Sprawozdanie do projektu:

Kalkulator

Autorzy projektu:
Tymoteusz Urbaniak
Kamil Jadczuk vel Bartoszuk
Dariusz Korszun
Maciej Budziński

Informatyka VI semestr

Koszalin 2015

Spis treści

1. **Protokół założycielski**

PROTOKÓŁ ZAŁOŻYCIELSKI

Protokół z zebrania mającego na celu utworzenia grupy projektowej odbytego w dniu 8.03.2015 w Koszalinie

§1. Obecni na zebraniu studenci na ul Śniadeckich 2 Koszalin:

1. Dariusz Korszun
2. Maciej Budziński
3. Kamil Jadczuk
4. Tymoteusz Urbaniak

Po zapoznaniu się z celami, zadaniami oraz wymaganiami przedmiotu: „Projekt grupowy”, podjęli decyzję o utworzeniu grupy projektowej złożonej z ww. studentów.

§2. Tematem projektu jest program „Kalkulator”

§3. Ustalili drogą demokratyczną kierownika grupy projektowej.

- kierownik projektu – Tymoteusz Urbaniak

§3.1 Ustalili role członków grupy projektowej:

- Dariusz Korszun – analityk, tester, projektant;
- Maciej Budziński – programista, tester, analityk;
- Kamil Jadczuk – programista, tester, projektant;

§4. Obowiązki poszczególnych stanowisk:
a. kierownik projektu
- kontrolowanie postępów pracy
- przydzielanie zadań projektowych
- ustalanie harmonogramu zadań

b. analityk
- analiza problemu
- przedstawienie wymagań w specyfikacji

c. projektant
- projektowanie elementów programu za pomocą odpowiednich narzędzi
- dokumentowanie projektu

d. programista

- tworzenie programu na podstawie dokumentacji

e. tester

- testowanie aplikacji
- dokumentowanie testów
- analiza błędów

§5. Ustalili że za pierwsze niewykonane zadanie lub nie wykonanie go w terminie otrzymuje dana osoba upomnienie, a kolejne przewinienie podlega karze dostarczenia pizzy pozostałym członkom grupy projektowej..

§6. Odpowiedzialność za niepowodzenia ponosi cała grupa.

2. Harmonogram projektu

Harmonogram projektu													
poniedziałek		16 Mapa konceptualna	23 - Specyfikacja	30 Identyfikacja	6 - Diagramy klas i	13 - Projektowanie	20 - Kodowanie	27 - Przygotowanie	4 - Ewaluacja	11	18	25	1	8
wtorek		17 Mapa konceptualna	24 - Specyfikacja	31 Diagram przypadk	7 - Diagramy klas i	14 - Projektowanie	21 - Kodowanie	28 - Przygotowanie	5 - Ewaluacja	12	19	26	2	9
środa		18 - Opis funkcjonalny	25 - Specyfikacja	1 - Diagram przypadk	8 - Diagramy klas i	15 - Projektowanie	22 - Kodowanie	29 - Przygotowanie	6 - Jak dobrze pójdzie	13	20	27	3	10
czwartek		19 - Opis funkcjonalny	26 - Specyfikacja	2 - Diagram przypadk	9 - Diagramy klas i	16 - Projektowanie	23 - Kodowanie	30 - Przygotowanie	7	14	21	28	4	11
piątek	III Zjazd - Burza mózgów,	20 - Opis funkcjonalny	IV Zjazd - Ewaluacja i	3 - Diagram przypadk	V Zjazd - Ewaluacja	17 - Projektowanie	VI Zjazd - Testy i poprawki	1 - Przygotowanie	VII Zjazd	VIII Zjazd	22	IX Zjazd	5	X Zjazd
sobota	III Zjazd - Burza mózgów	21 - Opis funkcjonalny	IV Zjazd - Ewaluacja i	4 - Diagram przypadk	V Zjazd - Ewaluacja	18 - Projektowanie	VI Zjazd - Testy i poprawki	2 - Przygotowanie	VII Zjazd	VIII Zjazd	23	IX Zjazd	6	X Zjazd
niedziela	15 Mapa konceptualna	22 - Opis funkcjonalny	IV Zjazd - Identyfikacja	5 - Diagramy klas i	V Zjazd - Ewaluacja	19 - Projektowanie	VI Zjazd - Testy i poprawki	3 - Ewaluacja	VII Zjazd	VIII Zjazd	24	IX Zjazd	7	X Zjazd

3. Cel projektu

Celem projektu jest stworzenie aplikacji "kalkulator" służącej do obliczeń matematycznych, a także stworzenie dokumentacji do tego projektu.

4. Wymagania:

a) funkcjonalne

- Intuicyjny interfejs
- Przejrzystość
- Duża dokładność
- Możliwość zapamiętywania wyniku
- Możliwość wykonywania działań arytmetycznych
- Możliwość wykonywania obliczeń budowlanych:
- Obliczanie ilości farby, gładzi, kafelek, paneli, drewna potrzebnych do remontu
- Kasowanie danych
- Funkcje trygonometryczne
- Duże przyciski
- Przejrzysty wyświetlacz
- Szybkie uruchamianie
- Zmiana funkcji przycisków

b) poza funkcjonalne

- zmiana koloru przycisków
- zmiana czcionek na przyciskach
- zmiana wielkości przycisków
- zmiana układu przycisków
- zmiana koloru czcionki przycisków
- zmiana klawisza uruchomiającego przycisk
- zmiana wyglądu wyświetlacza
- zmiana układu wyświetlacza
- zmiana czcionki wyświetlacza

c) dziedzinowe

- Procesor: zgodny z wymaganiami systemu operacyjnego
- Pamięć RAM: zgodna z wymaganiami systemu operacyjnego
- Twardy dysk: 50MB wolnej przestrzeni dyskowej
- Drukarka: laserowa lub atramentowa
- System operacyjny: Windows XP Service Pack 2, Windows Vista, Windows 7

5. Mapa myśli

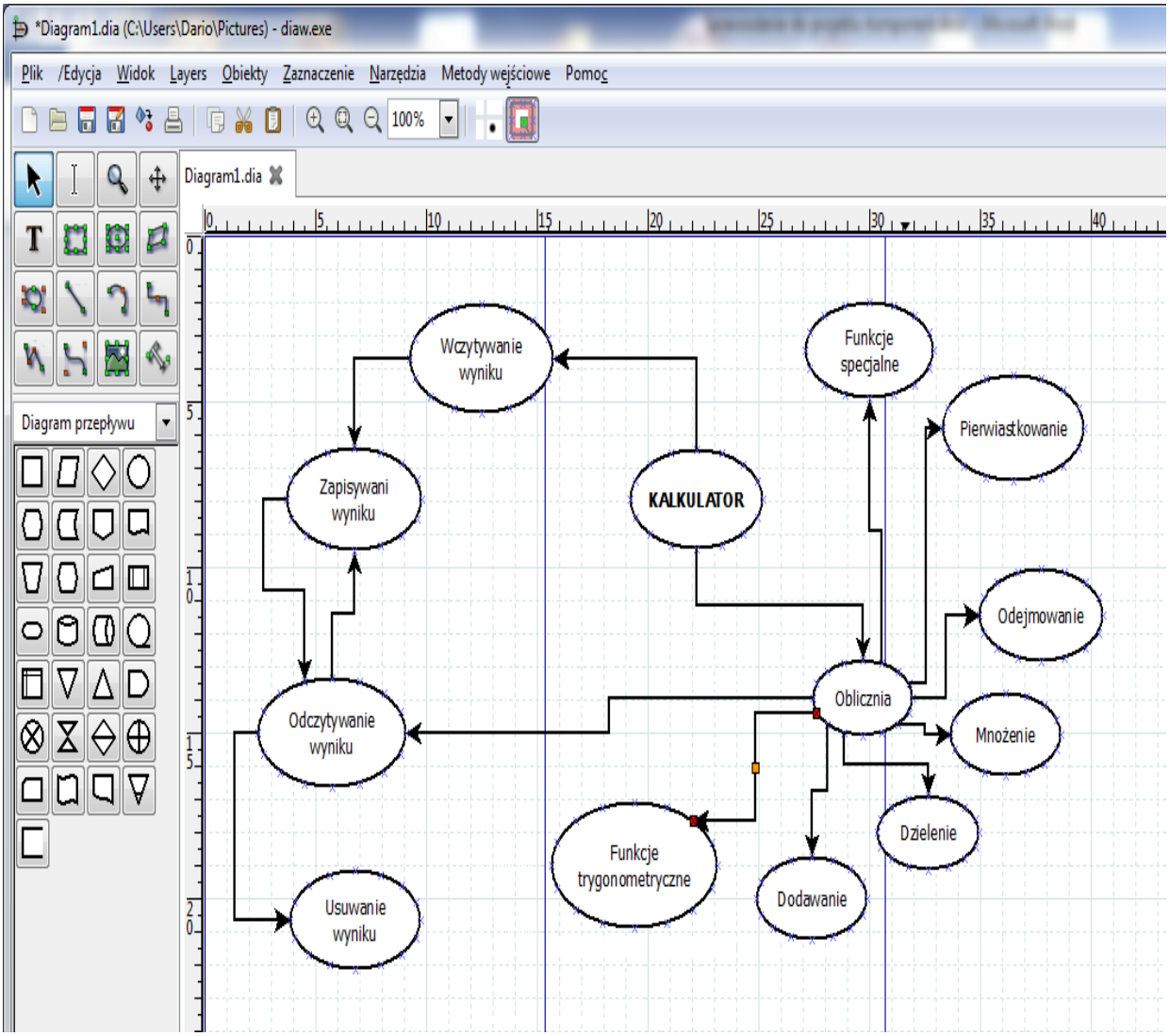
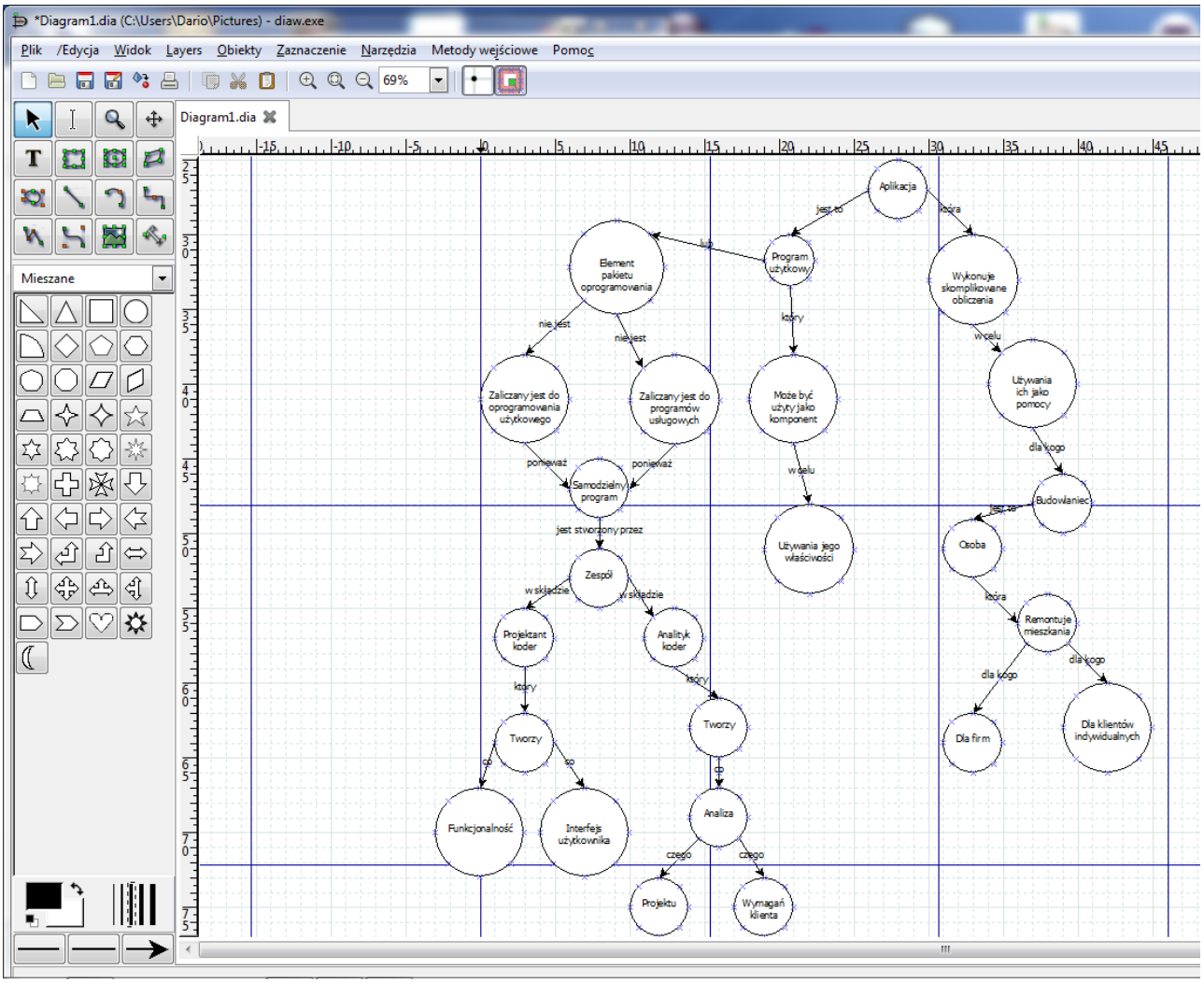
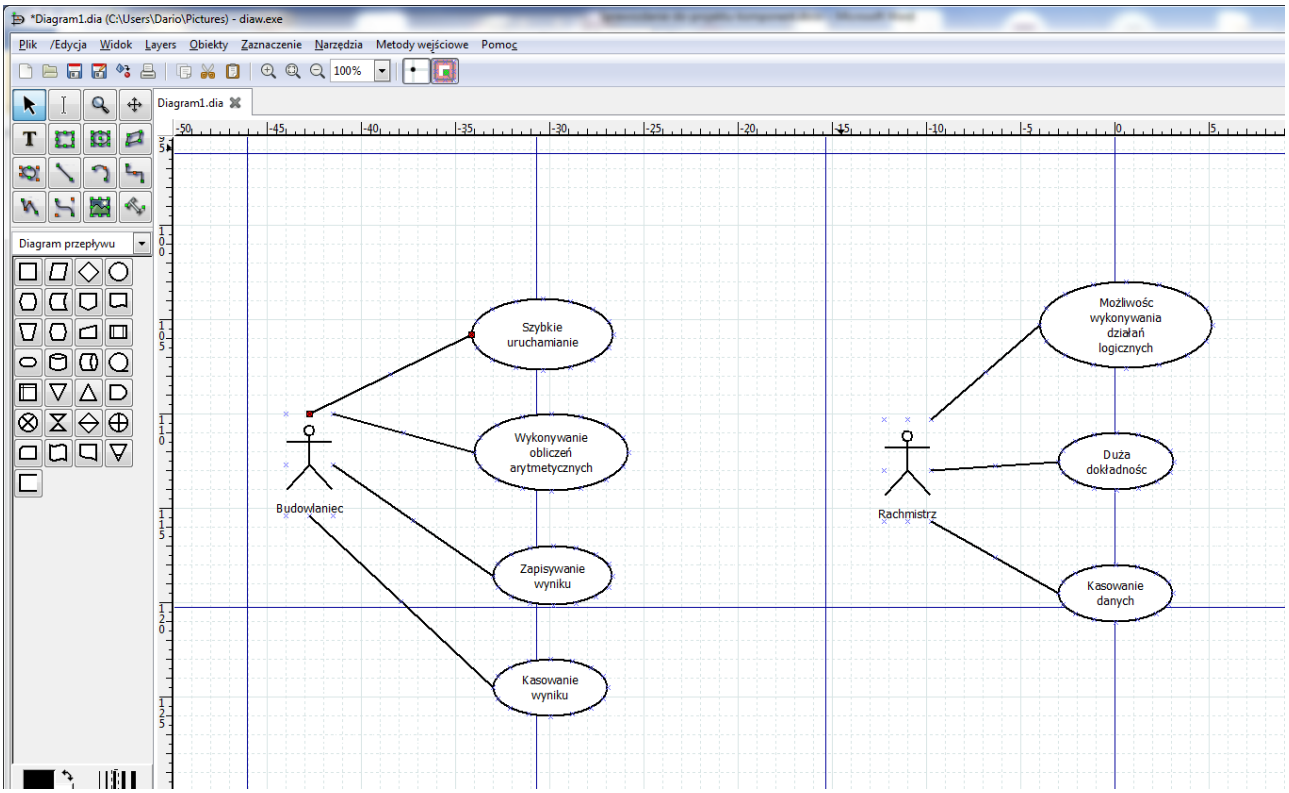


*Diagram1.dia (C:\Users\Dario\Pictures) - diaw.exe
Plik /Edycja Widok Layers Obiekty Zaznaczenie Narzędzia Metody wejściowe Pomoc
100%
Diagram1.dia
Diagram przepływu
Wczytywanie wyniku
Funkcje specjalne
Pierwiastkowanie
Zapisywani wyniku
KALKULATOR
Odejmowanie
Oblicznia
Odczytywanie wyniku
Mnożenie
Dzielenie
Funkcje trygonometryczne
Dodawanie
Usuwanie wyniku

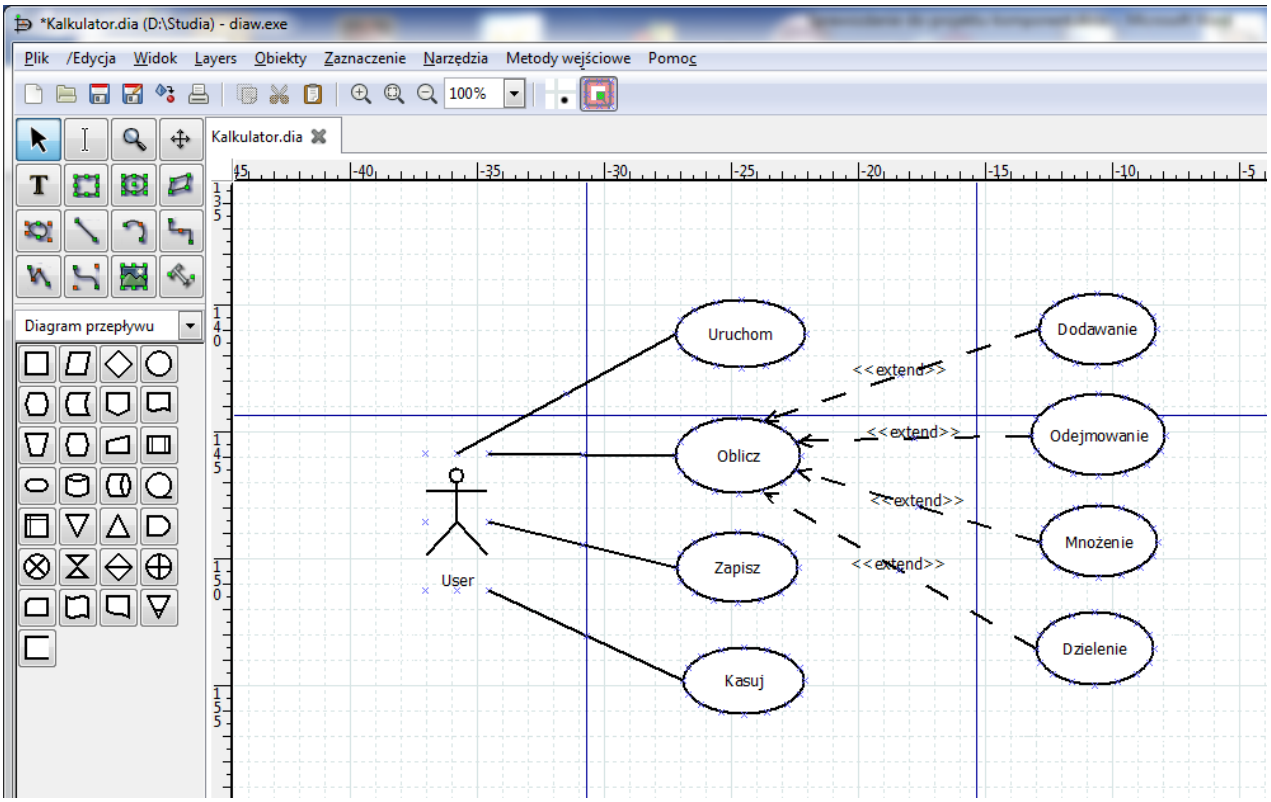
6. Mapa konceptualna



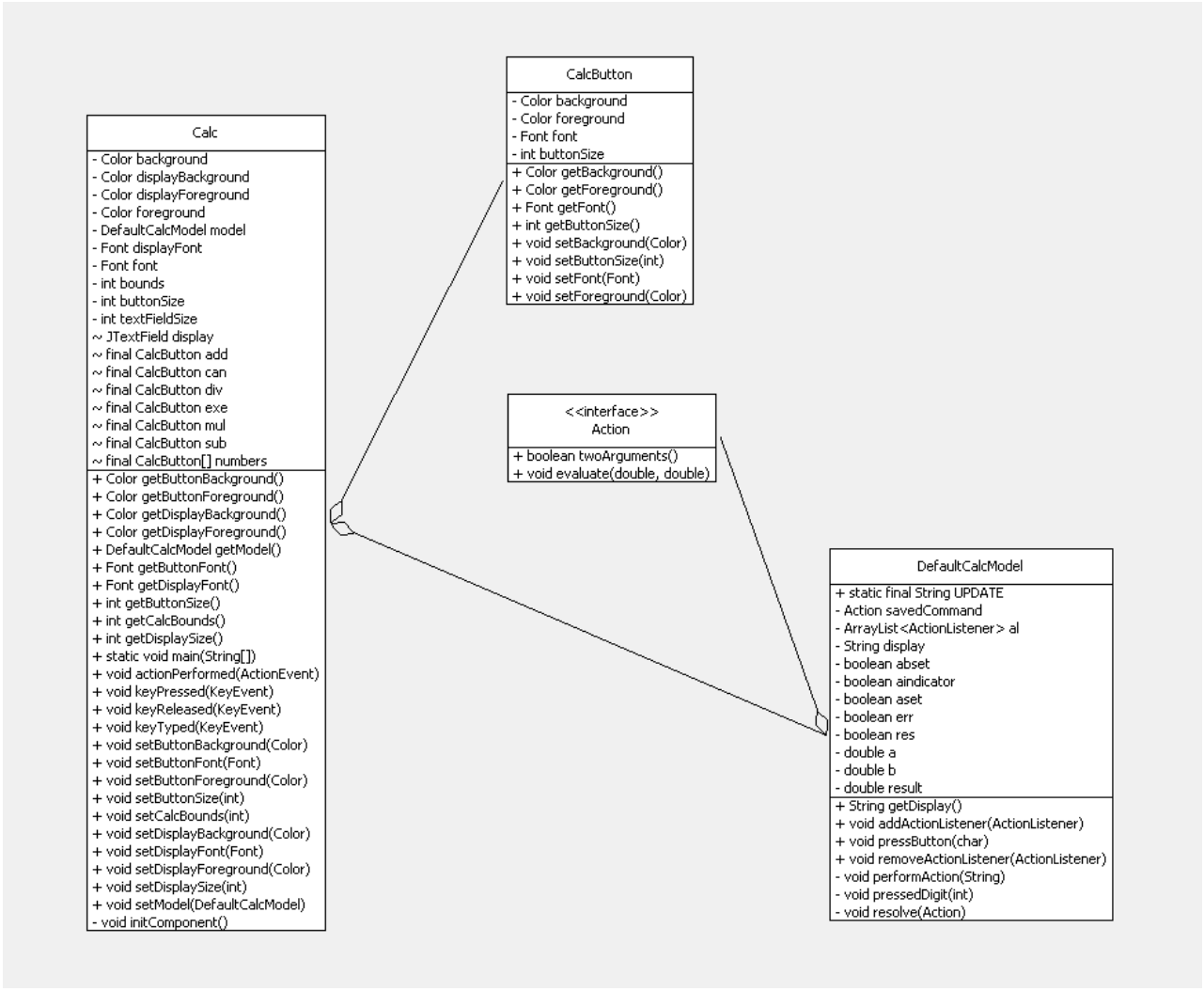
7. Identyfikacja aktorów



8. Diagram przypadków użycia

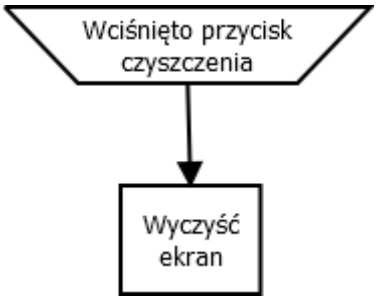


9. Diagram klas

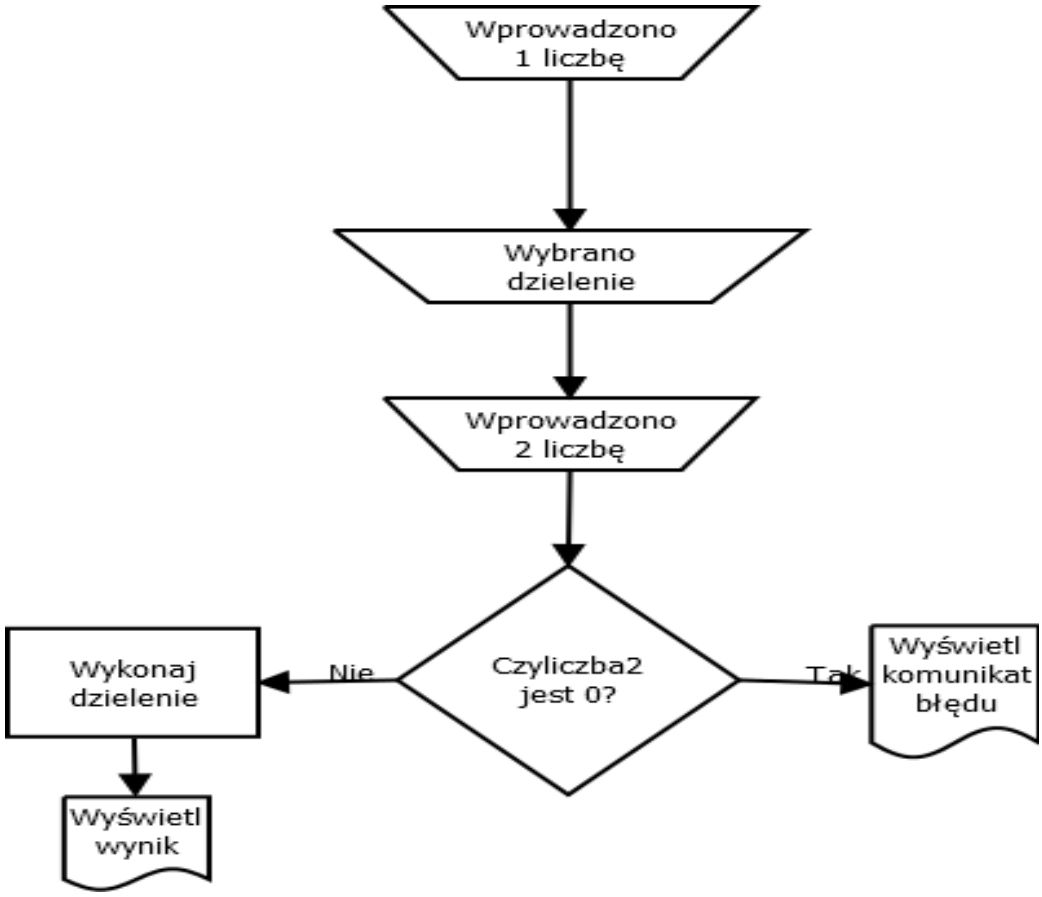


10. Diagramy proceduralne

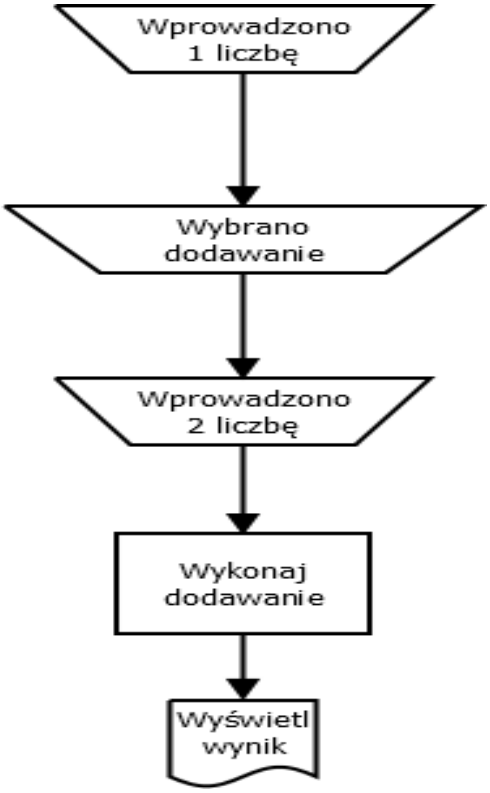
a) Czyszczenie



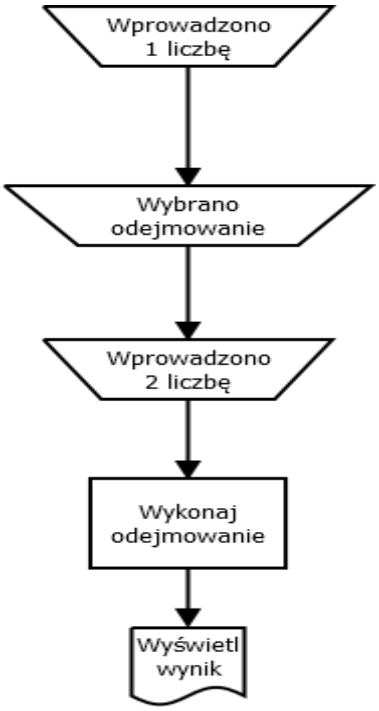
b) Dzielenie



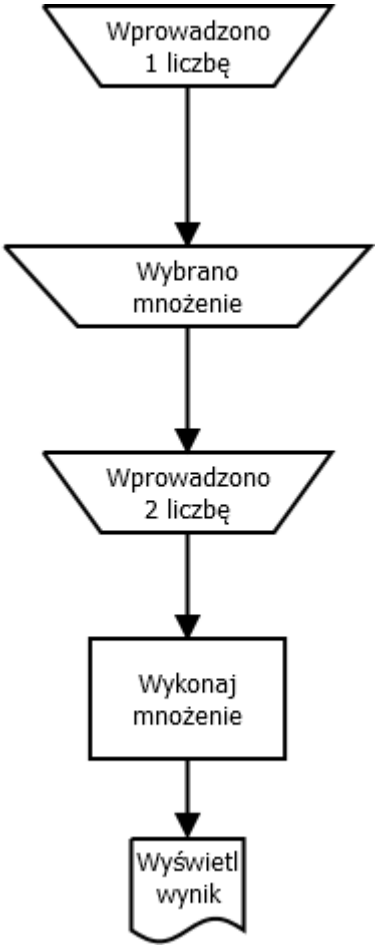
c) Dodawanie



d) Odejmowanie



e) Mnożenie



11. Specyfikacja przypadków użycia

Funkcja	Dodawanie
Nazwa	Add
Klasa	DefaultCalcModel.java
Opis funkcji	Funkcja dodaje liczby wprowadzone przez użytkownika i wyświetla sumę w oknie wyniku.
Autorzy	KJ, TU
Dane wejściowe	Double
Dane wyjściowe	Double

Funkcja	Odejmowanie
Nazwa	Sub
Klasa	DefaultCalcModel.java
Opis funkcji	Funkcja odejmuje liczby wprowadzone przez użytkownika i wyświetla różnicę w oknie wyniku.
Autorzy	KJ, TU
Dane wejściowe	Double
Dane wyjściowe	Double

Funkcja	Dzielenie
Nazwa	Div
Klasa	DefaultCalcModel.java
Opis funkcji	Funkcja dzieli liczby wprowadzone przez użytkownika i wyświetla iloraz w oknie wyniku.
Autorzy	KJ, TU
Dane wejściowe	Double
Dane wyjściowe	Double

Funkcja	Mnożenie
Nazwa	Mul
Klasa	DefaultCalcModel.java
Opis funkcji	Funkcja mnoży liczby wprowadzone przez użytkownika i wyświetla iloczyn w oknie wyniku.
Autorzy	KJ, TU
Dane wejściowe	Double
Dane wyjściowe	Double

12. Zarys interfejsu użytkownika GUI

13. Fragmenty kodu

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;


public class DefaultCalcModel {
	private double a,b,result;
	public static final String UPDATE = "UPDATE";
	private boolean abset=false;
	private boolean aindicator=false;
	private boolean aset =false;
	private boolean res = false;
	private boolean err = false;
	private String display="";
	private Action savedCommand;
	private ArrayList<ActionListener>al = new ArrayList<ActionListener> ();
	public  void addActionListener (ActionListener alist) {
		al.add(alist);
	}
	private void performAction (String action) {
```

```
                for (int i=0; i<al.size(); i++)
                        al.get(i).actionPerformed(new ActionEvent (this,
ActionEvent.ACTION_PERFORMED, action));
        }
        public String getDisplay() {
                // TODO Auto-generated method stub
                return display;
        }

        public void pressButton(char command) {
                if (command >= '0' && command <= '9') pressedDigit(command - '0');
                if (command == '+') resolve (new Action(){

                        @Override
                        public void evaluate(double a, double b) {
                                result = a+b;
                                System.out.println(result);

                        }

                        @Override
                        public boolean twoArguments() {

                                return true;
                        }

                });
                if (command == '-') resolve (new Action(){

                        @Override
                        public void evaluate(double a, double b) {
                                result = a-b;

                        }

                        @Override
                        public boolean twoArguments() {
                                // TODO Auto-generated method stub
                                return true;
                        }

                });
                if (command == '/') resolve (new Action(){

                        @Override
                        public void evaluate(double a, double b) {
                                if (b==0) {
                                        display = "ERR DIV BY 0";
                                        err=true;
                                }
```

```
                else result = a/b;

            }

            @Override
            public boolean twoArguments() {
                // TODO Auto-generated method stub
                return true;
            }

        });
        if (command == '*') resolve (new Action(){

            @Override
            public void evaluate(double a, double b) {
                result = a*b;
            }

            @Override
            public boolean twoArguments() {
                // TODO Auto-generated method stub
                return true;
            }

        });
        if (command == '=') resolve (new Action(){

            @Override
            public void evaluate(double a, double b) {
                if (savedCommand!=null)savedCommand.evaluate(a, b);
                savedCommand = null;
                res = true;
            }

            @Override
            public boolean twoArguments() {
                // TODO Auto-generated method stub
                return false;
            }
        });
        if (command == 'C') resolve (new Action(){

            @Override
            public void evaluate(double a, double b) {
                if (!display.equals("")) {
                    display="";
                }
                else if (aset) {
                    display = ""+a;
                    aset = false;
```

```
                        }
                    else {
                        display = "";
                    }
                }

                @Override
                public boolean twoArguments() {
                    // TODO Auto-generated method stub
                    return false;
                }

            });


    }

    private void resolve(Action action) {
        if (display.equals("")) return;
        if (aset) {

            if (!action.twoArguments()) {
                b = Double.parseDouble(display);
                aindicator = true;
                action.evaluate(a, b);
                if (!err) display = ""+result;
                else err=false;
                abset=true;
                aset=false;
            }

        }
        if (!aset) {
            if (!abset) {
                a = Double.parseDouble(display);
                aindicator = true;
                aset = true;
                if (action.twoArguments()) savedCommand = action;
                else action.evaluate(a, b);
            }
            else abset = false;
        }

        performAction(UPDATE);
        System.out.println ("display: |" + display + "| a: " + a + " b: " + b +
                " ASET: " + aset + " savedAction: " +savedCommand);
    }
```

```
        private void pressedDigit(int i) {
                if (res) {
                        display = "";
                        res = false;
                }
                if (aindicator) {
                        display = "";
                        aindicator = false;
                }
                if (display.equals("0.0")) display = "";
                display+=i;
                System.out.println ("display: |" + display + "| a: " + a + " b: " + b +
                                " ASET: " + aset + " savedAction: " +savedCommand);
                performAction(UPDATE);
        }
        public void removeActionListener(ActionListener a) {
                al.remove(a);
        }

}
```

14. Instrukcja obsługi

Uwagi dotyczące formy instrukcji, informacje o zauważonych błędach proszę kierować na adres 1232kalk@gmail.com

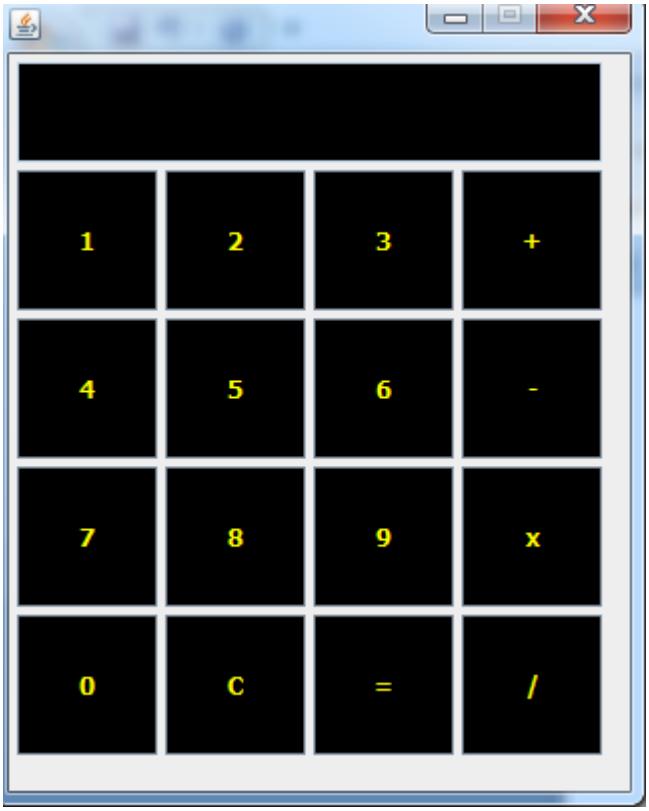
Wersja instrukcji:	1.0 z dn. 13-11-2015
Wersja oprogramowania:	1.0
Instrukcje opracowali:	MB

Program został wykonany w oparciu o technologię Java NetBeans IDE 8.1. Ze względu na dużą ilość funkcjonalności jest dobrym narzędziem do użytku domowego. Jest przeznaczony do pracy z systemem Microsoft Windows XP i wyższe z pakietem NETFramework.

Dane techniczne

Wyświetlacz	22 cyfry
Technologia	Java NetBeans IDE 8.1
Wymagania	Miejsce na dysku twardym 50MB
	Procesor: zgodny z wymaganiami systemu operacyjnego
	Pamięć RAM: zgodna z wymaganiami systemu operacyjnego

Budowa Kalkulatora



Klawiatura

Projekt zakłada rozmieszczenie klawiszy w schemacie klawiatury numerycznej. Dzięki takiemu ustawieniu kalkulator staje się bardziej intuicyjny.

Wyświetlacz

Umożliwia wyświetlanie liczb dodatnich i ujemnych. Uzyskamy również informacje o błędnej definicji danych.

Operacje arytmetyczne

Dodawanie

Dodawanie liczb kalkulator realizuje w następny sposób:
1. wprowadzamy pierwszą liczbę za pomocą myszki,
2. wybieramy operację "+",
3. wprowadzamy drugą liczbę,
4. wciskamy klawisz "="

Odejmowanie

Odejmowanie liczb kalkulator realizuje w następny sposób:
1. wprowadzamy pierwszą liczbę za pomocą myszki,
2. wybieramy operację "-",
3. wprowadzamy drugą liczbę,
4. wciskamy klawisz "="

Mnożenie

Mnożenie liczb kalkulator realizuje w następny sposób:
1. wprowadzamy pierwszą liczbę za pomocą myszki,
2. wybieramy operację "*",
3. wprowadzamy drugą liczbę,
4. wciskamy klawisz "="

Dzielenie

Dzielenie liczb kalkulator realizuje w następny sposób:
1. wprowadzamy pierwszą liczbę za pomocą myszki,
2. wybieramy operację "/",
3. wprowadzamy drugą liczbę,
4. wciskamy klawisz "="

Operacje trygonometryczne

sinus*
cosinus*
tangens*
cotangens*

Inne

Zapamiętanie wyniku*

Wygląd

Użyte zostały kolory o dużym kontraście względem siebie. Tekst jest w kolorze żółtym, tło przycisków i wyświetlacza ma kolor czarny.

Błędy sygnalizowane na wyświetlaczu

Błąd dzielenia liczby przez 0 oraz 0/0 sygnalizuje wyświetlacz napisem "ERR DIV BY 0".

*w trakcie realizacji

15. Podsumowanie

Aplikacja w obecnej wersji została wyposażona tylko w podstawowe elementy kalkulatora arytmetycznego jakim są dodawanie, odejmowanie, mnożenie i dzielenie. Wygląd jest dość ubogi w szatę graficzną co w przyszłości wymagałoby poprawienia, natomiast dzięki temu zachowana została prostota i ergonomia.
W przyszłości aplikację można rozbudować o dodatkowe elementy takie jak:
- funkcje trygonometryczne
- wyświetlanie wykresów funkcji
- zmiana kolorów
- zmiana funkcji przycisków