**Temat projektu: Trójkąt czy nie trójkąt**

Projekt wykonał:

Aleksander Micyk

Data: 15.11.2020r.

1. **Cel pracy.**

Pierwszym celem pracy jest stworzenie prostego programu przyjmującego 3 dane wejściowe, które są liczbami naturalnymi, a następnie sprawdzenie czy z danych liczb jesteśmy w stanie stworzyć trójkąt.

Drugim celem jest napisanie aplikacji testującej sprawdzającej naszą aplikację , pod względem założeń i cech programowania komponentowego.

1. **Zakres pracy.**
2. **Komponent:**

* Stworzenie krótkiego opisu aplikacji,
* Stworzenie 3 pól do których użytkownik będzie mógł wpisać wartości,
* Stworzenie przycisku „oblicz” służącego do obliczania(zaimplementowanie logiki aplikacji)
* Stworzenie przycisku „losowanie” służącego do losowania liczb naturalnych,
* Stworzenie przycisku „wyczyść” służącego do czyszczenia pól,
* Stworzenie pola do wypisywania informacji o wyniku,
* Ustawienie odpowiednich informacji i kolorów do poszczególnych operacji.

1. **Aplikacja testująca:**

* Stworzenie 3 przycisków do zmiany zakresu liczb losujących (1-10, 10-100, 100-1000),
* Stworzenie 2 pól kombinacji do zmiany kolorów dla wyniku pozytywnego i negatywnego.

1. **Podstawy teoretyczne.**

Do stworzenia algorytmu sprawdzającego czy z podanych 3 liczb możemy stworzyć trójkąt niezbędna jest zależność trójkąta. Długość najdłuższego odcinka musi być mniejsza niż suma pozostałych dwóch.

Weźmy trzy odcinki długości:

* 3, 5 i 8 – nie uda się z nich zbudować trójkąta, gdyż najdłuższy z tych odcinków, czyli odcinek o długości 8, **nie jest krótszy** od sumy długości dwóch pozostałych: 3 + 5 = 8
* 4, 9 i 6 – uda się z nich zbudować trójkąt, gdyż najdłuższy z tych odcinków, czyli odcinek o długości 9, **jest krótszy** od sumy długości dwóch pozostałych: 4 + 6 = 10
* 11, 1 i 3 – nie uda się z nich zbudować trójkąta, gdyż najdłuższy z tych odcinków, czyli odcinek o długości 11, **nie jest krótszy** od sumy długości dwóch pozostałych: 1 + 3 = 4.

1. **Zmienne.**
2. **Komponent:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lp. | Nazwa | Przeznaczenie | Typ | Przykład |
| 1 | jLabelInfo | Informacje dla użytkownika | JLabel | "Wprowadź 3 długości odcinków lub je wylosuj następnie wciśnij oblicz " |
| 2 | jTextField1 | Pole wymaga od użytkownika wprowadzenia długości pierwszego odcinka | JTextField | 7 |
| 3 | jTextField2 | Pole wymaga od użytkownika wprowadzenia długości drugiego odcinka | JTextField | 8 |
| 4 | jTextField3 | Pole wymaga od użytkownika wprowadzenia długości trzeciego odcinka | JTextField | 5 |
| 5 | jTextField4 | Podaje informację o możliwości stworzenie lub braku możliwości stworzenia trójkąta | JTextField | „ Udało się stworzyć trójkąt” , „ Nie udało się stworzyć trójkąta” |
| 6 | jButtonCalculate | Przycisk to obliczenia. Na podstawie wartości zawartych w jTextField1, 2, 3. | JButton | „oblicz” |
| 7 | jButtonRandom | Przycisk do wypełnienia JTextField 1, 2 , 3 losowymi wartościami z przedziału 1 – 10. | JButton | 5, 4, 5 |
| 8 | jButtonClean | Przycisk do czyszczenia zawartości JTextField | JButton | „wyczyść” |
| 9 | randomValueMin | Ustawia wartość minimalnej liczby zakresu liczb losowych | int | 1 |
| 10 | randomValueMax | Ustawia wartość maksymalnej liczby zakresu liczb losowych | int | 10 |
| 11 | trueColor | Ustawia kolor dla wariantu, w którym udaje się stworzyć trójkąt | Color | Color.green |
| 12 | falseColor | Ustawia kolor dla wariantu, w którym nie udaje się stworzyć trójkąta | Color | Color.red |

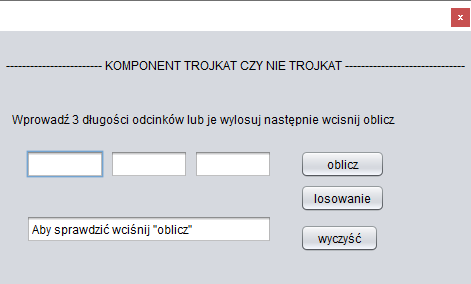
1. Aplikacja testująca

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lp. | Nazwa | Przeznaczenie | Typ | Przykład |
| 1 | jButton1 | Przycisk zmiany przedziału losowania liczb | JButtoon | 1 -10 |
| 2 | jButton2 | Przycisk zmiany przedziału losowania liczb | JButtoon | 10 - 100 |
| 3 | jButton3 | Przycisk zmiany przedziału losowania liczb | JButtoon | 100 - 1000 |
| 4 | jComboBox2 | Pole kombinacji zmienia koloru wyniku pozytywnego | JTextField | „green” |
| 5 | jComboBox3 | Pole kombinacji zmienia koloru wyniku negatywnego | JTextField | „red” |
| 6 | jLabel1 | Informuje użytkownika o możliwościach przycisku | JLabel | „Przedział losowania liczb” |
| 7 | jLabel2 | Informuje użytkownika o możliwościach pola kombinacji | JLabel | „Kolor dla wyniku pozytywnego” |
| 8 | jLabel3 | Informuje użytkownika o możliwościach pola kombinacji | JLabel | „Kolor dla wyniku negatywnego” |
| 9 | trojkatKomponent1 | Komponent trójkąt czy nie trójkąt | TrojkatKomponent | Komponent |

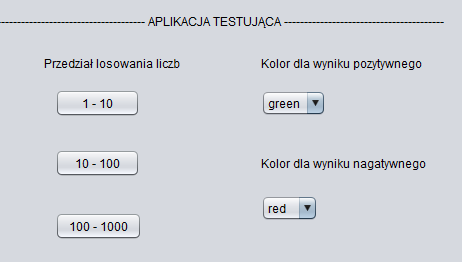
1. **Opis metod.**

* getRandomValueMin - metoda get-er zwraca wartość dolnej granicy liczb pseudolosowych,
* setRandomValueMin - metoda seter umożliwia wprowadzenie wartości dolnej granicy liczb pseudolosowych,
* getRandomValueMax - metoda get-er zwraca wartość górną granicy liczb pseudolosowych,
* setRandomValueMax - metoda set-er umożliwia wprowadzenie wartości górnej granicy liczb pseudolosowych,
* getTrueColor - metoda get-er zwraca kolor dla pozytywnego rezultatu - udało się stworzyć,
* setTrueColor - metoda set-er umożliwia ustawienie odpowiedniego koloru dla pozytywnego rezultatu,
* getFalseColor - metoda get-er zwraca kolor dla negatywnego rezultatu - nie udało się stworzyć,
* setFalseColor - metoda set-er umożliwia ustawienie odpowiedniego koloru dla negatywnego rezultatu,
* checkScore – w metodzie zawarta jest logika programu i zasady związane z budowaniem trójkąta,
* randomLogic- metoda odpowiedzialna za wygenerowanie liczb pseudolosowych z przedziału liczbowego.

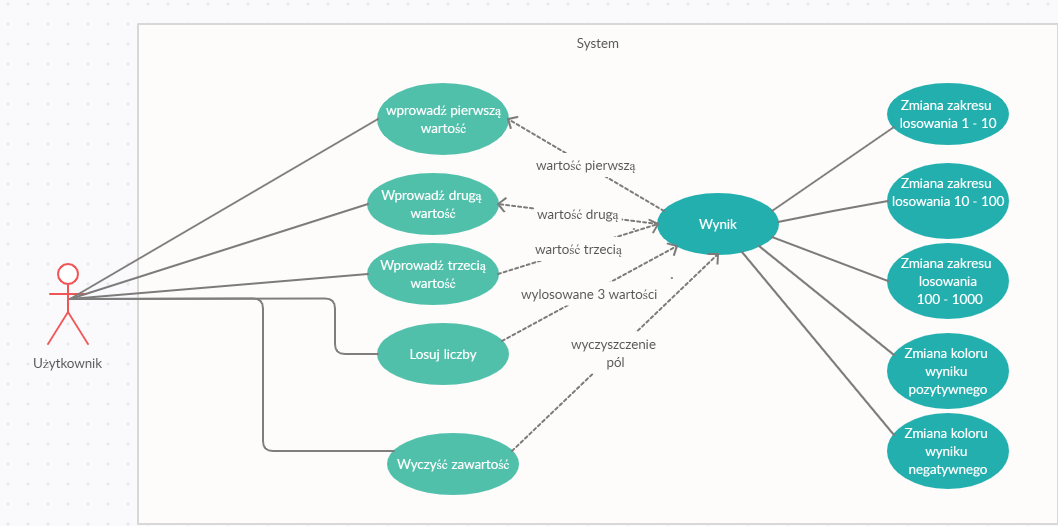
1. **Prototyp Gui komponentu i aplikacji testującej.**
2. **Komponent**



1. **Aplikacja testująca**

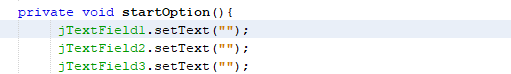


1. **Schemat UML i use case’y.**

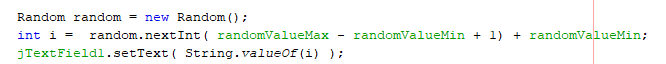




1. **Przypadki użycia komponentu.**
2. Wprowadzenie długości odcinków



1. Generowanie liczby losowej



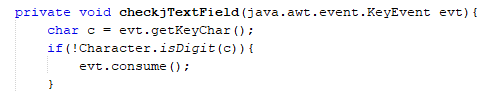
1. Zmiana koloru tła



1. Wyświetlenie wyniku

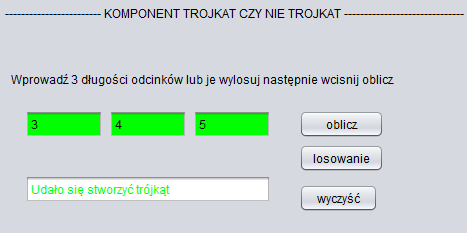


1. Brak możliwości wprowadzenia innych znaków niż 1-9



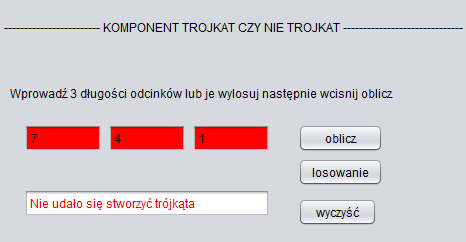
1. **Przypadki testowe.**
2. **ID. 1**

|  |  |
| --- | --- |
| Tytuł | Testowanie przycisku oblicz aplikacji trójkąt czy nie trójkąt |
| Środowisko | Komponent Trójkąt czy nie trójkąt |
| Warunki wstępne | Użytkownik uruchomił aplikacje trójkąt czy nie trójkąt |
| Kroki | * Wprowadzenie pierwszej wartości (3) * Wprowadzenie drugiej wartości (4) * Wprowadzenie trzeciej wartości (5) * Wciśniecie przycisku oblicz |
| Oczekiwany wynik | Wypisana pozytywnej informacji o możliwość stworzenia trójkąta |
| Warunki końcowe | Wypisanie informacji „Udało się stworzyć trójkąt” |
| Wartości wejściowe | Dane liczbowe: liczby naturalne 3, 4 i 5 |
| Kryteria zaliczeniowe | Podświetlenie pól tekstowych na zielono i wyświetlenie napisu „ Udało się stworzyć trójkąt„ Wyświetlone do 5 sekund. |



1. **ID. 2**

|  |  |
| --- | --- |
| Tytuł | Testowanie przycisku losuj aplikacji trójkąt czy nie trójkąt |
| Środowisko | Komponent Trójkąt czy nie trójkąt |
| Warunki wstępne | Użytkownik uruchomił aplikacje trójkąt czy nie trójkąt |
| Kroki | * Wciśniecie przycisku losuj * Wciśniecie przycisku oblicz |
| Oczekiwany wynik | Wypisana pozytywnej lub negatywna informacji o możliwość stworzenia trójkąta |
| Warunki końcowe | Wypisanie informacji „Udało się stworzyć trójkąt”, „Nie udało się stworzyć” |
| Wartości wejściowe | Dane liczbowe: liczby naturalne w tym przypadku losowania 7, 4 , 1 |
| Kryteria zaliczeniowe | Podświetlenie pól tekstowych na czerwono i wyświetlenie napisu „ Nie udało się stworzyć trójkąta„ Wyświetlone do 5 sekund. |

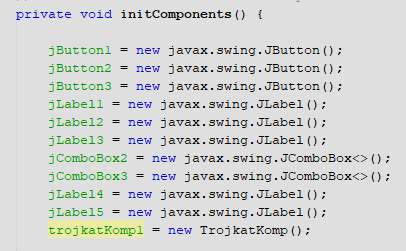


1. **ID. 3**

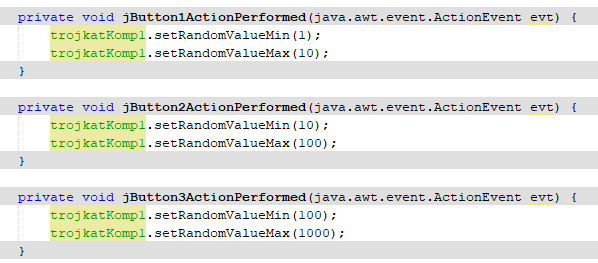
|  |  |
| --- | --- |
| Tytuł | Testowanie przycisku wyczyść aplikacji trójkąt czy nie trójkąt |
| Środowisko | Komponent Trójkąt czy nie trójkąt |
| Warunki wstępne | Użytkownik uruchomił aplikacje trójkąt czy nie trójkąt |
| Kroki | * Wciśniecie przycisku losuj * Wciśniecie przycisku oblicz * Wciśnij przycisk wyczyść |
| Oczekiwany wynik | Wypisana informacja „Wprowadź lub losuj wartości” |
| Warunki końcowe | Wypisanie informacji „Wprowadź lub losuj wartości”, zmiana koloru tła w polach tekstowych na białe |
| Wartości wejściowe | Dane liczbowe: liczby naturalne w tym przypadku losowania 7, 4 , 1 |
| Kryteria zaliczeniowe | Wyczyszczenie pól tekstowych z liczb, zmiana koloru tła pól tekstowych z czerwonego na biały, zmiana pola informacji na „Wprowadź lub losuj wartości”. Zrealizowane do 5 sekund. |



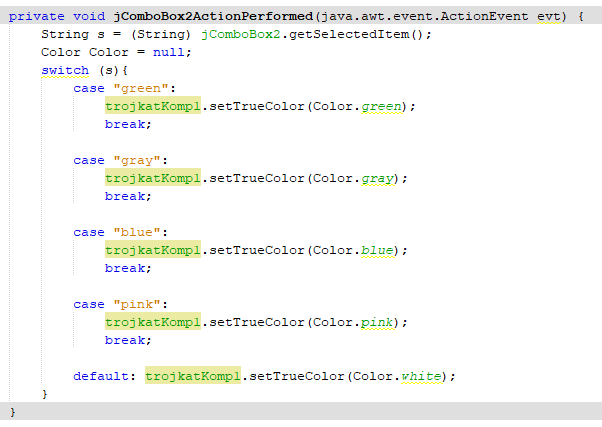
1. **Wywołanie obiektu komponentu w klasie testującej.**



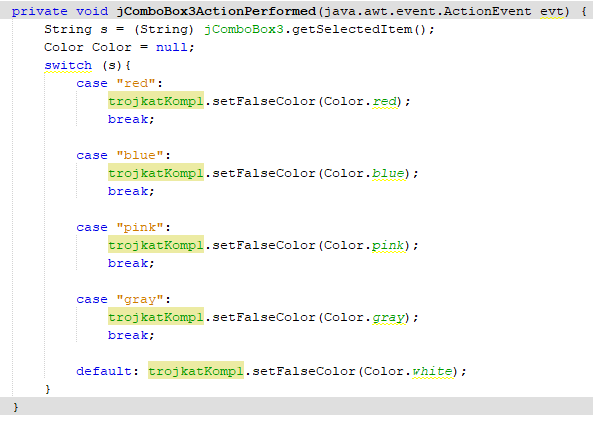
1. **Akcje Listener’y i evt’y w aplikacji testującej.**
2. **Akcje dzięki, którym jesteśmy w stanie zmienić zakres liczb pseudolosowych**



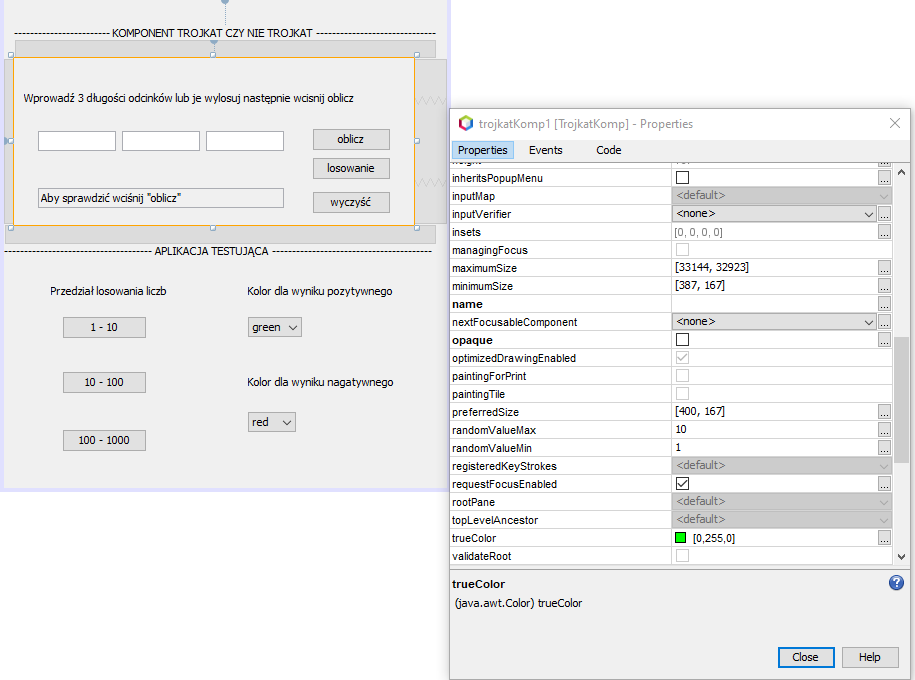
1. **Akcje dzięki, którym możemy zmienić kolor pól tekstowych dla wariantu pozytywnego**

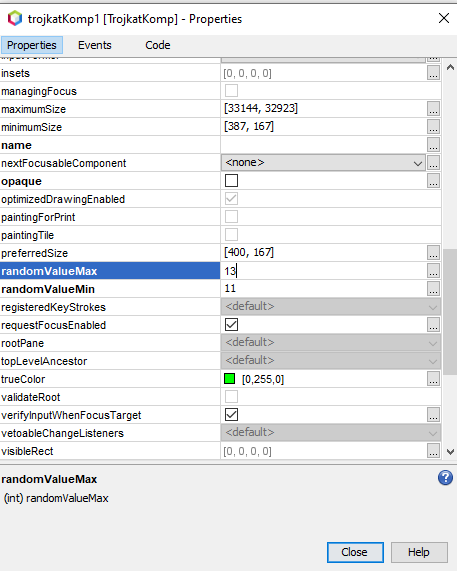


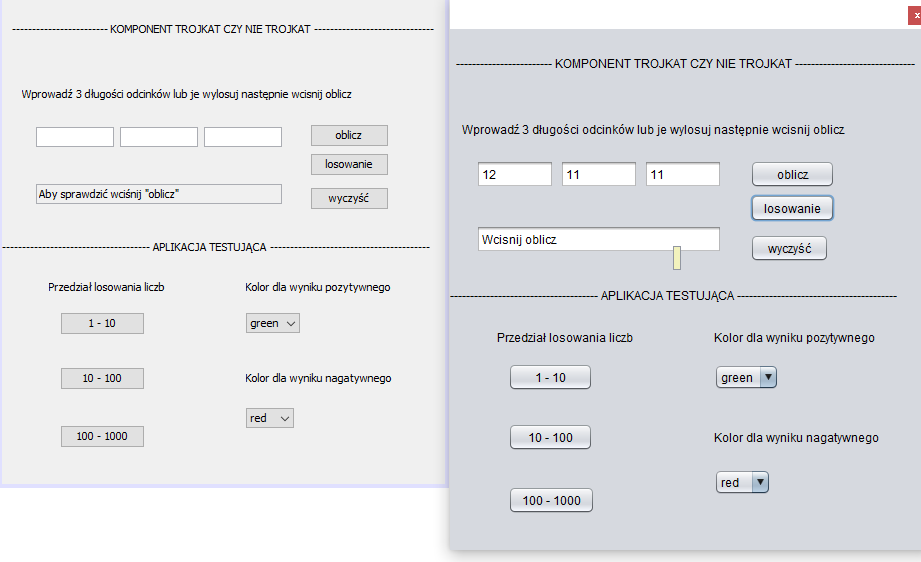
1. **Akcje dzięki, którym możemy zmienić kolor pól tekstowych dla wariantu negatywnego**

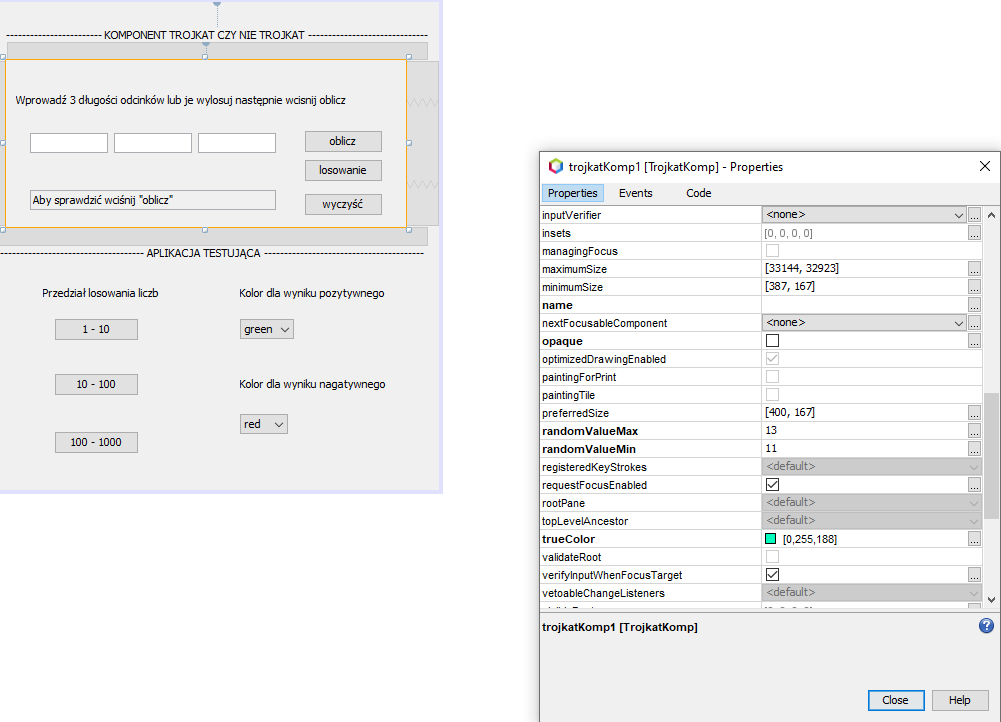


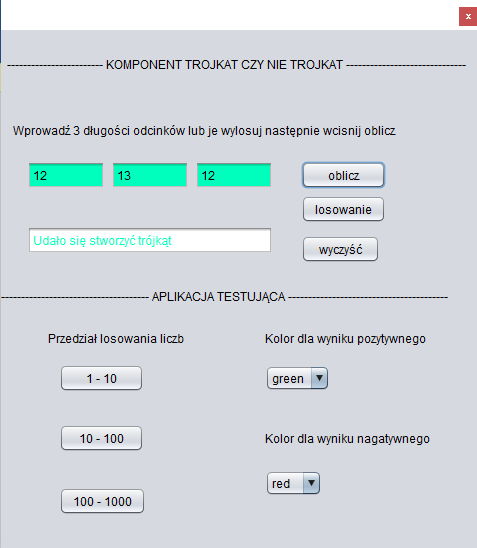
1. **Panel properties komponentu.**











1. **Wnioski.**

Komponent i aplikacja testująca została zrealizowana zgodnie z założeniami.