna za reakcje na interakcję z użytkownikiem (wciśnięcie przycisku, wybranie opcji w menu głównym itp.) i obsługę dodatkowych okienek aplikacji (dodawania wydarzeń, usuwania wydarzeń, listy wydarzeń oraz okna „O programie”).

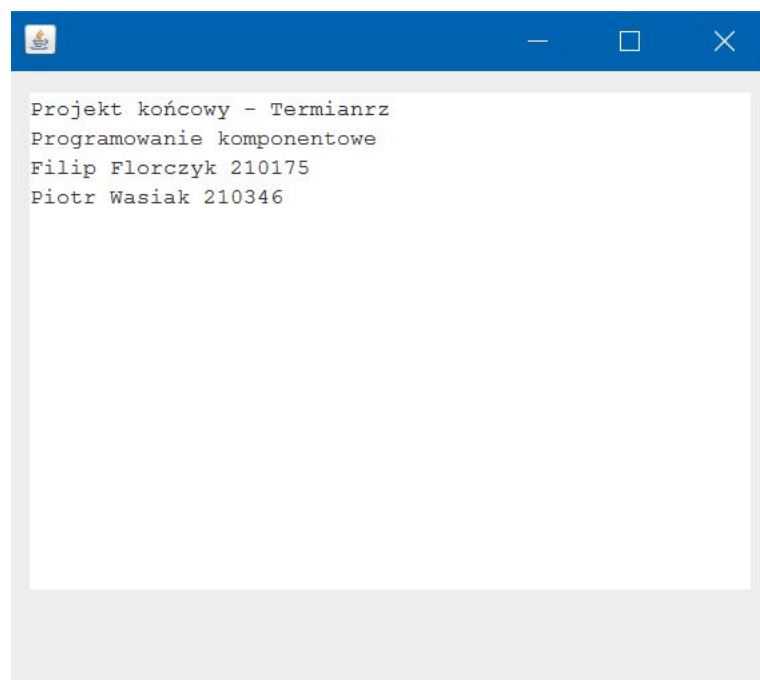
Alarm – jest to klasa realizująca obsługę alarmu dźwiękowego, przypominającego użytkownikowi o najbliższym w czasie wydarzeniu

3. Widoki poszczególnych okien

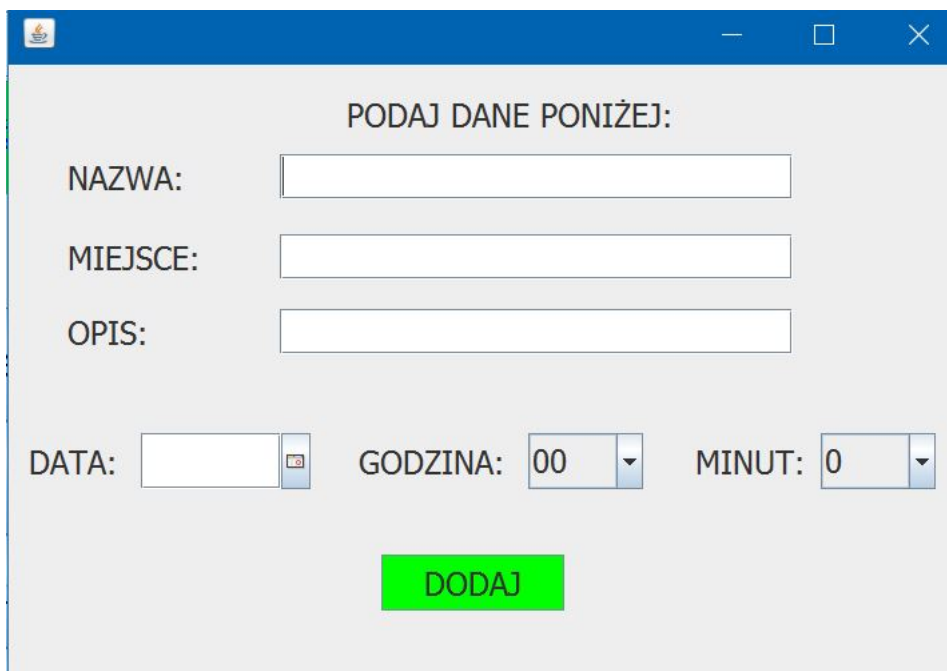
3.1 Główne okno aplikacji



3.2 Okno o programie



3.3 Okno dodawania wydarzeń




A screenshot of a Windows application window. The title bar is blue and contains a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area is light gray and contains the following form elements:


PODAJ DANE PONIŻEJ:


NAZWA:

MIEJSCE:

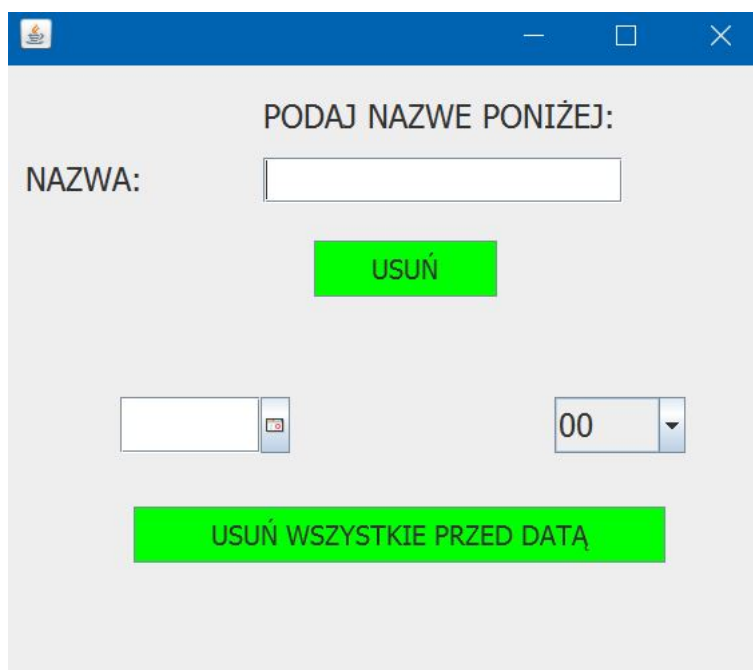
OPIS:

DATA: 

GODZINA: 

MINUT: 

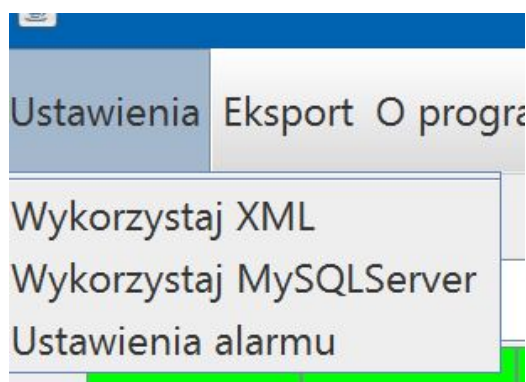
3.4 Okno usuwania wydarzeń



The screenshot shows a window titled 'Okno usuwania wydarzeń' with a blue header bar. The main area is light gray and contains the following elements:

- Text: 'PODAJ NAZWE PONIŻEJ:' (Provide the name below:)
- Text: 'NAZWA:' (Name:)
- Text input field for the name.
- Green button labeled 'USUŃ' (Delete).
- Text input field for a date, followed by a small calendar icon.
- Text input field for a time, showing '00' and a dropdown arrow.
- Large green button labeled 'USUŃ WSZYSTKIE PRZED DATA' (Delete all before date).

3.5 Menu ustawień



3.6 Widok bazy danych

				id	name	description	date	▼ 1	place
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	29	Spotkanie	Spotkanie biznesowe	2018-06-23 10:00:00		Wrocław
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	28	Spotkanie	Spotkanie biznesowe	2018-06-22 08:00:00		Warszawa
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	30	Urodziny	Urodziny Basi	2018-06-16 15:00:00		Zgierz
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	31	Spotkanie2	Spotkanie biznesowe	2018-06-12 14:00:00		Łódź

4. Podsumowanie

Projektowanie kalendarza pozwoliło nam rozwinąć swoje umiejętności programowania w języku Java oraz pewne ogólne umiejętności tworzenia aplikacji zgodnie z architekturą trójwarstwową. Dodatkowo, nauczyliśmy się również korzystać z gotowych komponentów, których implementacja w projekcie zajmuje znacznie mniej czasu niż napisanie całego mechanizmu od początku (np. jcalendar). Poznaliśmy także różne sposoby przechowywania danych, z których nasza aplikacja korzysta (plik *.xml, zewnętrzny serwer z bazą danych MySQL). Nabyliśmy również umiejętność tworzenia dokumentacji przy pomocy narzędzia javadoc.