

# Dia – rysowanie diagramów

Bogdan Kreczmer

bogdan.kreczmer@pwr.edu.pl

Katedra Cybernetyki i Robotyki  
Wydziału Elektroniki  
Politechnika Wrocławска

*Kurs: Programowanie obiektowe*

Copyright©2021 Bogdan Kreczmer

---

Niniejszy dokument zawiera materiały do wykładu dotyczącego programowania obiektowego. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych prywatnych potrzeb i może on być kopowany wyłącznie w całości, razem z niniejszą stroną tytułową.



*Niniejsza prezentacja została wykonana przy użyciu systemu składu L<sup>A</sup>T<sub>E</sub>X oraz stylu beamer, którego autorem jest Till Tantau.*

Strona domowa projektu Beamer:

<http://latex-beamer.sourceforge.net>

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

## 2 Rysowanie diagramu klas

- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

- Opis czynności
- Rysowanie diagramu – krok po kroku

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

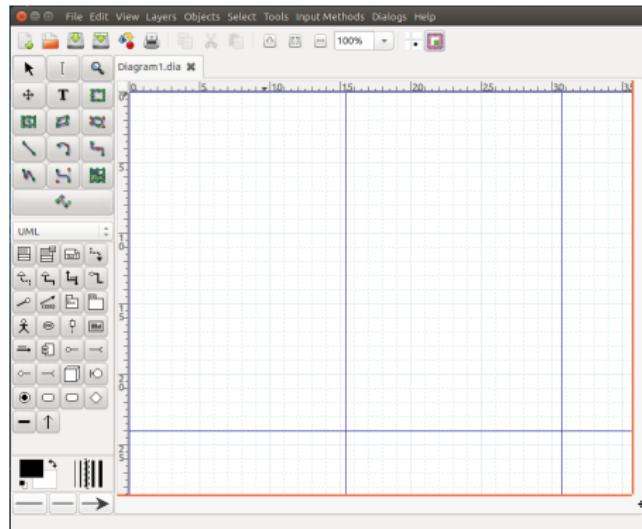
## 2 Rysowanie diagramu klas

- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

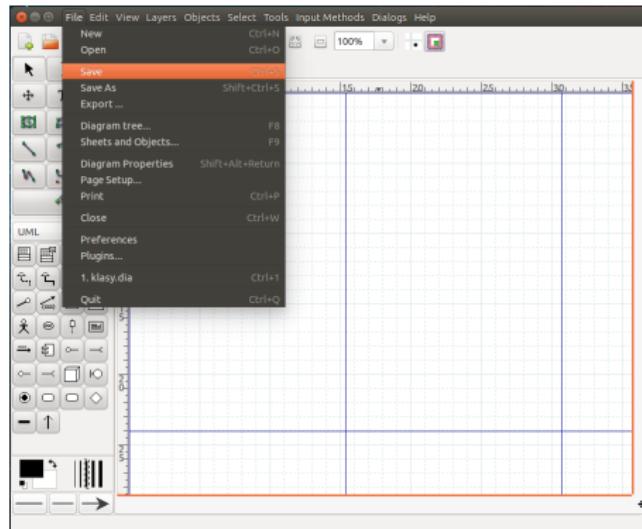
- Opis czynności
- Rysowanie diagramu – krok po kroku

# Główne okienko programu dia



W po lewej stronie w środkowej części znajduje się belka wyboru język pictogramów, który możemy stosować do tworzenia diagramu. Zazwyczaj domyślnie jest ustawiony UML, tak jak w tym przypadku.

# Zainicjalizowanie pliku



Dobrym pomysłem jest zainicjalizowanie pliku, do którego będzie składowany tworzony przez nas diagram. Możemy to zrobić wybierając właściwą pozycję menu lub posługując się skrótem Ctrl+S. Wprowadzona nazwa pliku pojawi się w tytule zakładki.

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

## 2 Rysowanie diagramu klas

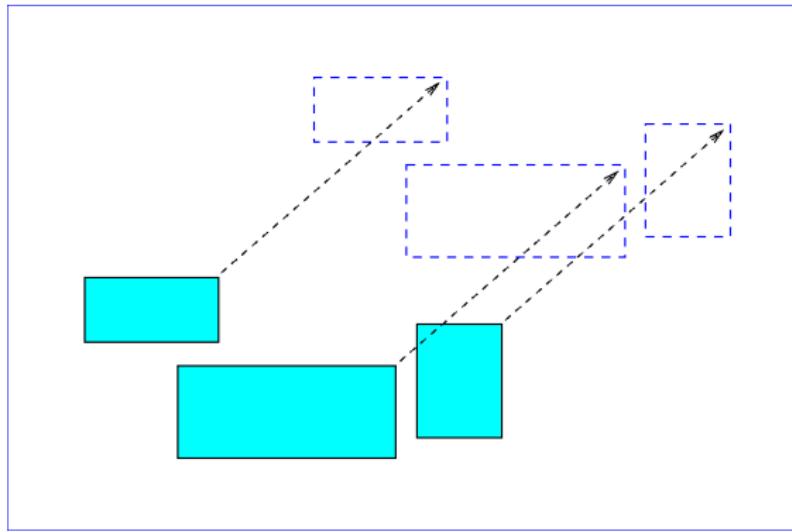
- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

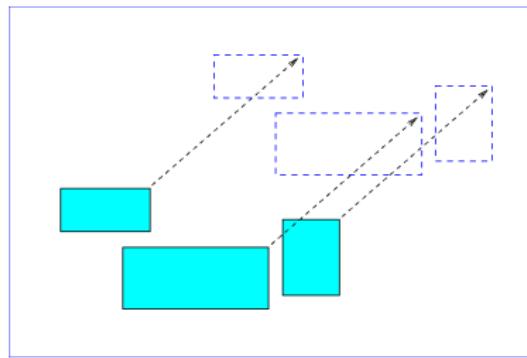
- Opis czynności
- Rysowanie diagramu – krok po kroku

# Opis problemu

Założymy, że chcemy zamodelować struktury danych dla prostego problemu. Mamy trzy prostokąty, o różnych wymiarach, które chcemy jednocześnie przesuwać o pewien wektor translacji.

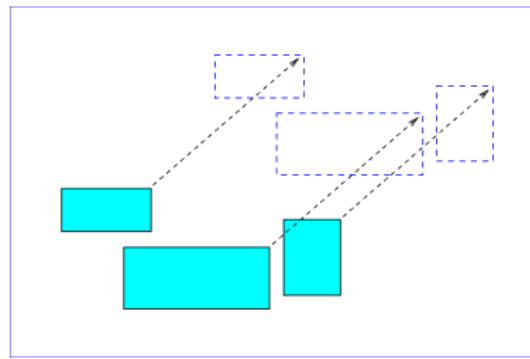


# Analiza problemu



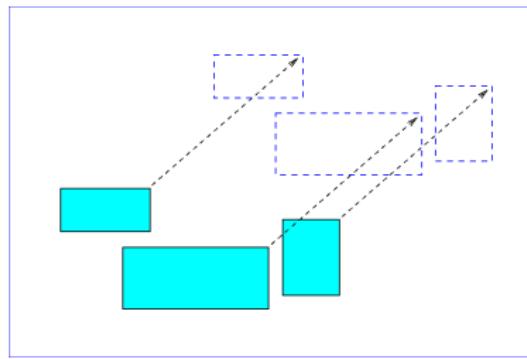
Aby zamodelować pojęcie prostokąta, stworzymy klasę Prostokąt.

# Analiza problemu



Prostokąt ma wierzchołki, które scharakteryzowane są współrzędnymi i niczym innym. Współrzędne te to nic innego jak współrzędne wektora zaczepionego w początku układu współrzędnych i poprowadzonego do punktu reprezentującego wierzchołek. Tak więc potrzebna jest nam klasa Wektor2D, którą wykorzystamy do reprezentowania wierzchołków w klasie Prostokat.

# Analiza problemu



Prostokąty są powiązane ze sobą, że operacja translacji dotyczy ich wszystkich. Znajdują się w pewnym umownym abstrakcyjnym świecie. Świat ten zamodelujemy za pomocą klasy Scena. Z tą klasą będzie związana operacja translacji.

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

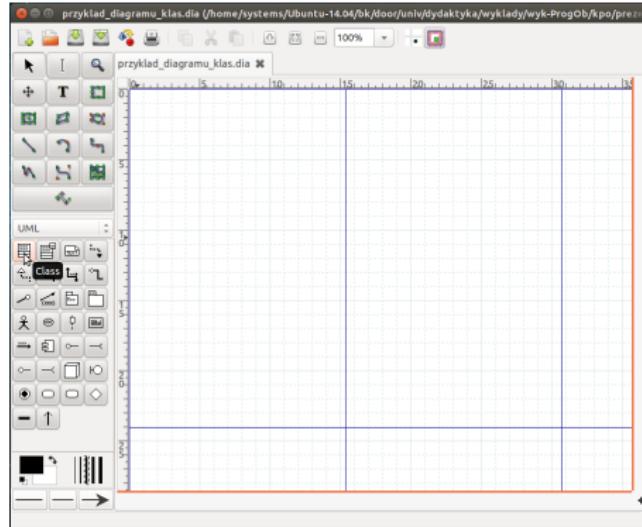
## 2 Rysowanie diagramu klas

- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

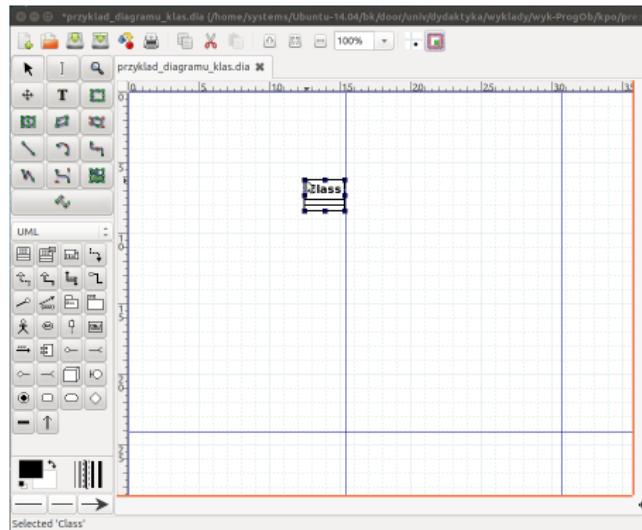
- Opis czynności
- Rysowanie diagramu – krok po kroku

# Wybór wstawianych symboli



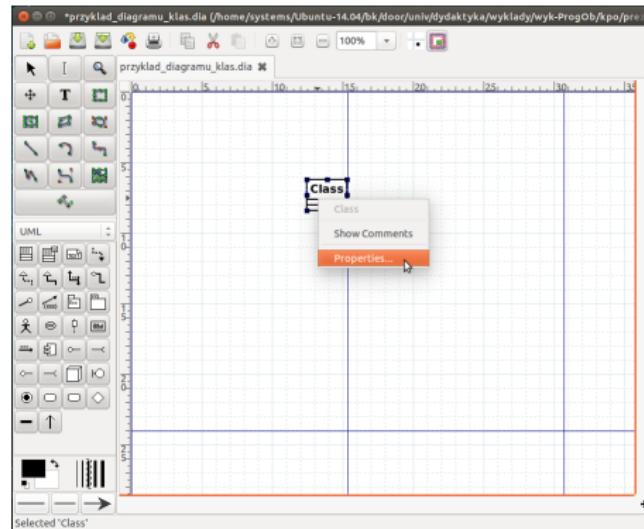
Aby przejść do rysowania symbolu klasy, musimy wybrać odpowiednie narzędzie.  
Zrobimy to poprzez naciśnięcie przycisku po lewej stronie, na którym znajduje się kurSOR myszki.

# Wstawianie opisu klasy



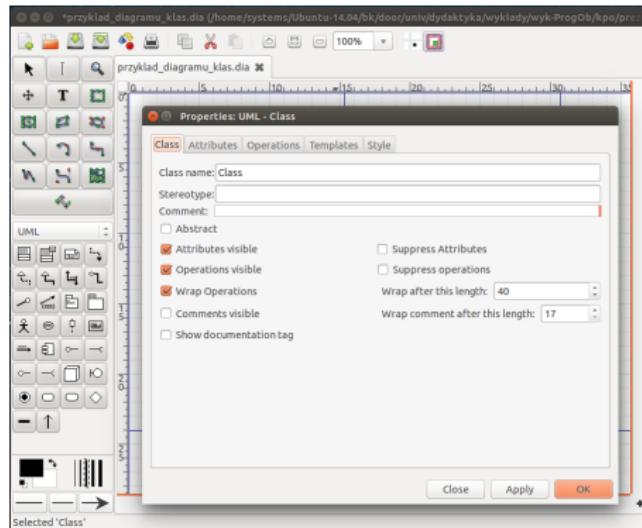
Klikając lewym przyciskiem myszki w obrębie pola rysunku wstawiamy graficzne oznaczenie klasy.

# Modyfikacja opisu klasy



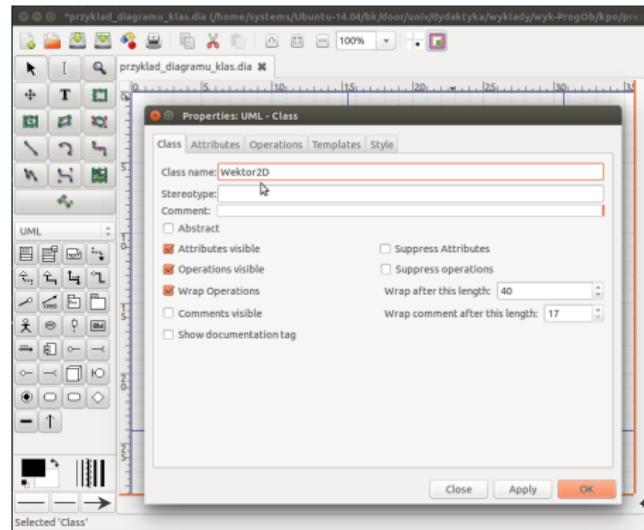
Klikając prawym przyciskiem myszki w obrębie oznaczenia klasy powoduje otwarcie podręcznego menu. W nim możemy wybrać opcje własności (ang. properties).

# Modyfikacja opisu klasy



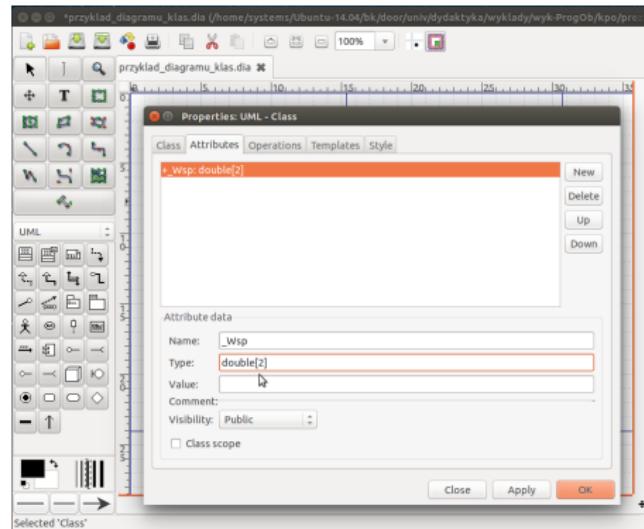
Powoduje to otwarcie okna własności, za pośrednictwem którego możemy zmienić . . . .

# Modyfikacja opisu klasy



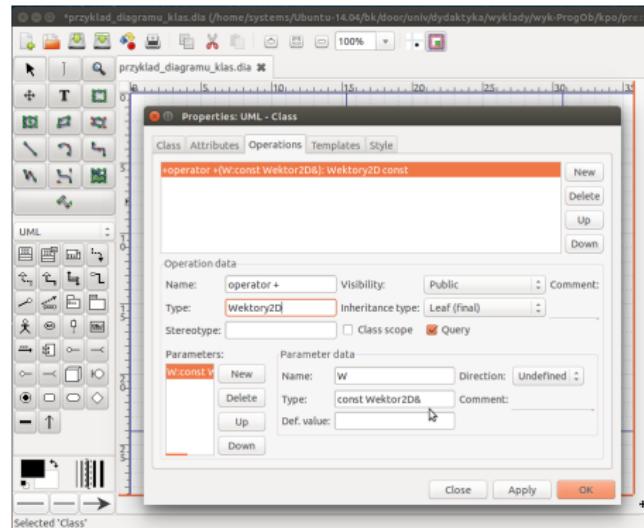
Powoduje to otwarcie okna własności, za pośrednictwem którego możemy zmienić nazwę klasy oraz to, które atrybuty i metody mają być widoczne itp.

# Modyfikacja opisu klasy



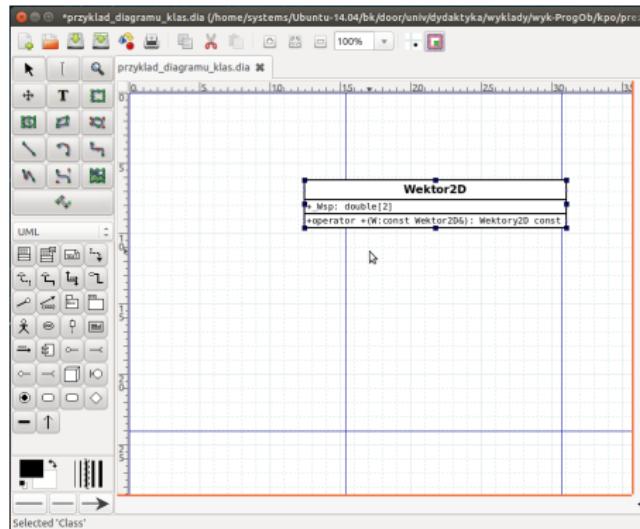
W kolejnej zakładce możemy dopisać nowe atrybuty.

# Modyfikacja opisu klasy



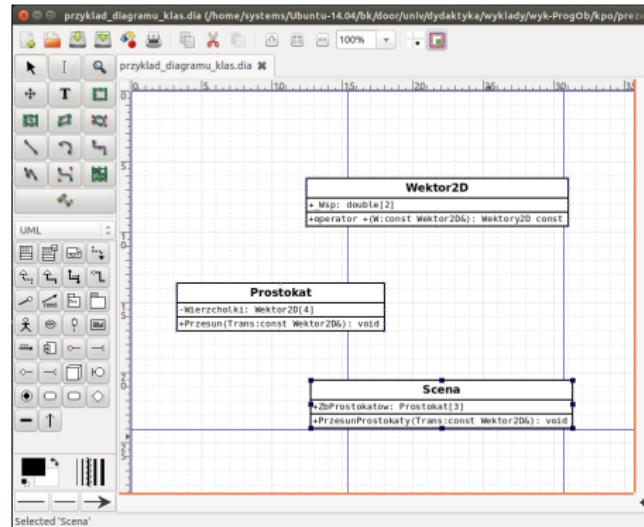
Przechodząc do następnej zakładki możemy dopisać operacje dla danej klasy.

# Modyfikacja opisu klasy



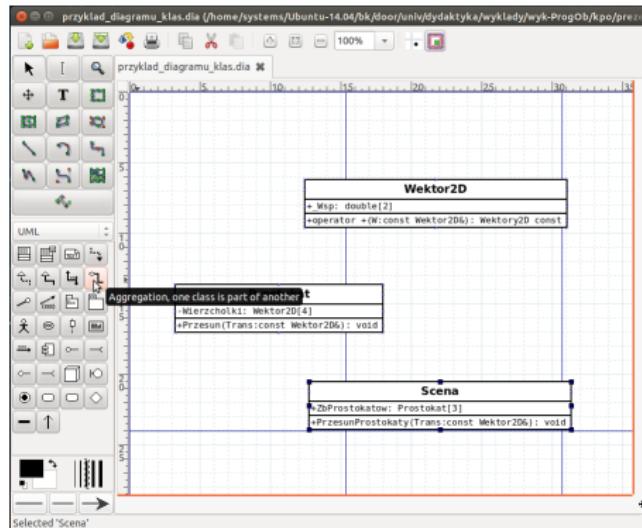
W końcowym rezultacie otrzymujemy wpisy atrybutów i operacji umieszczone w odpowiednich miejscach. Uwaga: ostatni `const` w zapisie operacji nie odnosi się do zwracanego typu. Odnosi się do metody, gdyż jest to operacja niemodyfikująca obiekt (zaznaczona została pozycja `Query` w opisie metody).

# Wstawianie klas



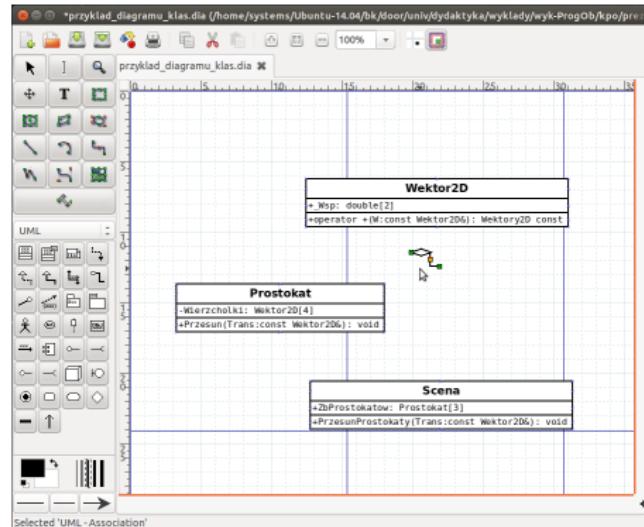
Analogicznie dodajemy pozostałe klasy.

# Zaznaczanie powiązań i relacji



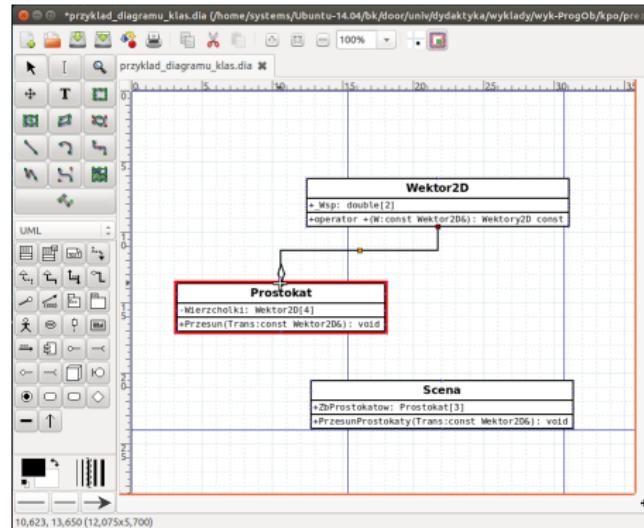
Teraz chcemy zaznaczyć relację kompozycji, która zachodzi między klasą Wektor2D i Prostokat. Przełączamy tryb rysowania naciskając odpowiedni klawisz narzędziowy po lewej stronie.

# Zaznaczanie powiązań i relacji



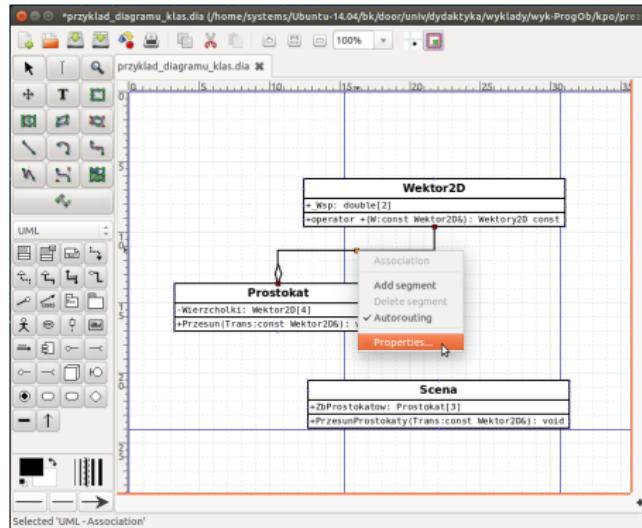
Klikając lewym przyciskiem wstawiamy rysunek agregacji (poźniej zmienimy go na symbol kompozycji).

# Zaznaczanie powiązań i relacji



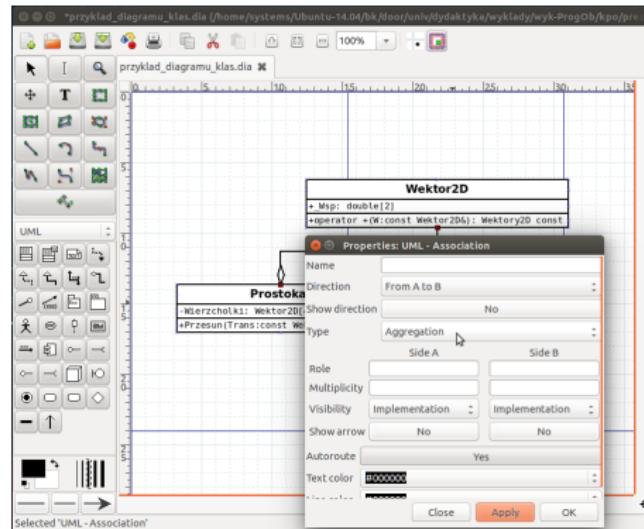
Następnie przeciągamy oba końce w miejsce, w których można dołączyć je do bloczków klas. Gdy trafimy na to miejsce, bloczek jest obrysowywany czerwoną obwódką. Miejsca łączenia są zaznaczone bardzo małymi znakami w kształcie znaku  $\times$  na brzegu oznaczenia klasy.

# Zaznaczanie powiązań i relacji



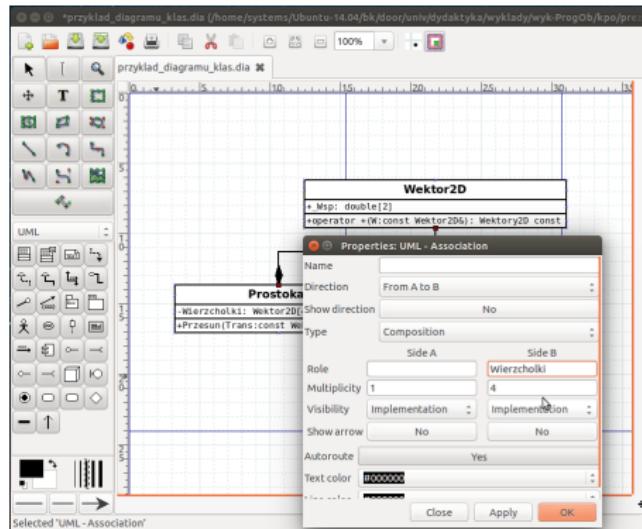
Teraz zmienimy oznaczenie agregacji na kompozycję. W tym celu kurSOR myszy naprowadzamy na linię relacji agregacji i klikamy lewym klawiszem, aby go zaznaczyć, a następnie prawym klawiszem myszy, aby otworzyć podręczne menu. Z niego wybieramy pozycję własności.

# Zaznaczanie powiązań i relacji



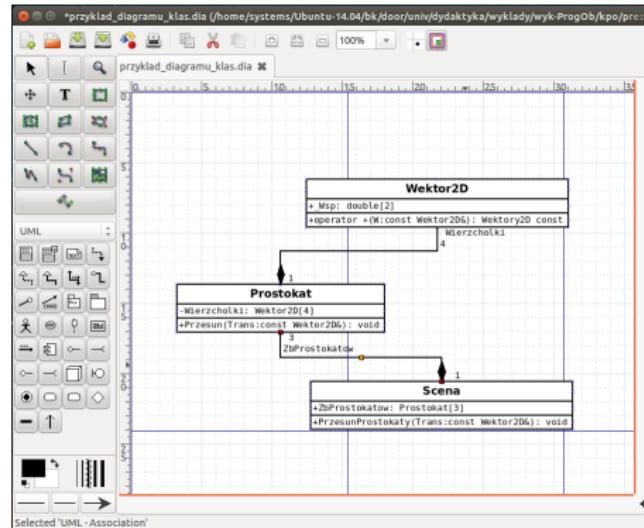
W belce wyboru zmieniamy Aggregation na Composition. Po naciśnięciu klawisza Apply kolor grota zmienia się na czarny.

# Zaznaczanie powiązań i relacji



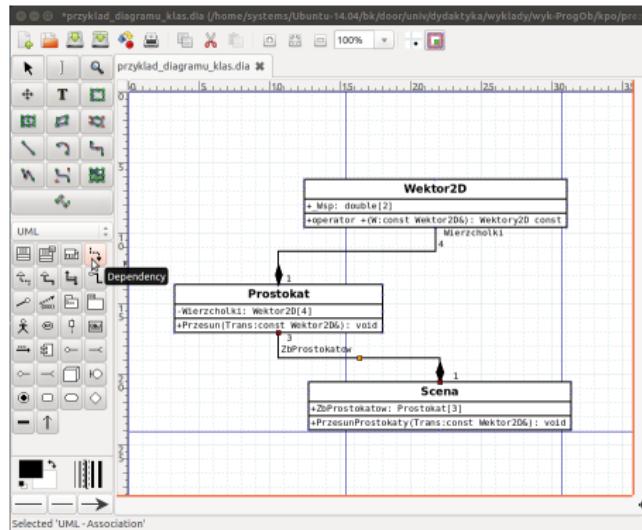
Dodatkowo możemy wpisać wielokrotności oraz rolę pełnią obiekty danej klasy względem drugiej. W naszym przypadku oznacza to wpisanie nazwy pola, którego elementami są dane obiekty.

# Zaznaczanie powiązań i relacji



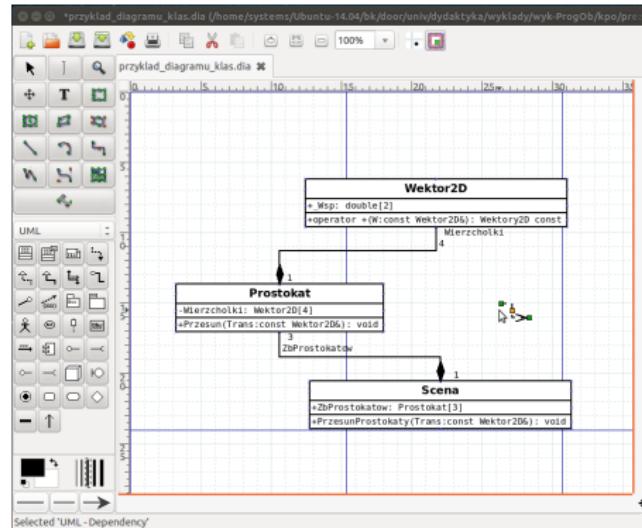
Analogicznie wprowadzamy relację kompozycji między klasą Prostokat i Scena.

# Relacja zależności



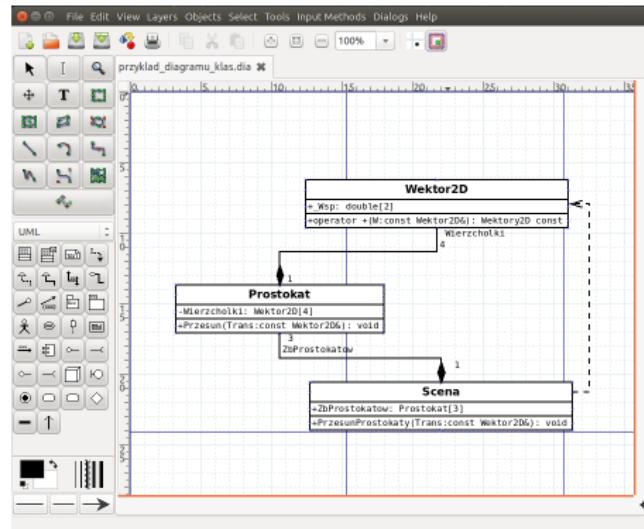
Ze względu na to, że w klasie Scena jest operacja, której parametrem jest obiekt klasy Wektor2D, musimy jeszcze zaznaczyć bezpośrednią zależność tej klasy Scena od klasy Wektor2D. Zmieniamy najpierw tryb rysowania naciskając odpowiedni klawisz narzędziowy po lewej stronie.

# Relacja zależności



Klikając lewym przyciskiem myszy w obszarze rysowania diagramu wstawiamy symbol zależności.

# Relacja zależności



Następnie przeciągamy końce tego symbolu do odpowiednich punktów łączenia z oznaczeniami klas, co kończy rysunek diagramu.

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

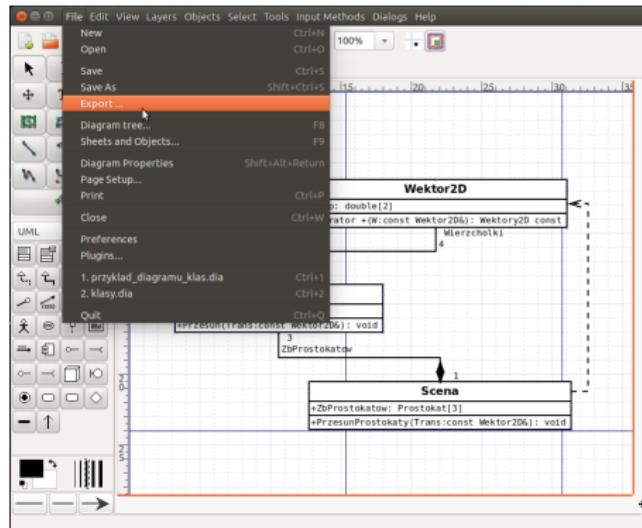
## 2 Rysowanie diagramu klas

- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

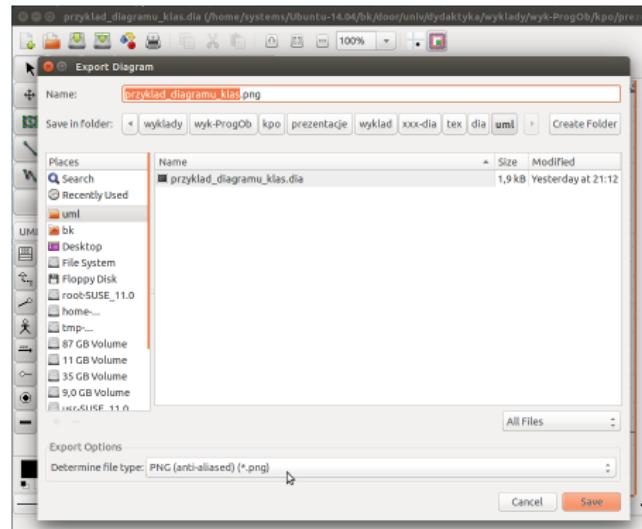
- Opis czynności
- Rysowanie diagramu – krok po kroku

# Eksport ...



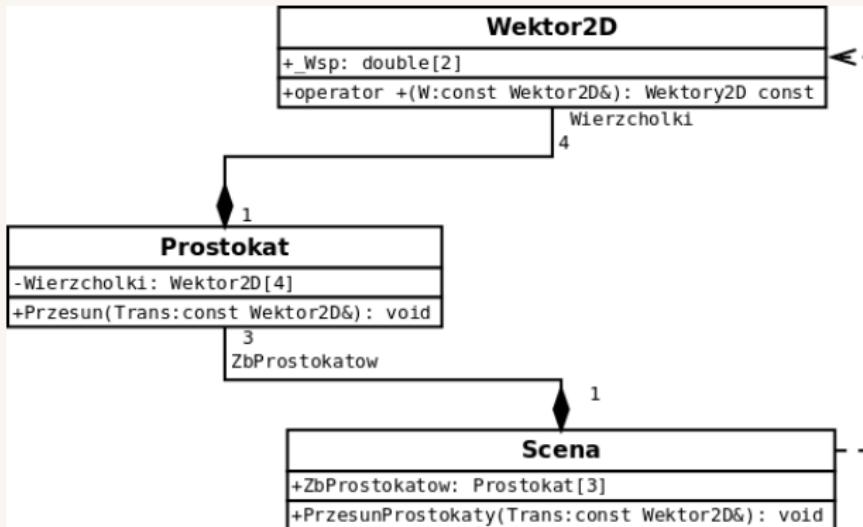
Tak utworzony diagram możemy wyeksportować do pliku o formacie graficznym.

# Eksport do pliku graficznego



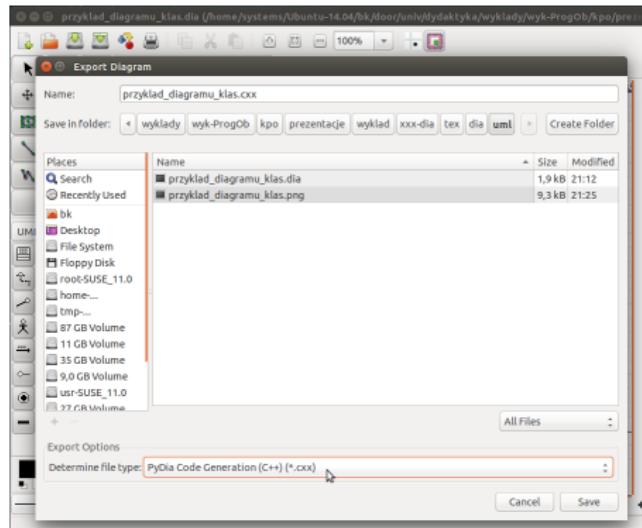
W okienku dialogowym w dolnej części znajduje się belka wyboru formatu. Najlepiej wybrać **PNG (anti-aliased)**.

# Wynik eksportu



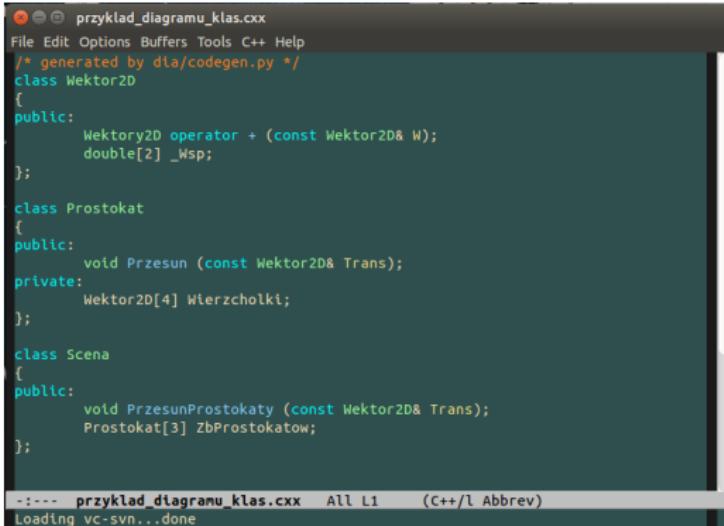
Tak wygląda wyeksportowany diagram

# Eksport do kodu C++



Można również ten plik wyeksportować do kodu w języku C++.

# Wyeksportowany kod C++



```
przyklad_diagramu_klas.cxx
File Edit Options Buffers Tools C++ Help
/* generated by dia/codegen.py */
class Wektor2D
{
public:
    Wektory2D operator + (const Wektor2D& W);
    double[2] _Wsp;
};

class Prostokat
{
public:
    void Przesun (const Wektor2D& Trans);
private:
    Wektor2D[4] Wierzcholki;
};

class Scena
{
public:
    void PrzesunProstokaty (const Wektor2D& Trans);
    Prostokat[3] ZbProstokatow;
};

----- przyklad_diagramu_klas.cxx  All L1      (C++/l Abbrev)
Loading vc-svn...done
```

Niestety otrzymany kod nie jest dokładnie taki jaki byśmy oczekiwali. W przypadku tablic konieczne jest przeprowadzenie korekty.

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

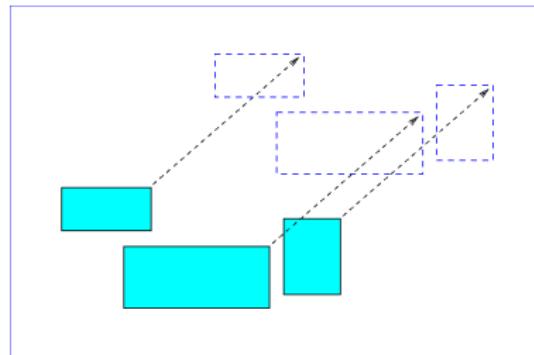
## 2 Rysowanie diagramu klas

- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

- Opis czynności
- Rysowanie diagramu – krok po kroku

# Główne okienko programu dia



Algorytm, który należy opisać, jest bardzo prosty. Należy dokonać iteracji wszystkich dostępnych prostokątów i do współrzędnych wierzchołków każdego z nich dodać wektor translacji.

# Plan prezentacji

## 1 Program dia

- Rozpoczęcie pracy

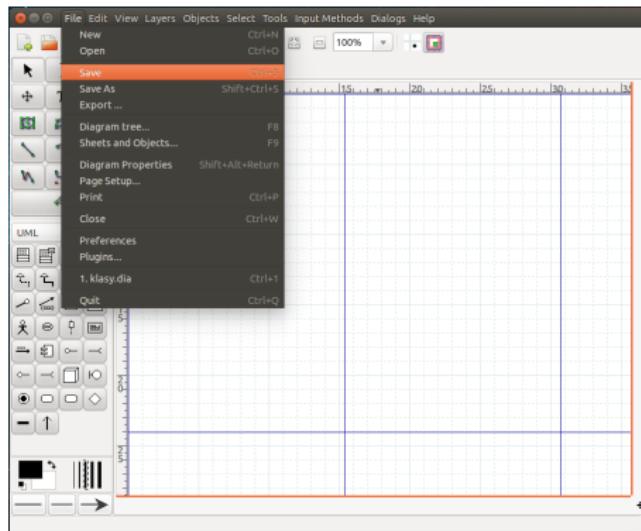
## 2 Rysowanie diagramu klas

- Opis problemu
- Diagram klasy – krok po kroku
- Eksportowanie diagramu

## 3 Rysowanie diagramu czynności

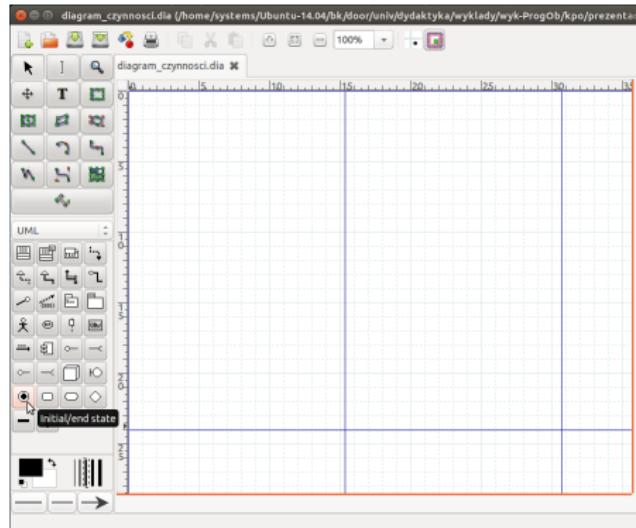
- Opis czynności
- Rysowanie diagramu – krok po kroku

# Zainicjalizowanie pliku



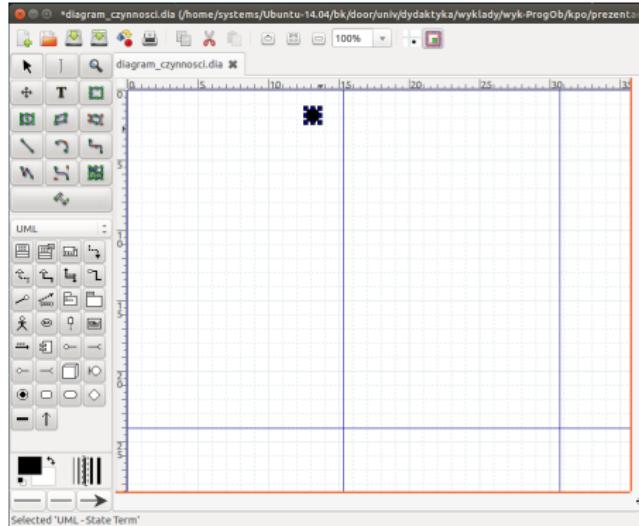
Tak jak jak w przypadku postępowania przy rysowaniu diagramu klas, dobrym pomysłem jest zainicjalizowanie pliku, do którego będzie składowany tworzony przez nas diagram. Możemy to zrobić wybierając właściwą pozycję menu lub posługując się skrótem **Ctrl+S**. Wprowadzona nazwa pliku pojawi się w tytule zakładki.

## Rysowanie stanu początkowego – wybór trybu



Na początku przełączamy tryb rysowanie na wprowadzanie symboli stanu początkowego.

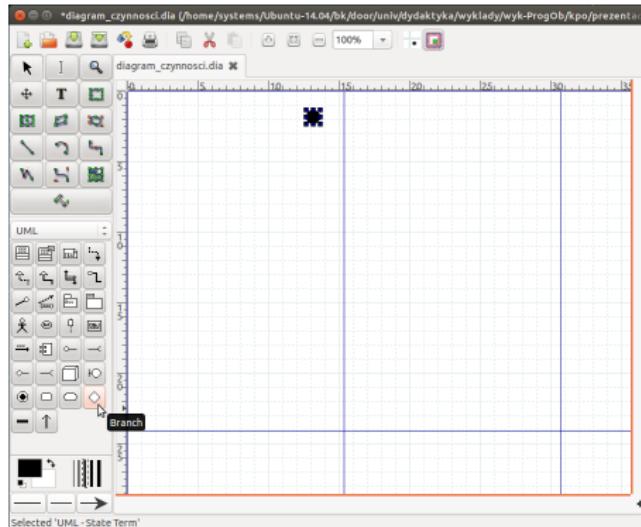
# Rysowanie stanu początkowego – wprowadzenie symbolu



Następnie klikając lewym przyciskiem myszy w obszarze rysowania diagramu wprowadzamy symbol stanu początkowego. Ma on kształt zaczernionego koła. Kwadraciki wokół niego oznaczają, że w tym momencie jest to element wyselekcyjowany, co pozwala np. na jego *uchwycenie* i przesunięcie.

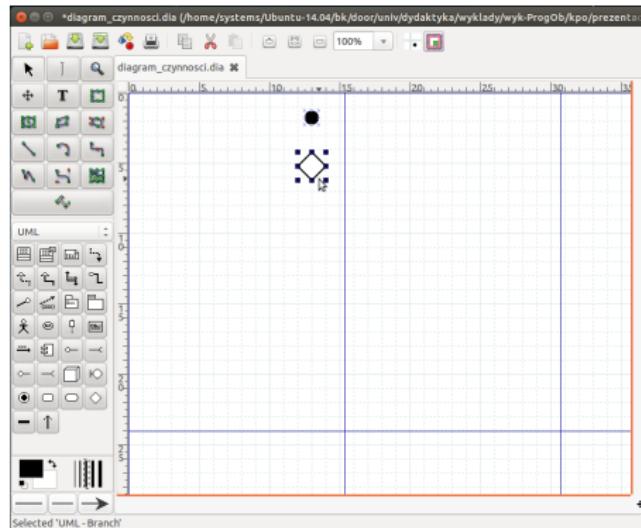
# Rysowanie stanu początkowego – wprowadzenie symbolu

Aby przejść do trybu wprowadzania węzłów decyzyjnych naciskamy klawisz po lewej stronie w dolnej części ekranu.



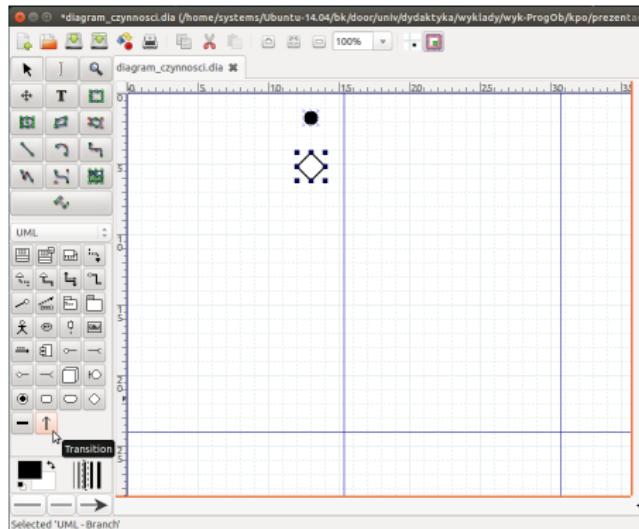
Konstruując zestaw czynności, które należy wykonać aby przesunąć prostokąty, dobrze jest sprawdzić, czy w ogóle mają one być wykonane. Jeżeli wektor translacji jest wektorem zerowym wówczas nic nie trzeba robić. Potrzebne będzie więc wprowadzenie węzła decyzyjnego z odpowiednim zapisem warunków.

# Rysowanie stanu początkowego – wprowadzenie symbolu



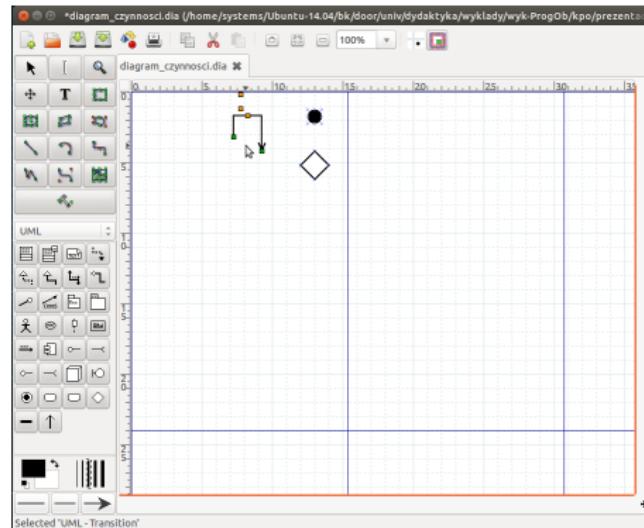
Wprowadzamy go do diagramu w analogiczny sposób jak inne symbole.

# Przejście między czynnościami – wybór symbolu



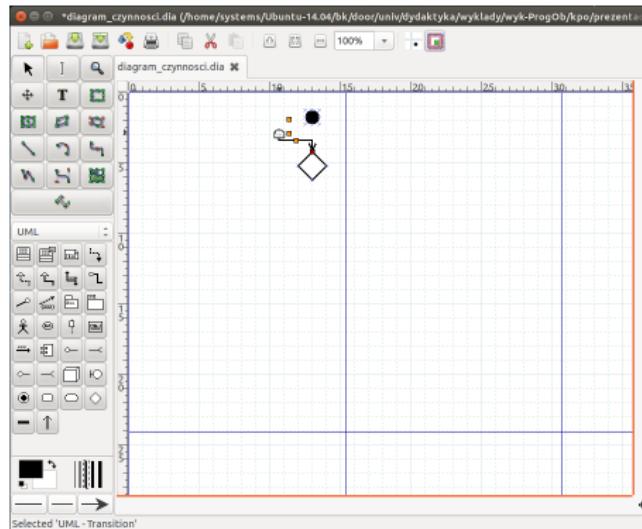
Zanim wpiszemy odpowiedni warunek najpierw połączmy go poprzez wskazanie przejścia między stanem początkowym, a danym węzłem. W tym celu wybieramy odpowiedni tryb wprowadzania symboli tranzycji (przejścia).

# Przyjście między czynnościami – wprowadzanie symbolu



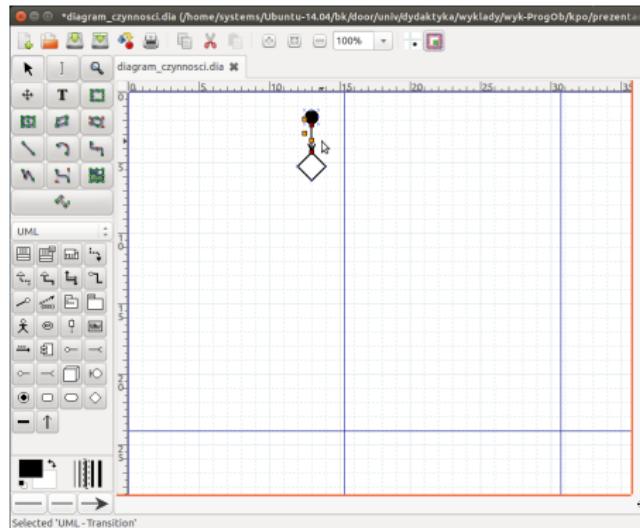
Symbol tranzycji wprowadzamy do diagramu tak jak każdy inny symbol.

# Przyjście między czynnościami – wprowadzanie symbolu



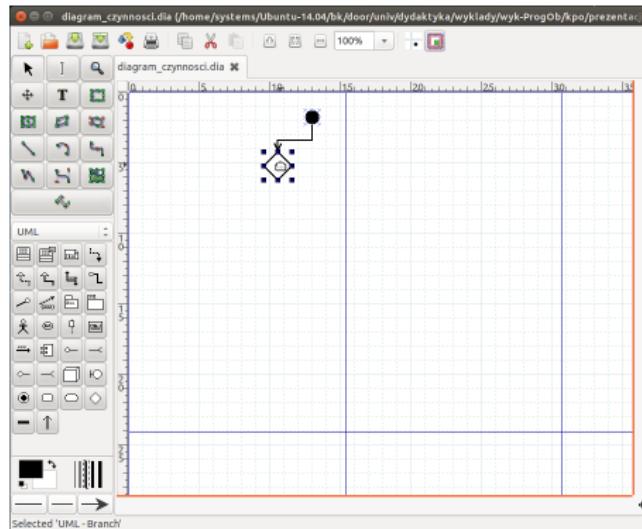
Następnie posługując się zaznaczonymi na linii strzałki punktami uchwytu odpowiednio ją przesuwamy i łączymy z punktami łączenia poszczególnych symboli.

# Przyjście między czynnościami – wprowadzanie symbolu



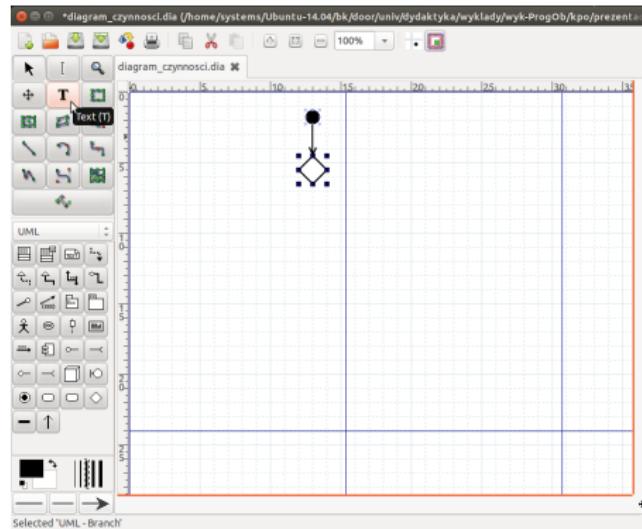
W końcowym efekcie mamy połączenie między stanem początkowym i węzłem warunku.

# Przyjście między czynnościami – wprowadzanie symbolu



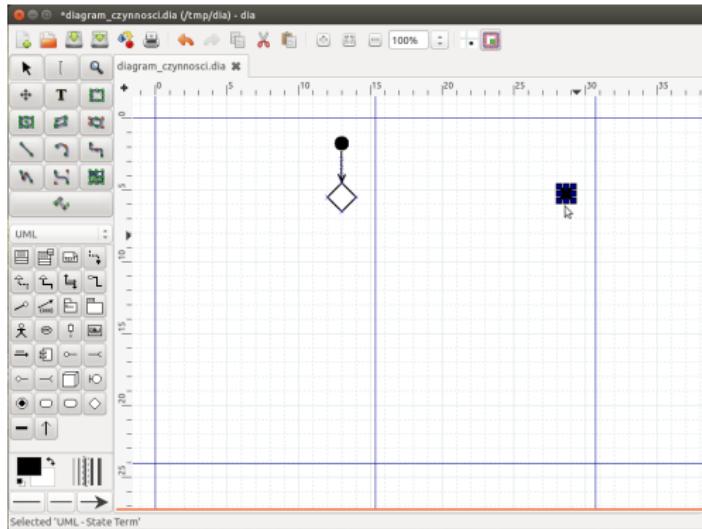
Jeżeli linia przejścia dobrze jest dołączona do poszczególnych elementów, to po jego *uchwyceniu* i przesunięciu linia będzie wciąż do niego *przykleiona*.

## Tekst warunku



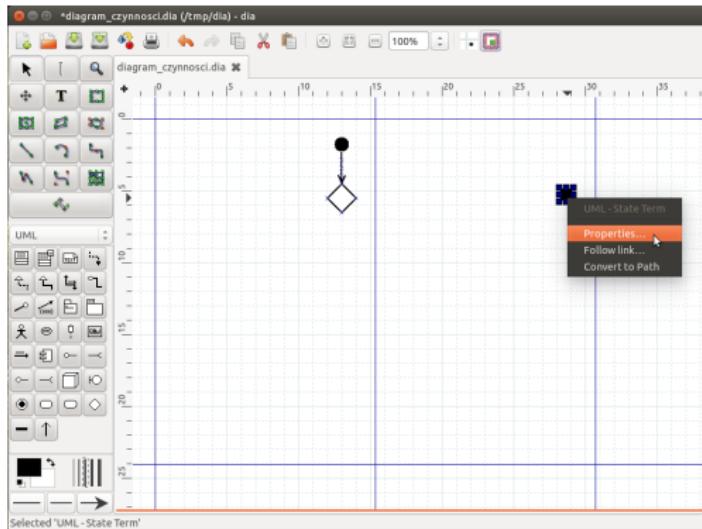
Tekst warunku możemy wprowadzić na dwa sposoby. W przypadku prostych warunków, ogranicza się to do odpowiedniego wpisu wzdłuż linii przepływu sterowania. Bardziej złożone warunki zapisujemy w postaci specjalnej notatki. Oba sposoby zostaną przedstawione w dalszej części.

# Stan końcowy



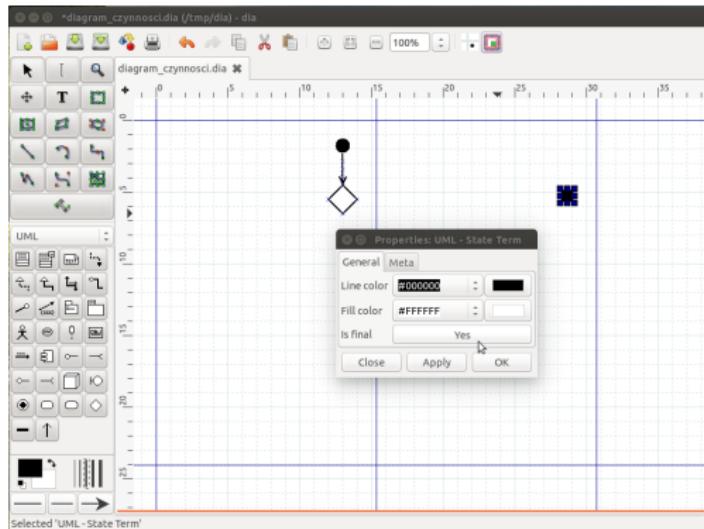
Założymy, że gdy wektor translacji jest wektorem zerowym to procedura postępowania zostaje zakończona, a funkcja, która będzie to realizowała zwróci wartość `false`. Aby to pokazać, najpierw na diagramie umieścimy oznaczenie stanu końcowego korzystając z tego samego trybu jak przy wprowadzaniu stanu początkowego.

# Stan końcowy



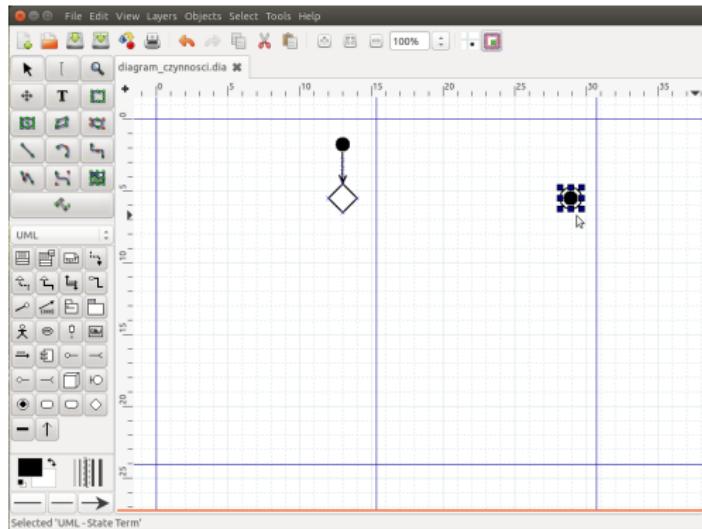
Jednak aby wprowadzone oznaczenie było faktycznie oznaczeniem stanu końcowego musimy skorzystać z podręcznego menu skojarzonego z tym obiektem i przejść do jego własności.

# Stan końcowy



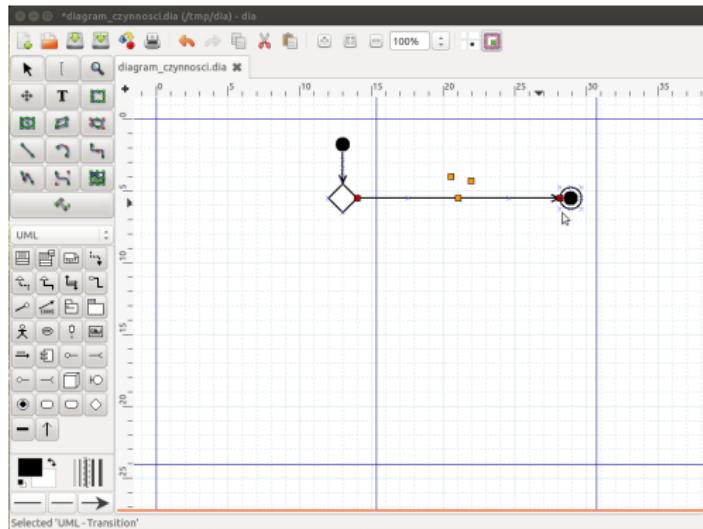
Właściwy efekt uzyskamy wtedy, w pozycji Is final wybierzemy Yes.

# Stan końcowy



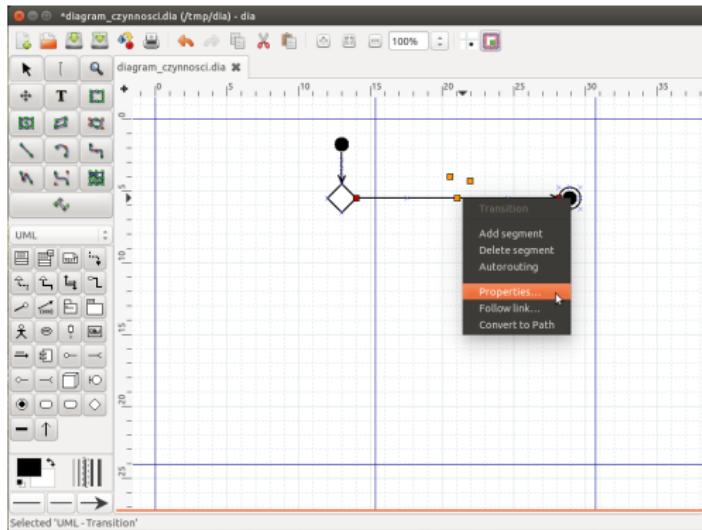
Po zaakceptowaniu uzyskamy oznaczenie stanu końcowego.

# Tranzycja z warunkiem dozoru



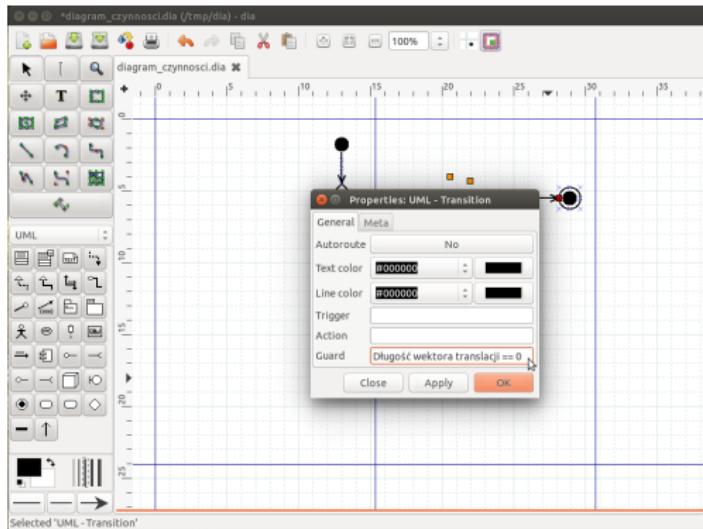
Następnie dokonujemy połączenie węzła decyzyjnego ze stanem końcowym.

# Tranzycja z warunkiem dozoru



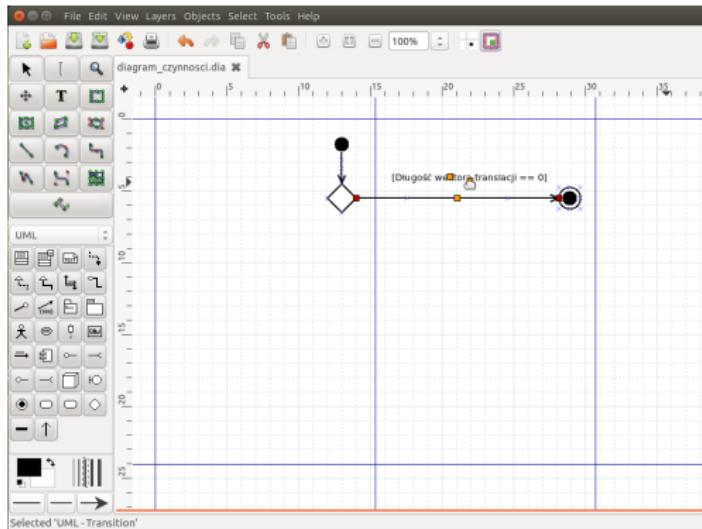
Aby wprowadzić treść warunku, którego spełnienie spowoduje, że sterowanie będzie mogło przejść *wzdłuż* danej tranzycji, należy skorzystać z podręcznego menu skojarzonego z tą tranzycją i przejść do jej własności. Warunek ten nazywany jest **warunkiem dozoru**.

# Tranzycja z warunkiem dozoru



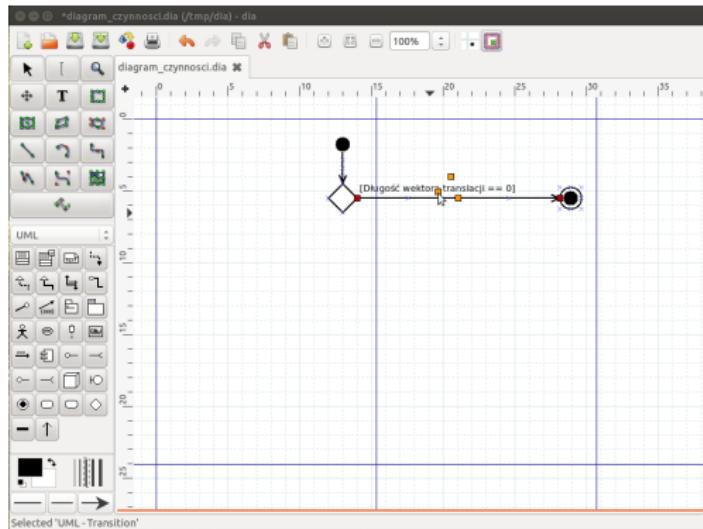
W polu **Guard** wpisujemy treść **warunku dozoru**.

# Tranzycja z warunkiem dozoru



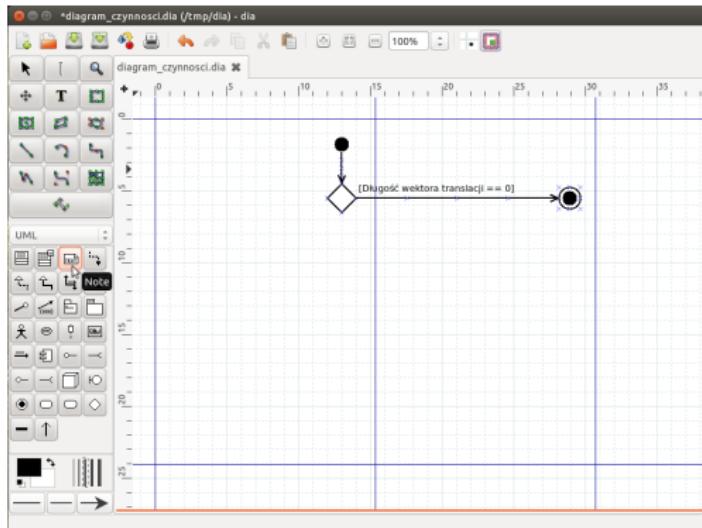
Tekst zazwyczaj nie pojawi się w miejscu, które zapewniałoby dobrą czytelność diagramu. Należy więc go *uchwycić* i przesunąć tak jak każdy inny symbol graficzny na diagramie.

# Tranzycja z warunkiem dozoru



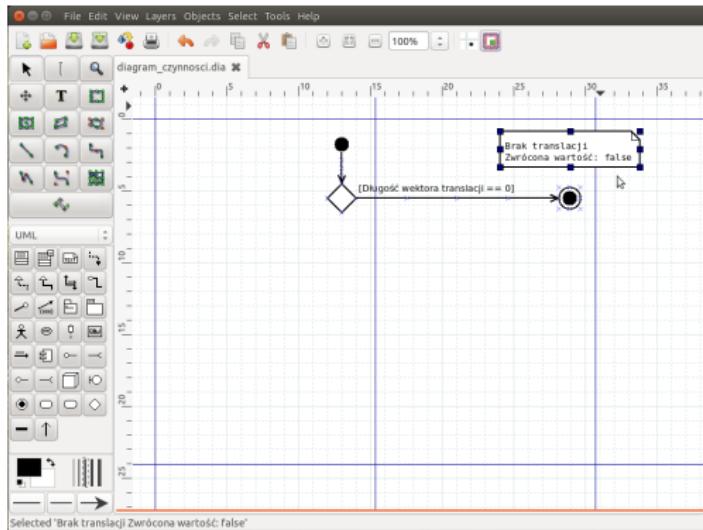
Warunek powinien być umieszczony rozsądnie blisko węzła warunku.

# Tworzenie notatki



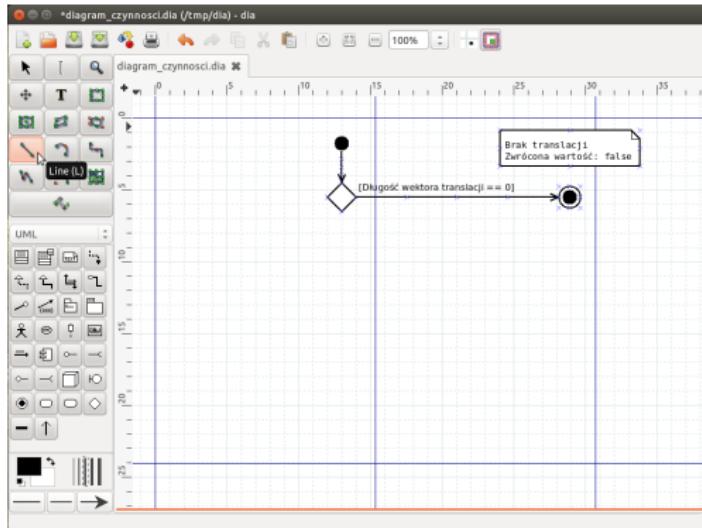
Jeżeli będzie kilka stanów końcowych, to dobrze jest z nim skojarzyć notatkę, w której zawrzemy opis danego stanu. W tym celu wybieramy tryb tworzenia notatek (patrz panel narzędziowy po lewej stronie).

# Tworzenie notatki



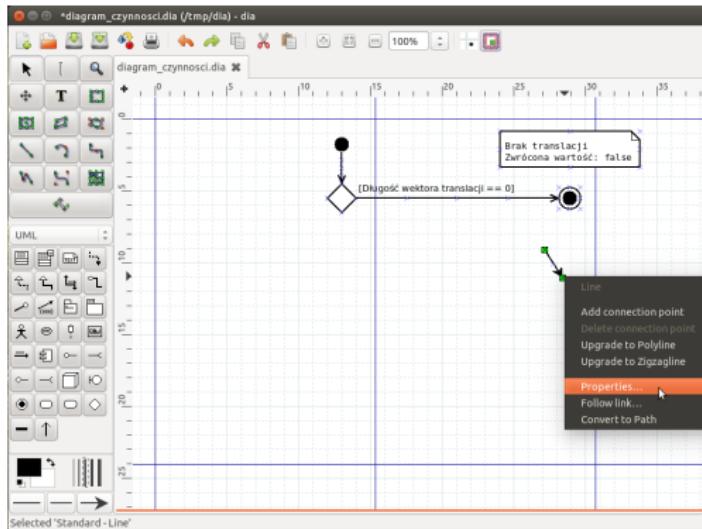
Wpisujemy odpowiednią zawartość notatki i umieszczamy ją w pobliżu elementu, którego ma dotyczyć.

# Łączenie notatki z symbolem graficznym



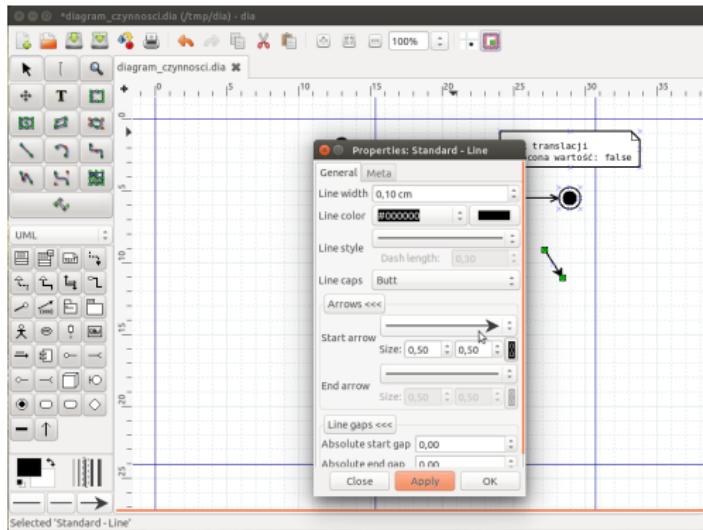
Notatkę należy skojarzyć z obiektem graficznym, do którego się odnosi, w tym przypadku symbolem stanu końcowego. Realizujemy to poprzez połączenie jej z danym symbolem przerywaną linią. W tym celu wybieramy tryb rysowania linii (patrz: panel narzędziowy po lewej stronie).

# Łączenie notatki z symbolem graficznym



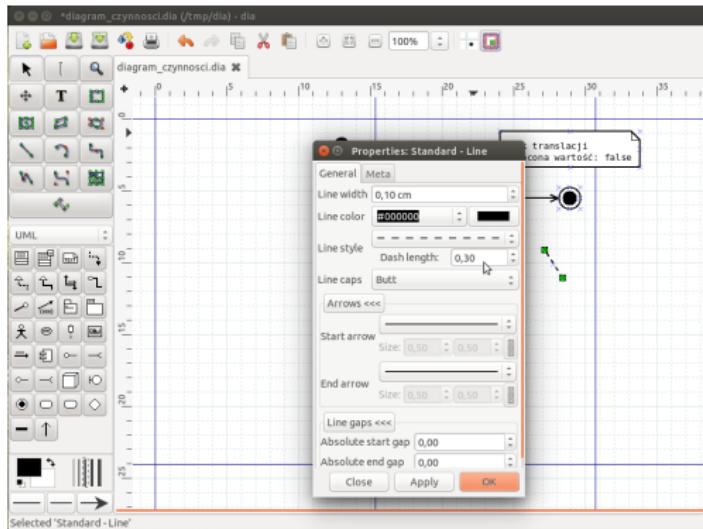
Domyślnie w trybie rysowania linii do diagramu wstawiana jest linia zakończona strzałką. Aby ją zmodyfikować należy przejść do jej właściwości (naciśnięcie prawego klawisza myszki).

# Łączenie notatki z symbolem graficznym



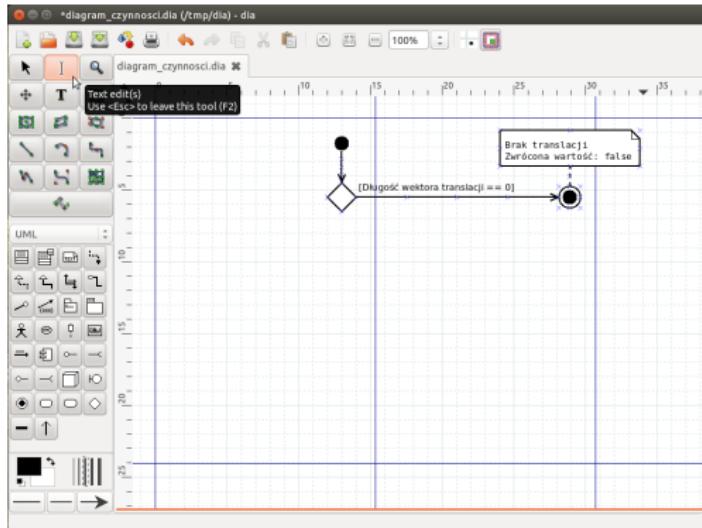
Należy zmodyfikować styl linii oraz pozycję Start arrow.

# Łączenie notatki z symbolem graficznym



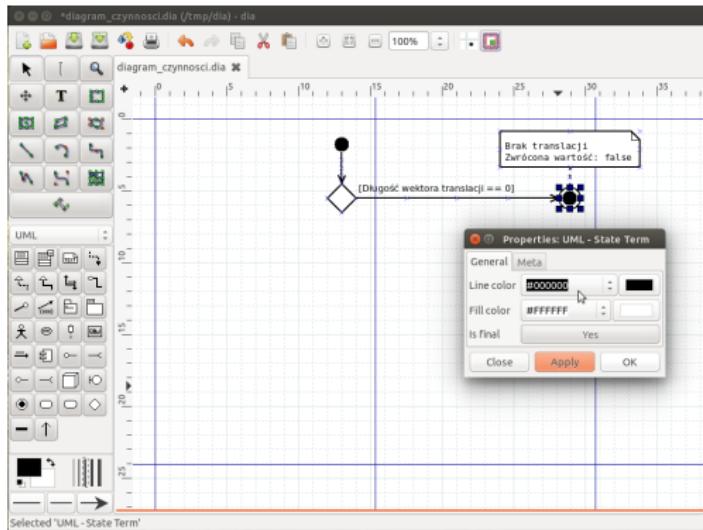
Dobrze jest również odpowiednio zmodyfikować **Dash length**, aby długość przerywanych kresek była odpowiednio mała.

# Łączenie notatki z symbolem graficznym



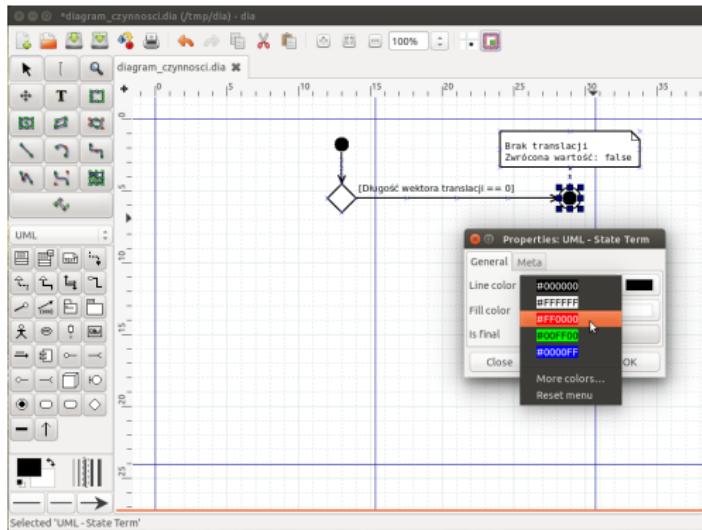
Następnie wykorzystując stworzoną linię łączymy nią notatkę z symbolem stanu końcowego. Jeżeli chcielibyśmy edytować tekst notatki, to wybieramy tryb edycji (patrz wskazany klawisz po lewej stronie). Dotyczy to wszystkich tekstowych obiektów.

# Modyfikacja wyglądu węzła stanu końcowego



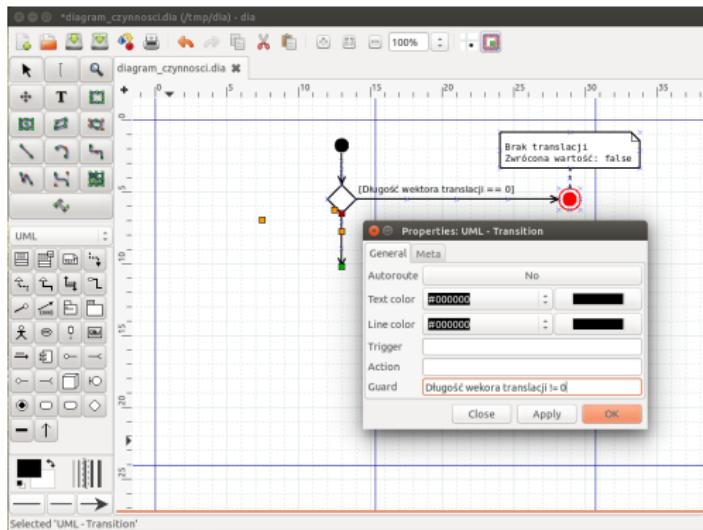
Chć lepiej odzwierciedlić znaczenie danego stanu, dobrze jest skorzystać z możliwości zmiany koloru węzła końcowego. W tym celu korzystamy z listy wyboru pozycji Line color.

# Modyfikacja wyglądu węzła stanu końcowego



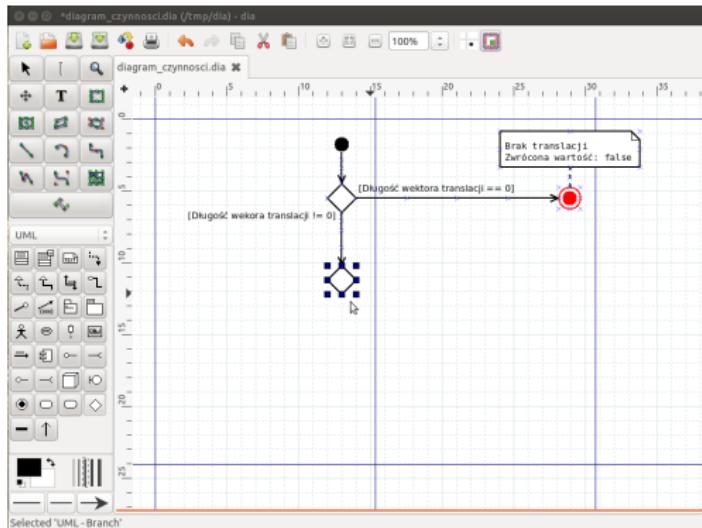
W tym przypadku dobrym wyborem wydaje się być kolor czerwony. Można też wprowadzić dodatkowe kolory spoza widocznej listy.

# Następny warunek



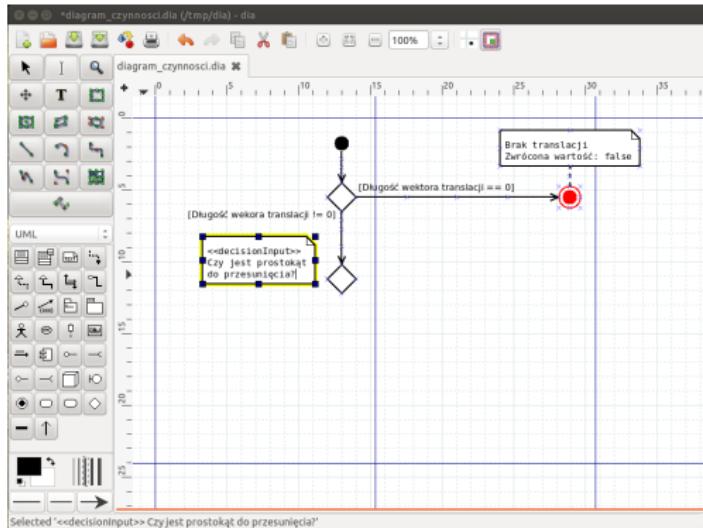
Tworzony drugą ścieżkę przepływu sterowania. Ze względu na to, że wychodzi ona z węzła decyzyjnego, dodajemy odpowiedni warunek dozoru. Tak jak wcześniej, wpisujemy go korzystając z dialogu modyfikacji własności danego elementu.

# Następny warunek



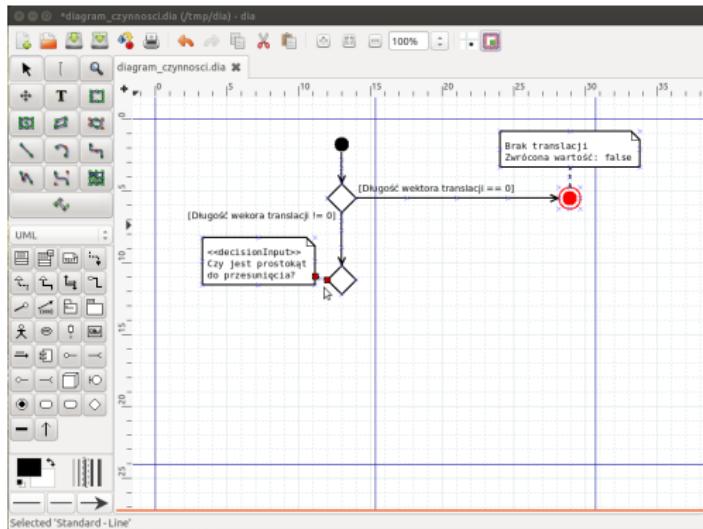
Postępując analogicznie jak wcześniej tworzymy następny następny węzeł decyzyjny. Tym razem jednak warunek, który będzie z nim związany, zapiszemy inaczej.

# Warunek dozoru



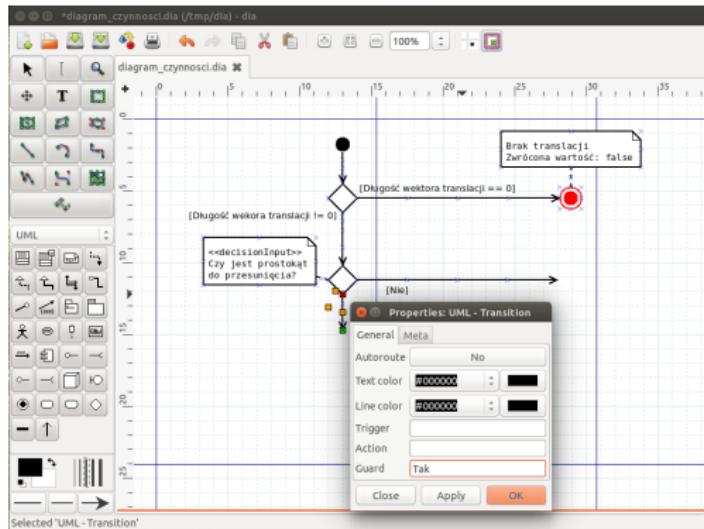
W przypadku, gdy warunek jest dłuższym zdaniem, możemy go zapisać w postaci notatki. Pragnąc zasygnalizować, że jest to specyficzna notatka odnosząca się do węzła decyzyjnego, umieszczamy na początku nagłówek **<<decisioinInput>>**. Określa on tzw. stereotyp, a więc specjalny typ danego piktogramu.

# Warunek dozoru



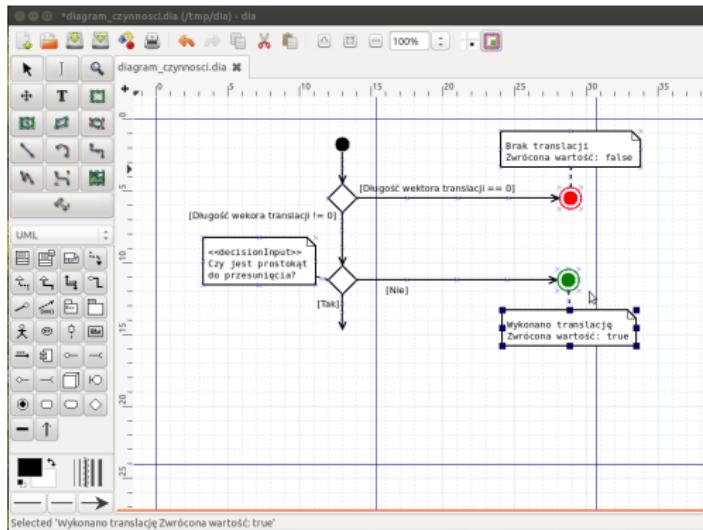
Notatkę z zapisem warunku łączmy z węzłem decyzyjnym w analogiczny sposób, jak to miało miejsce w przypadku stanu końcowego.

# Warunek dozoru



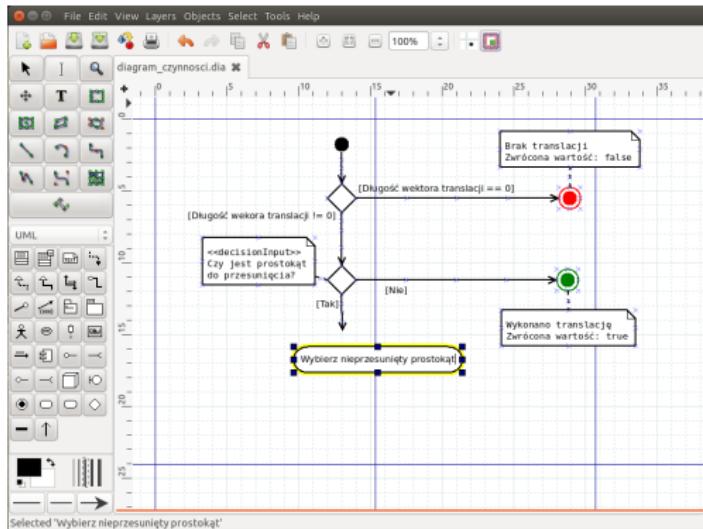
Tym razem warunek dozoru może zostać znaczco skrócony. W tym konkretnym przypadku wystarczą słowa TAK lub NIE, które są odpowiedzią na pytanie połączone z węzłem decyzyjnym.

# Warunek dozoru



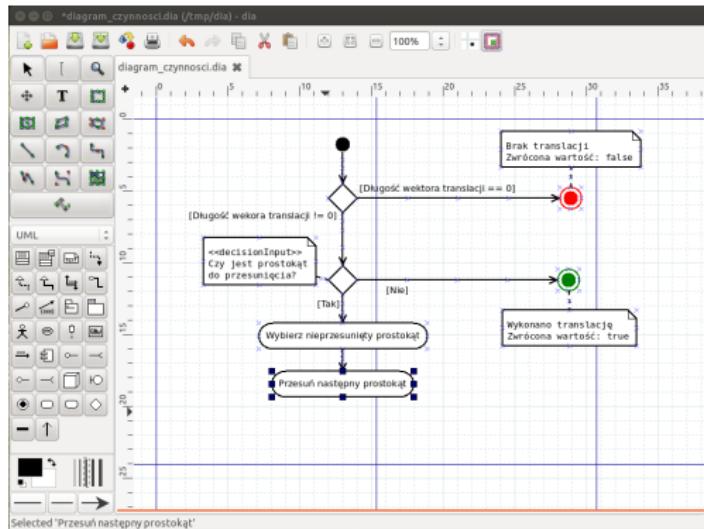
Dodajemy nowy stan końcowy. Tym razem jednak odnosi się on do odmiennej sytuacji, którą możemy opisać, analogicznie jak wcześniej, w notatce. Gdy pojawia się kilka różnych stanów końcowych, oprócz umieszczenia właściwych notatek, dobrze jest je również zróżnicować kolorystycznie korzystając z własności tych symboli.

# Wprowadzanie oznaczenia czynności



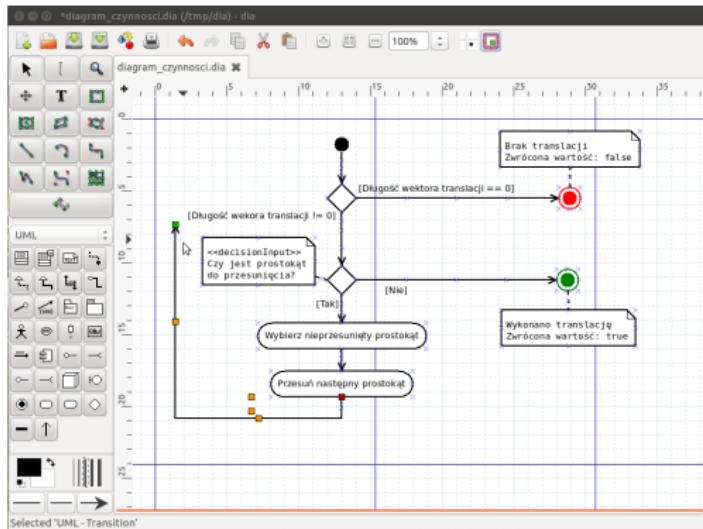
Przechodzimy następnie do wyboru trybu wprowadzania czynności. Umieszczamy w diagramie oznaczenie nowej czynności i wpisujemy właściwy tekst.

# Wprowadzanie oznaczenia czynności



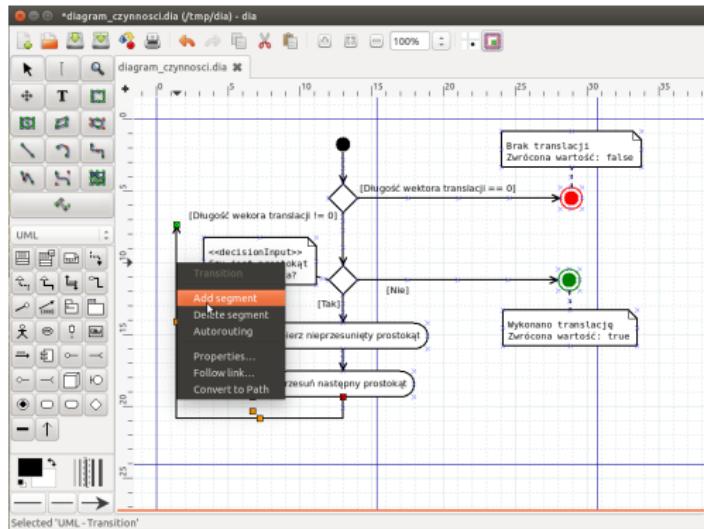
Umieszczamy wprowadzoną czynność w odpowiednim miejscu i łączymy z nią tranzycję.  
Analogicznie postępujemy w przypadku kolejnego oznaczenia czynności.

# Modyfikacja kształtu tranzycji



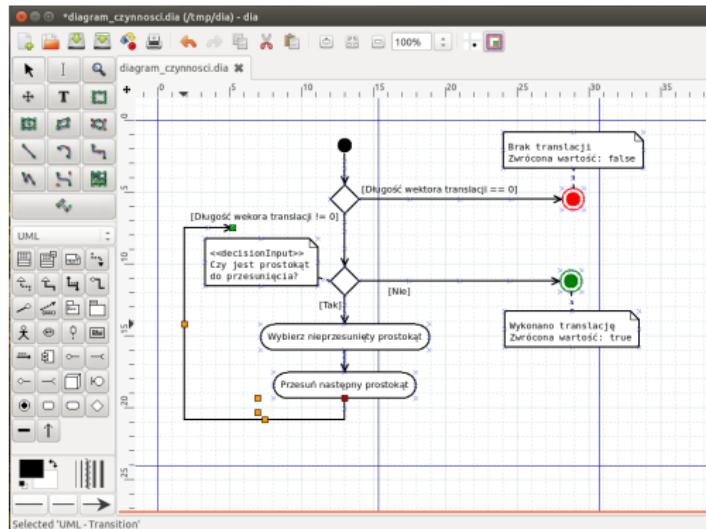
Teraz musimy zapętlić procedurę przesuwania prostokątów. Jednak tranzycja domyślnie ma tylko trzy odcinki.

# Modyfikacja kształtu tranzycji



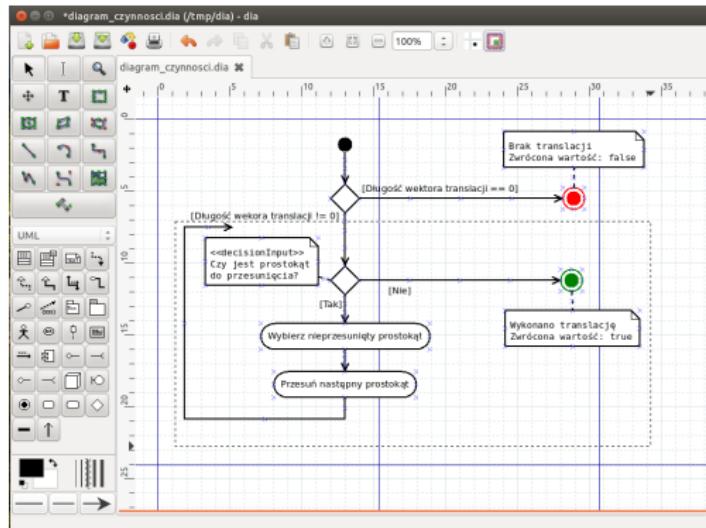
Należy skorzystać z podręcznego menu skojarzonego z tranzycją i dodać dodatkowy odcinek.

# Modyfikacja kształtu tranzycji



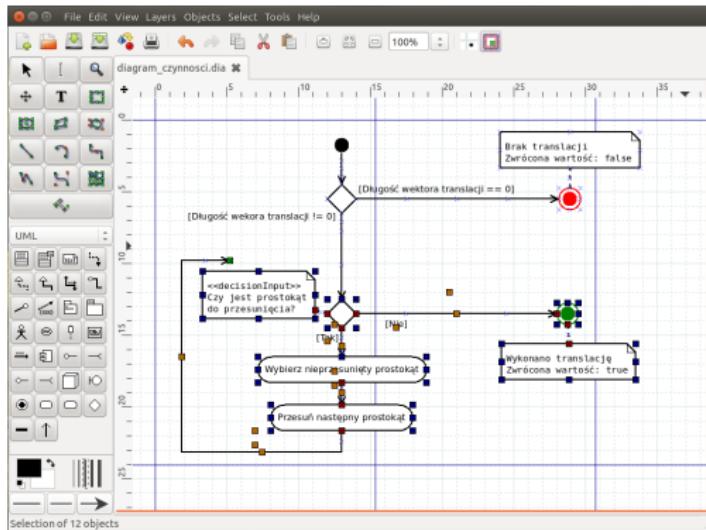
Aby otrzymać właściwy efekt należy uchwycić oznaczenie dodatkowego segmentu (czerwony kwadracik) i przeciągać linię w lewo lub prawo. Pojawia się dodatkowy problem, gdyż do każdego węzła wyboru może dochodzić tylko jedna tranzycja. Należy więc między istniejące dwa węzły wyboru wstawić węzeł łącznikowy. Będzie nim węzeł rozgałęzienia, gdyż pełni on dwie role. Może być użyty jako węzeł decyzyjny lub węzeł łącznikowy.

# Zbiorcza selekcja



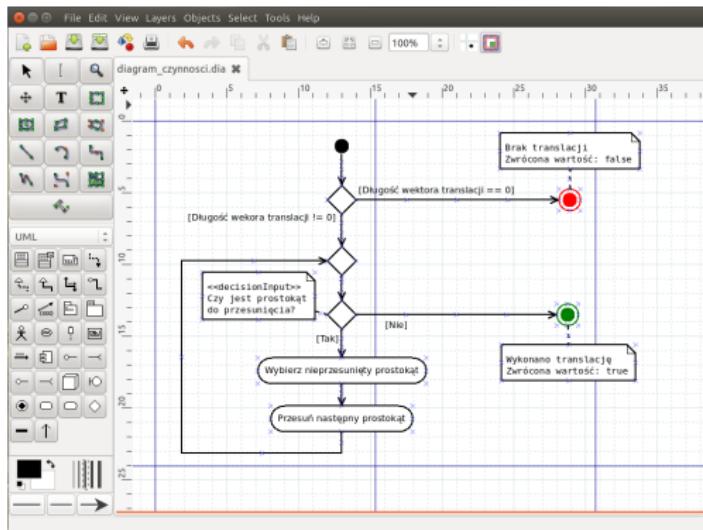
W tym celu wszystkie elementy selekcjonujemy poprzez zaznaczenie całego obszaru.

# Przesuwanie



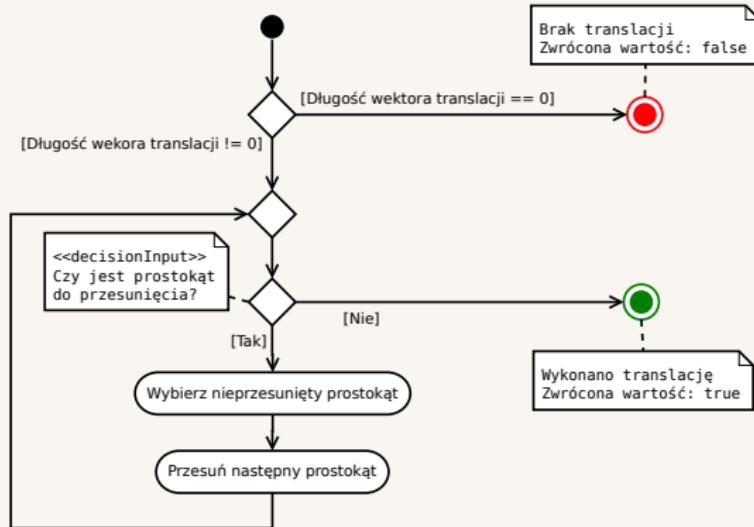
Następnie musimy *uchwycić* jeden z zaznaczonych elementów i przesuwając go przesunięte zostaną jednocześnie pozostałe elementy.

# Dodanie węzła łącznikowego



Wprowadzona modyfikacja kończy rysowanie tego diagramu.

# Wynik eksportu



Po wyeksportowaniu do pliku graficznego otrzymujemy końcowy rezultat przedstawiony powyżej.

Koniec prezentacji  
Dziękuję za uwagę