

Dokumentacja Techniczna Systemu Eksperckiego

Wspomagającego Wybór postaci do kampanii D&D.

Inżynieria Wiedzy i Systemy Ekspertowe

Sebastian Mrowiński

Informatyka III rok

Index:97727

Spis treści

[1. Ogólne założenia systemu 3](#_Toc155979274)

[2. Schemat działania 4](#_Toc155979275)

[Wybór Rasy: 4](#_Toc155979276)

[Wybór Klasy: 4](#_Toc155979277)

[Stworzenie Postaci: 4](#_Toc155979278)

[Podsumowanie Gotowej Postaci: 4](#_Toc155979279)

[Przykład wynikowego podsumowania: 4](#_Toc155979280)

[3. Właściwości aplikacji: źródła, fasety 5](#_Toc155979281)

[Baza Metafor: 6](#_Toc155979282)

[Baza Wyjaśnień CO TO JEST?: 6](#_Toc155979283)

[Ekspercka Baza Wiedzy: 6](#_Toc155979284)

[Fasety: 6](#_Toc155979285)

[Napotkane Trudności: 6](#_Toc155979286)

[4. Terminologia i stałe wartości 7](#_Toc155979287)

[SYSTEM: 7](#_Toc155979288)

[UŻYTKOWNIK: 7](#_Toc155979289)

[PYTANIA: 7](#_Toc155979290)

[Wartości Preferowane przez System: 7](#_Toc155979291)

[5. Materiały multimedialne i objaśnienia 8](#_Toc155979292)

[Grafika: 8](#_Toc155979293)

[Nowatorski Format Grafiki: 8](#_Toc155979294)

[Źródła Wizualne: 8](#_Toc155979295)

[Objaśnienia i metafory: 8](#_Toc155979296)

[Baza Wyjaśnień CO TO JEST?: 8](#_Toc155979297)

[Baza Metafor: 9](#_Toc155979298)

[6. Schemat działania — drzewa decyzyjne 10](#_Toc155979299)

[Baza Reguł źródeł wiedzy: 10](#_Toc155979300)

[Konkluzja postaci „Dragonborn Rouge” w programie CAKE: 12](#_Toc155979301)

[Drzewo Systemu tworzenia Postaci w D&D: 12](#_Toc155979302)

[7. DeTreex 14](#_Toc155979303)

[8. Implementacja systemu ekspertowego w języku Python (automatyczny) 16](#_Toc155979304)

[Biblioteki: 17](#_Toc155979305)

[Wczytanie Danych: 17](#_Toc155979306)

[Label Encoding: 17](#_Toc155979307)

[Budowa Drzewa Decyzyjnego: 17](#_Toc155979308)

[Interfejs Graficzny: 17](#_Toc155979309)

[Przewidywanie Postaci: 17](#_Toc155979310)

[Aktualizacja Opisów: 17](#_Toc155979311)

[Wynik działania programu: 18](#_Toc155979312)

[9. Implementacja systemu ekspertowego w języku Python (ręczny) 19](#_Toc155979313)

[Biblioteki i Interfejs Graficzny: 20](#_Toc155979314)

[Wczytywanie Opisów: 20](#_Toc155979315)

[Reguły Systemu Eksperckiego: 20](#_Toc155979316)

[Prezentacja Wyników: 20](#_Toc155979317)

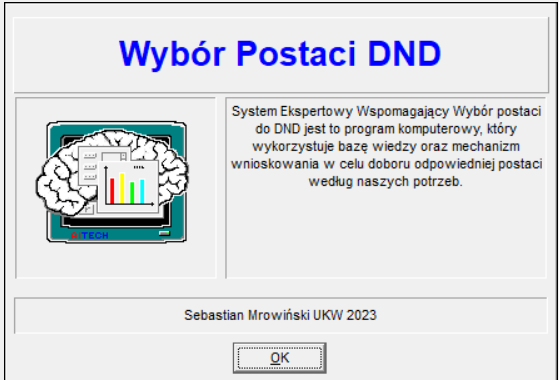
[Zakończenie Programu: 20](#_Toc155979318)

[Działanie Reguł: 21](#_Toc155979319)

[Interaktywność: 21](#_Toc155979320)

[10. Porównanie implementacji SPHINX i Python 21](#_Toc155979321)

# 1. Ogólne założenia systemu



Prosty i przyjazny dla użytkownika interfejs systemu eksperckiego, który opiera się na serii kilku pytań, został zaprojektowany w celu ułatwienia procesu tworzenia postaci dla graczy D&D. Główny nacisk kładziony jest na wybór rasy i klasy postaci - dwóch kluczowych aspektów postaci w uniwersum gry.

System wykorzystuje modele wnioskowania i reguły oparte na wyborach gracza, stylu gry i ogólnych oczekiwaniach wobec postaci, korzystając z bazy wiedzy odnoszącej się do uniwersum D&D. Celem tej metody jest dopasowanie gustów i wymagań użytkownika do odpowiednich ras i klas postaci.

Podobnie jak inne narzędzia tego typu, system ekspercki ma na celu ułatwienie graczom podejmowania decyzji, aby mogli szybko i skutecznie stworzyć postać, która będzie reprezentować ich wyobrażenie o tym, jak wygląda bohater w uniwersum D&D. Ponad to dodatkowe opisy I grafiki dodane do systemu eksperckiego pozwala na większe zrozumienie jakie gracz dokonuje podczas jego używania.

# 2. Schemat działania

## Wybór Rasy:

* + Gracz wybiera rasę spośród dostępnych opcji (np. Human, Elf, Dwarf, Dragonborn, Tiefling).
  + Na podstawie wyboru przedstawiany jest opis rasy.

## Wybór Klasy:

* + Gracz wybiera klasę postaci spośród dostępnych opcji (np. Fighter, Wizard, Rogue, Barbarian).
  + Na podstawie wyboru przedstawiany jest opis klasy.

## Stworzenie Postaci:

* + Na podstawie wybranych rasy i klasy wybierana jest najbardziej pasująca postać dla gracza.

## Podsumowanie Gotowej Postaci:

* + System prezentuje gotową postać, uwzględniając imię, rasę, klasę oraz krótki opis.

## Przykład wynikowego podsumowania:

Stworzona Postać!

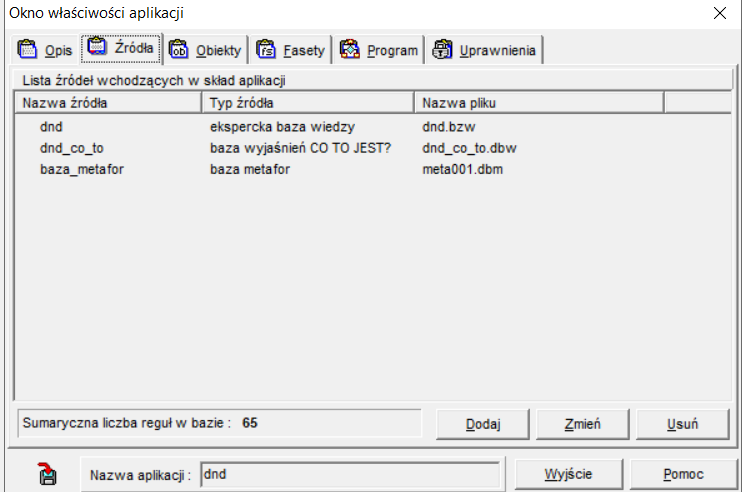
Imie: Roland Barns

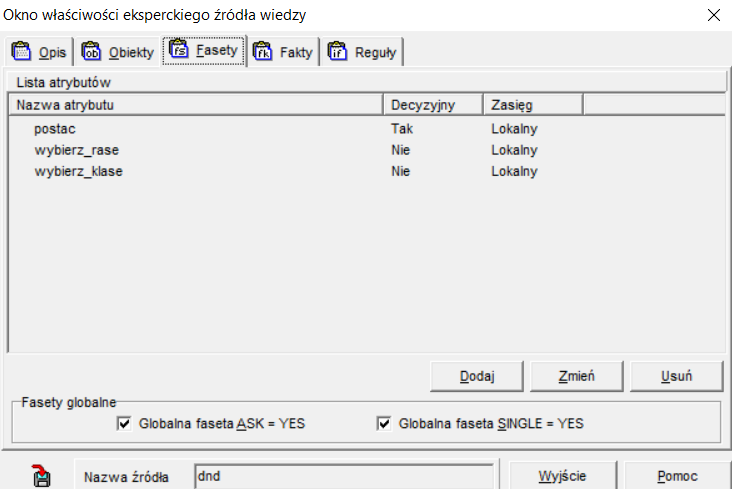
Rasa: Dragonborn

Klasa: Barbarian

Historia: Smoczy wojownik który opuścił swoje rodzinne miasto Nemfis w poszukiwaniu przygód.

# 3. Właściwości aplikacji: źródła, fasety





## Baza Metafor:

* + Stanowi fundament systemu, definiując zasady.

## Baza Wyjaśnień CO TO JEST?:

* + Pomaga użytkownikowi zrozumieć związki między wyborami, ułatwiając świadome podejmowanie decyzji podczas tworzenia postaci.

## Ekspercka Baza Wiedzy:

* + Obejmuje podstawową wiedzę ekspertów na temat tworzenia postaci w D&D.

## Fasety:

* + Opis: Zbiór kluczowych atrybutów w bazie wiedzy, umożliwiający precyzyjne dopasowanie postaci do preferencji gracza.

## Napotkane Trudności:

* + Problem techniczny związany z bazą metafor nie został rozwiązany. Aplikacja utrudnia dostęp do danych mimo istnienia pliku oraz nie pozwala na jego usunięcie w celach poprawy.

# 4. Terminologia i stałe wartości

## SYSTEM:

* + System ekspercki wspomagający tworzenia postaci w świecie D&D, uwzględniający preferencje gracza.

## UŻYTKOWNIK:

* + Osoba korzystająca z aplikacji do tworzenia postaci, dokonująca wyborów dotyczących rasy, klasy zgodnie z własnymi preferencjami.

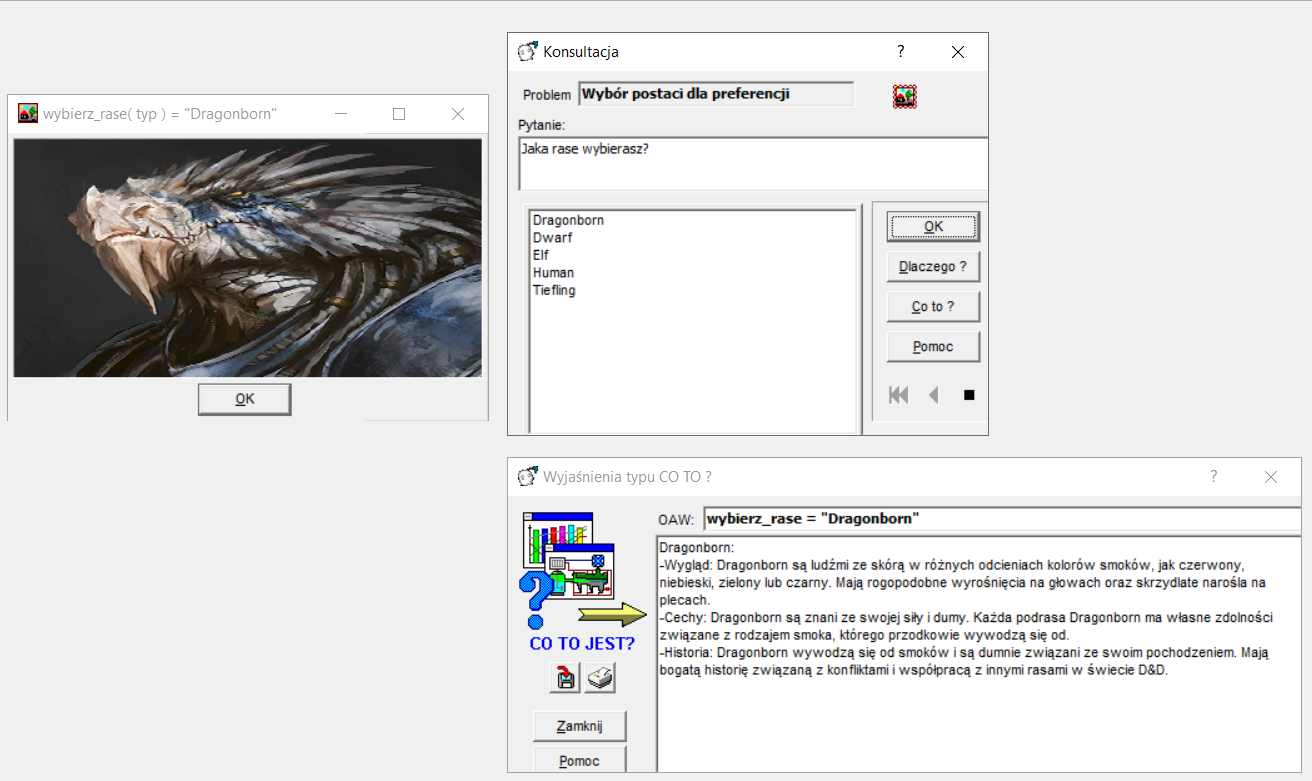
## PYTANIA:

* + Seria kilku pytań, które użytkownik musi odpowiedzieć w celu dostosowania postaci do swoich preferencji. Pytania obejmują wybór rasy oraz klasy.

## Wartości Preferowane przez System:

* + *Jaka rase wybierasz:* Dragonborn, Dwarf, Elf, Human, Tiefling.
  + *Jaki typ postaci Cie interesuje:* Furia i Wściekłość, Mistrz Miecza, Sztuki Walki i Duchowa Energia, Straznik Swiatla, Infiltracja i Obrazenia krytyczne, Wsparcie i Leczenie, Inzynieria magiczna, Wiedza i Arkana Magi, Narodzony z Magia, Muzyka i Magia, Zrecznosc i Bron dystansowa, Magia paktow i Cienie, Przywiazanie do natury.

# 5. Materiały multimedialne i objaśnienia



## Grafika:

* + Aplikacja została wzbogacona o elementy wizualne, takie jak zdjęcia, aby zwiększyć zrozumienie i atrakcyjność pytań dotyczących tworzenia postaci.

## Nowatorski Format Grafiki:

* + Pliki graficzne zostały przekonwertowane na format BMP, co pozwala na efektywne ich wykorzystanie w systemie.

## Źródła Wizualne:

* + Materiały graficzne pochodzą głównie z Google Images, ponieważ posiada największą różnorodność dzięki czemu elementy wizualne są spójne i różnorodne.

## Objaśnienia i metafory:

W moim programie do tworzenia postaci do D&D, objaśnienia i metafory są kluczowym elementem, który pomaga użytkownikom zrozumieć, dlaczego system podejmuje konkretne decyzje dotyczące postaci.

## Baza Wyjaśnień CO TO JEST?:

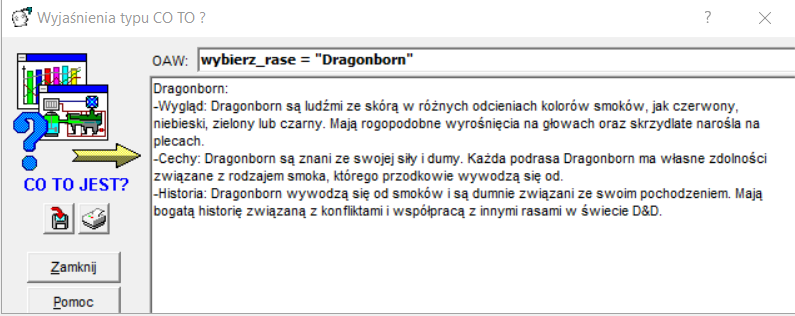
* Wyjaśnienia tłumaczą, dlaczego system wybrał daną postać na podstawie udzielonych przez użytkownika odpowiedzi.

## Baza Metafor:

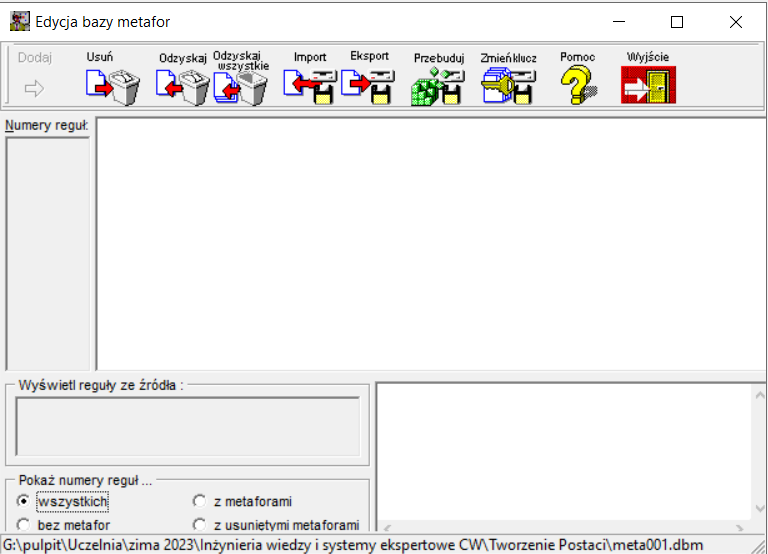
* Metafory opisują reguły systemu w bardziej zrozumiały sposób, pomagając użytkownikowi lepiej zrozumieć, jakie zasady kierują procesem tworzenia postaci.

Te dwie bazy wiedzy działają razem, aby umożliwić użytkownikom pełniejsze zrozumienie decyzji systemu. Po zakończeniu interakcji z aplikacją, użytkownicy mogą szczegółowo przeanalizować, jakie reguły i metafory wpłynęły na stworzenie konkretnej postaci w świecie D&D.

Przykład wyjaśnienia - CO TO?

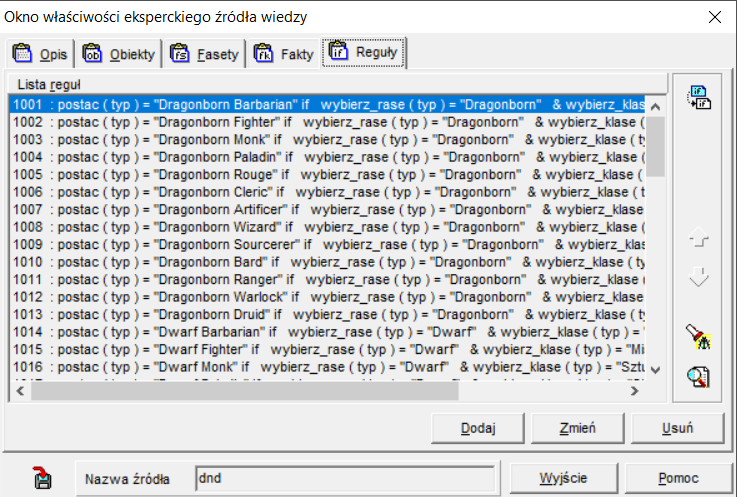
****

Przykład wyjaśnienia – baza metafor



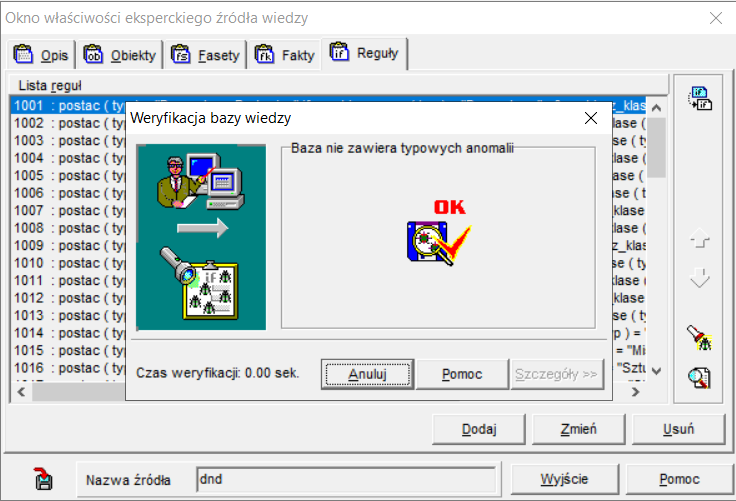
Niestety baza metafor ma problemy przez które nie byłem w stanie przejść.

# 6. Schemat działania — drzewa decyzyjne

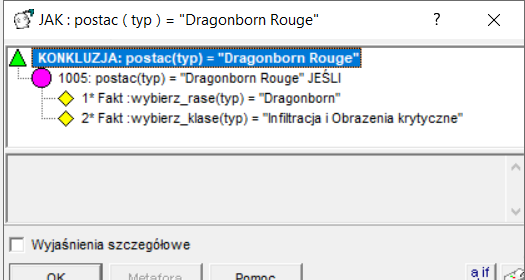
Drzewa decyzyjne to jeden ze sposobów na wizualne przedstawienie wspomagania decyzji w systemie.

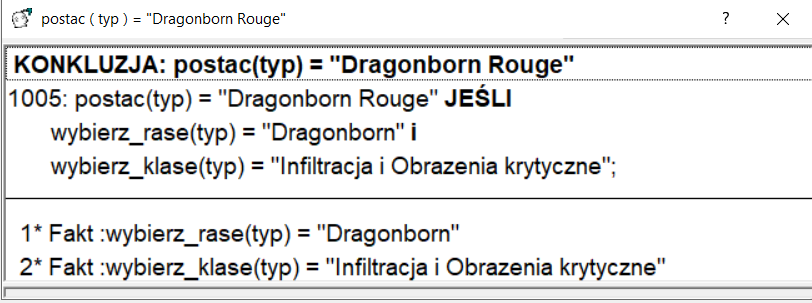
## Baza Reguł źródeł wiedzy:

W systemie eksperckim do tworzenia postaci w D&D, Baza Reguł stanowi fundament zaimplementowany zgodnie z przemyślanym drzewem decyzyjnym. Reguły są opisane przy użyciu trójki OAW (Obiekt-Atrybut-Wartość), co jest podstawową strukturą reprezentacji wiedzy w tym systemie. Każdy powiązany atrybut tworzy fasetę, która precyzyjnie definiuje cechy postaci.



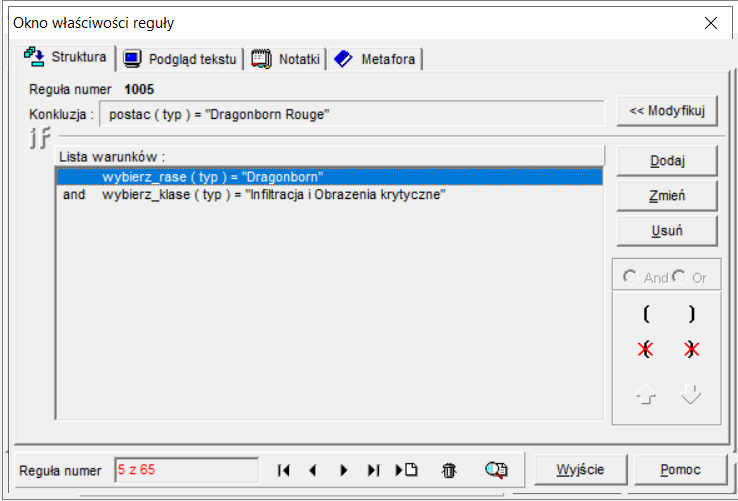
Baza wiedzy została pomyślnie zweryfikowana. Wszystko zostało wprowadzone poprawnie.





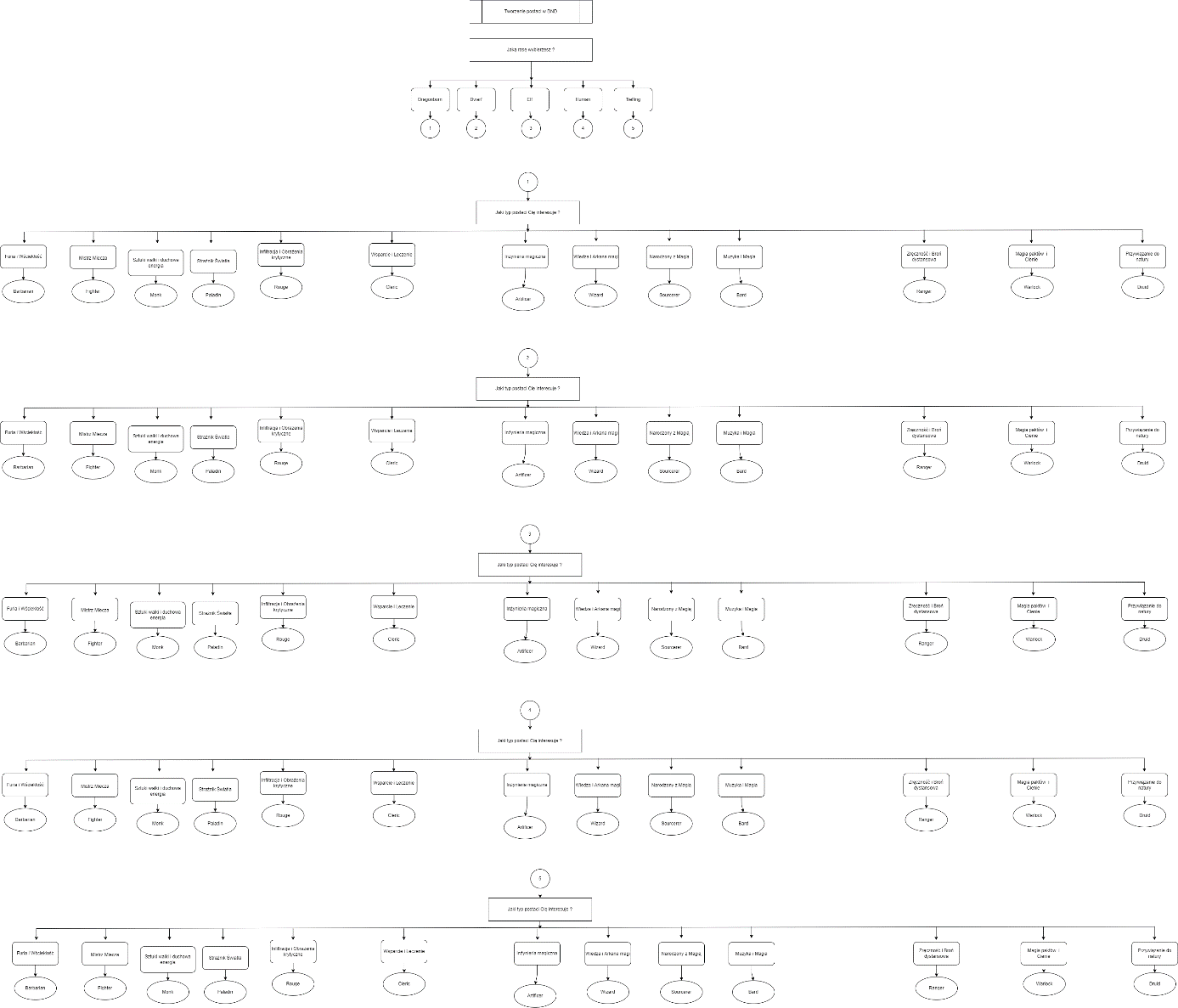
Wnioskowanie w systemie generuje konkluzje, czyli rezultaty decyzji. Te konkluzje przedstawiają schemat instrukcji warunkowych, które kierują procesem tworzenia postaci. Dzięki temu, użytkownicy mogą zobaczyć, jakie reguły i atrybuty wpływają na ostateczny wybór postaci.

## Konkluzja postaci „Dragonborn Rouge” w programie CAKE:



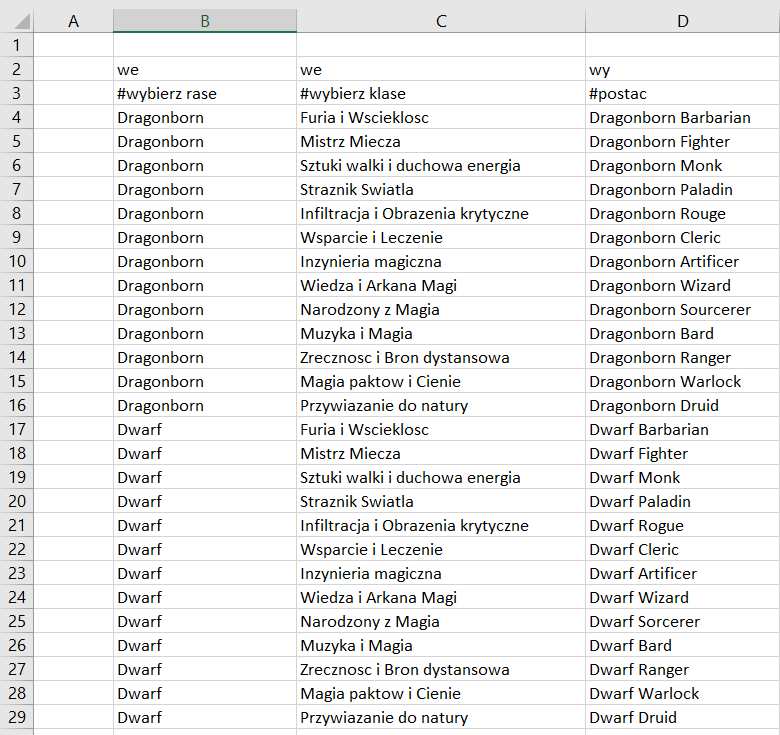
## Drzewo Systemu tworzenia Postaci w D&D:

W systemie eksperckim do tworzenia postaci w D&D, drzewo systemu wyboru zostało rozbite na mniejsze segmenty, co pozwala na lepsze zrozumienie jego założeń. Ten schemat decyzyjny został stworzony przy użyciu narzędzia na stronie draw.io, co pozwoliło na precyzyjne zobrazowanie kroków wyboru postaci w systemie. Każdy fragment drzewa reprezentuje konkretne pytania i decyzje, które użytkownik podejmuje, prowadząc do ostatecznego stworzenia postaci w świecie D&D.

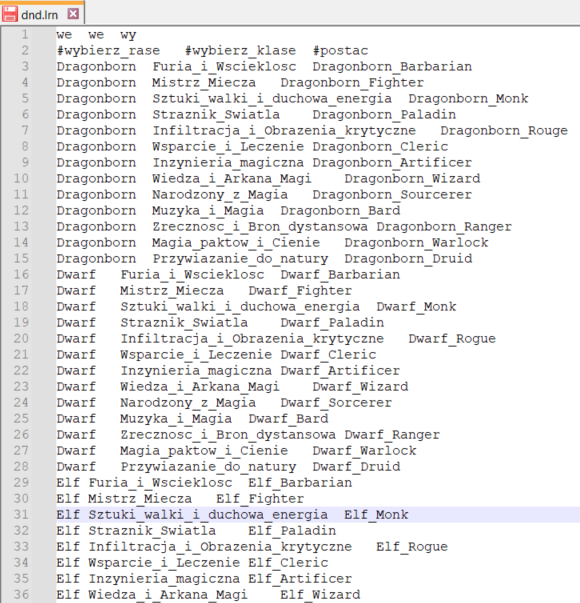


# 7. DeTreex

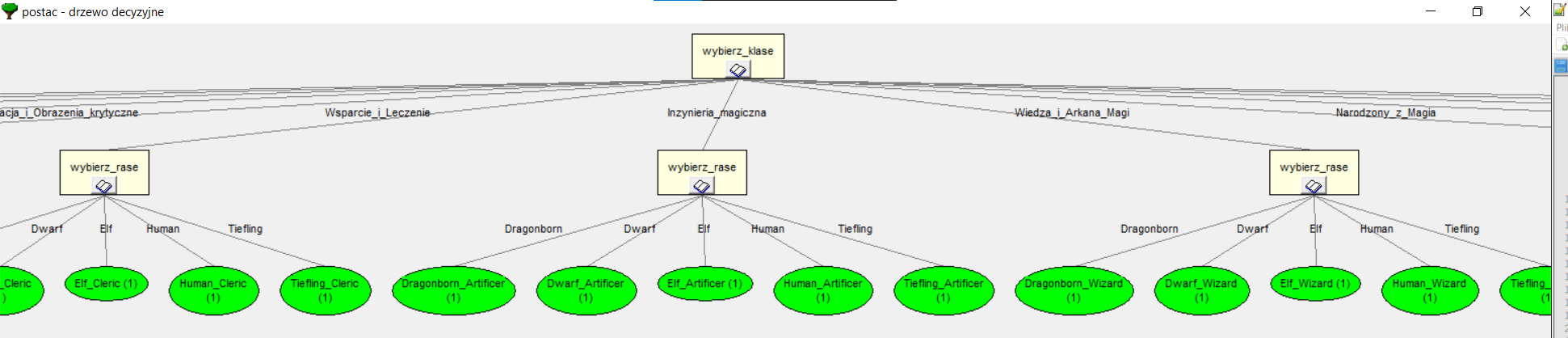
**Plik uczący w Excelu**



**Plik uczący Irn**



**Drzewo wygenerowane przez program:**



# 8. Implementacja systemu ekspertowego w języku Python (automatyczny)



Powyższy kod stanowi implementację systemu eksperckiego w języku Python, służącego do prognozowania postaci w świecie D&D na podstawie wybranych rasy i klasy. Poniżej przedstawiono kluczowe elementy oraz działanie kodu:

## Biblioteki:

* + Użyto bibliotek takich jak **pandas** do manipulacji danymi, **DecisionTreeClassifier** z **scikit-learn** do budowy drzewa decyzyjnego oraz **tkinter** do stworzenia interfejsu graficznego.

## Wczytanie Danych:

* + Dane postaci są wczytywane z pliku CSV o nazwie "csv.csv" z wykorzystaniem biblioteki **pandas**.

## Label Encoding:

* + Wykorzystano **LabelEncoder** z **scikit-learn** do zakodowania etykiet (rasy i klasy postaci) na liczby całkowite, co jest wymagane do zastosowania drzewa decyzyjnego.

## Budowa Drzewa Decyzyjnego:

* + Stworzono drzewo decyzyjne (**DecisionTreeClassifier**), które przewiduje postać na podstawie wybranych rasy i klasy.

## Interfejs Graficzny:

* + Stworzono prosty interfejs graficzny przy użyciu **tkinter** z elementami takimi jak etykiety, pola wyboru (**Combobox**), przyciski oraz pola wyświetlające wyniki.

## Przewidywanie Postaci:

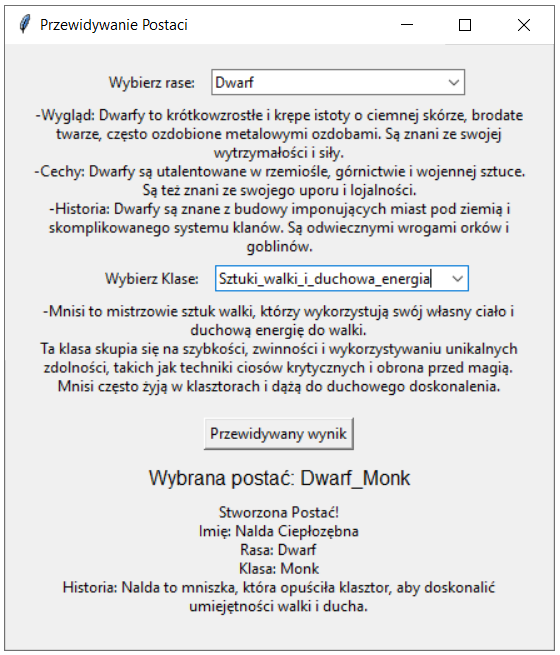
* + Funkcja **predict\_character** korzysta z drzewa decyzyjnego do prognozowania postaci na podstawie wybranych rasy i klasy. Wyniki są prezentowane w etykietach na interfejsie.

## Aktualizacja Opisów:

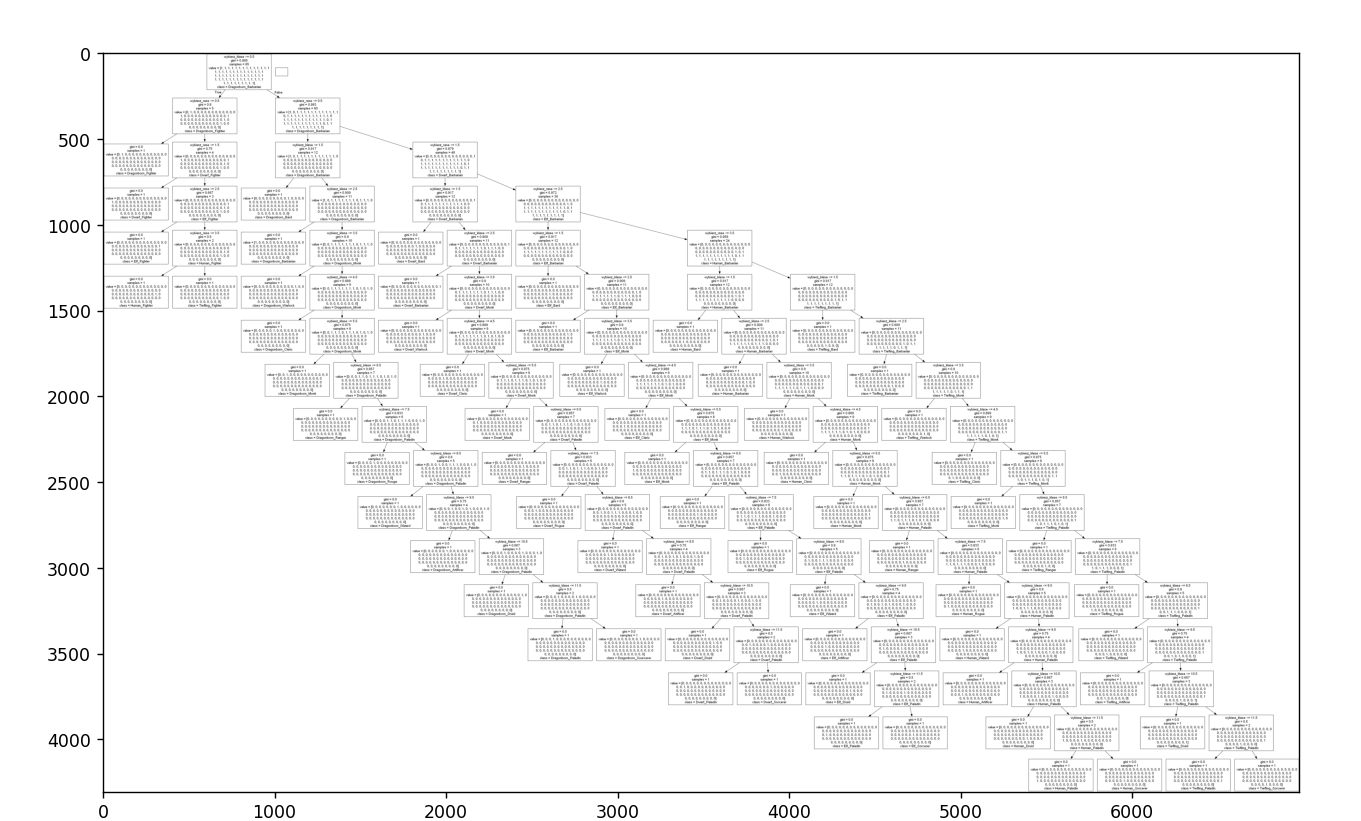
* + Dodano funkcje aktualizujące opisy ras i klas w odpowiedzi na wybór użytkownika.

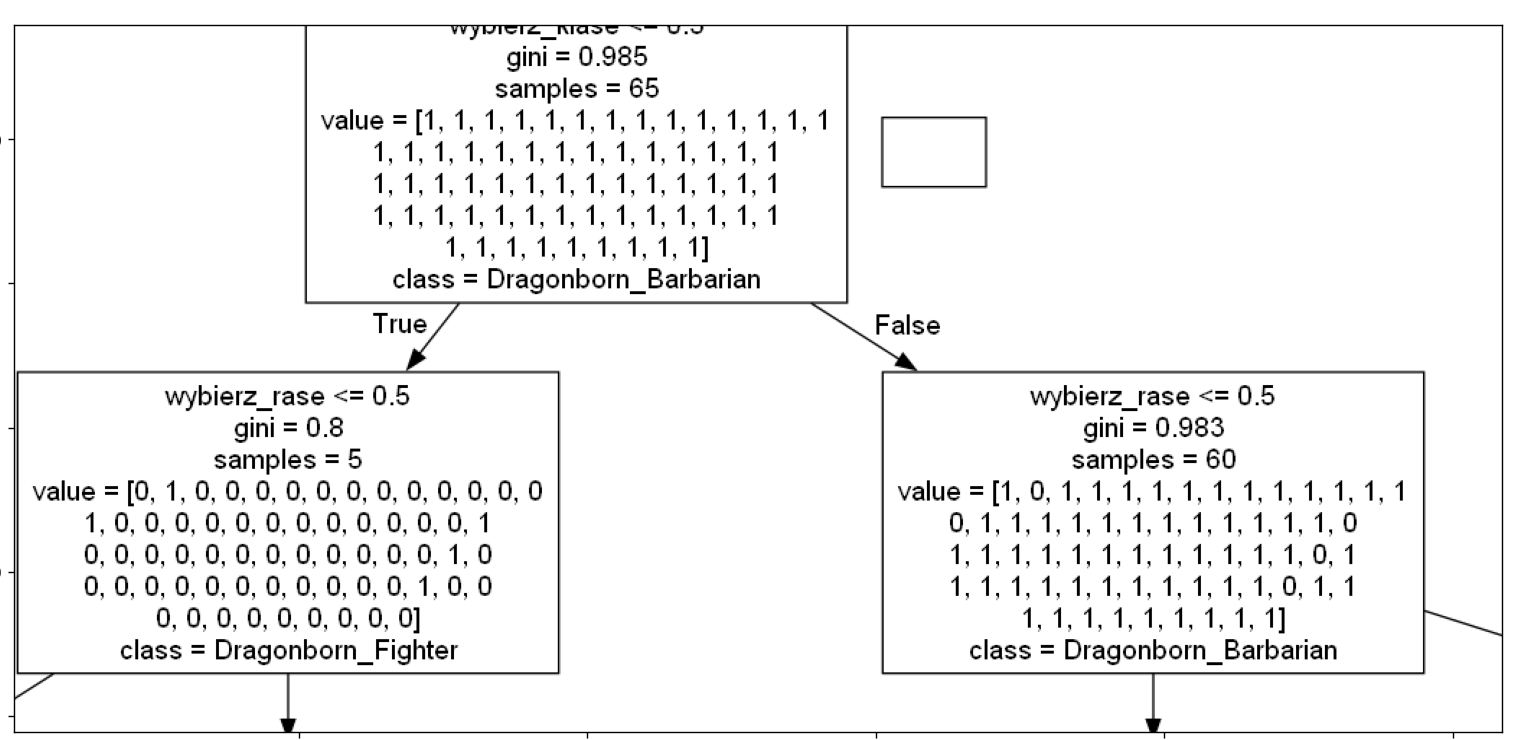
Ten kod umożliwia interaktywne tworzenie postaci w świecie D&D, a interfejs graficzny ułatwia użytkownikom wybór rasy i klasy, prezentując jednocześnie prognozowaną postać w opisie.

## Wynik działania programu:



Wygląd mydecisiontree.





# 9. Implementacja systemu ekspertowego w języku Python (ręczny)



Powyższy kod to ręczna implementacja systemu eksperckiego w języku Python z wykorzystaniem biblioteki PyQt do prognozowania postaci w świecie D&D na podstawie wybranych rasy i klasy. Kod został skrócony aby umożliwić wstawienie go do dokumentacji jego pełna wersja znajduje się w plikach. Poniżej przedstawiono kluczowe elementy oraz działanie kodu:

## Biblioteki i Interfejs Graficzny:

* + Kod wykorzystuje bibliotekę PyQt5 do stworzenia interfejsu graficznego, umożliwiającego użytkownikowi wybór rasy i klasy postaci w grze D&D. Widżety, takie jak etykiety, pola wyboru (**QComboBox**), przyciski oraz pola wyświetlające wyniki, są używane do zbudowania intuicyjnego interfejsu.

## Wczytywanie Opisów:

* + Funkcja **load\_descriptions** odpowiedzialna jest za wczytywanie opisów ras i klas postaci z plików tekstowych. To pozwala na dynamiczne aktualizowanie interfejsu z nowymi danymi.

## Reguły Systemu Eksperckiego:

* + Implementacja reguł systemu eksperckiego oparta jest na bibliotece **durable\_rules**. Reguły te definiują związki między wybraną rasą a klasą postaci. Każda reguła zawiera opis postaci oraz dodatkowe informacje, co pozwala na elastyczne dodawanie nowych postaci do systemu.

## Prezentacja Wyników:

* + Po dokonaniu wyboru rasy i klasy przez użytkownika, system ekspercki przewiduje postać na podstawie zdefiniowanych reguł. Opis wybranej postaci prezentowany jest w interfejsie użytkownika, co umożliwia szybkie zrozumienie, kim jest stworzona postać.

## Zakończenie Programu:

* + Interfejs zawiera przycisk "Zakończ", który umożliwia użytkownikowi zakończenie programu. Ta funkcjonalność pozwala na wygodne zamykanie aplikacji po utworzeniu postaci.

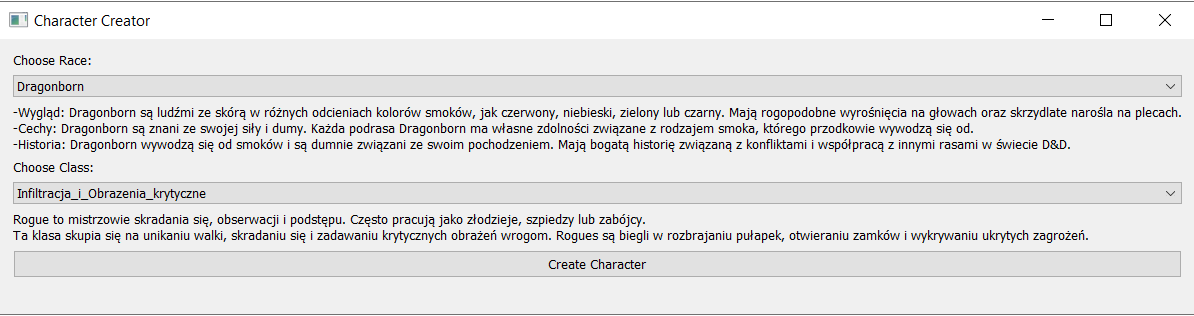
## Działanie Reguł:

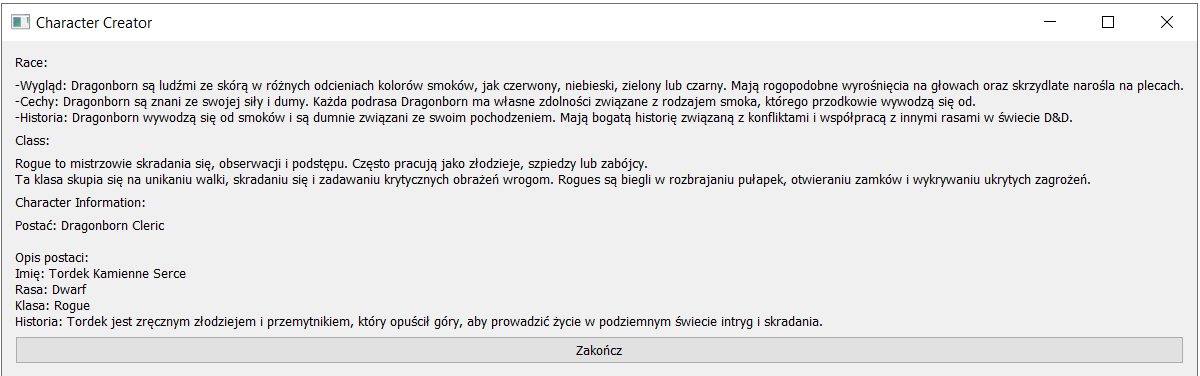
* + System ekspercki działa na zasadzie wnioskowania opartego na regułach. Po dokonaniu wyboru przez użytkownika, system przegląda zdefiniowane reguły, aby znaleźć odpowiednią postać. Wynik wnioskowania prezentowany jest w czytelny sposób na interfejsie.

## Interaktywność:

* + Kod uwzględnia interakcję z użytkownikiem poprzez obsługę zdarzeń, takich jak wybór rasy i klasy. Działa to jak interaktywny kreator postaci, który dostarcza informacji o każdej postaci oraz jej historii.

## Wynik działania programu:





# 10. Porównanie implementacji SPHINX i Python

Według mojej oceny, Python jest znacznie lepszym wyborem w porównaniu z Sphinxem. Python oferuje prostotę, czytelność kodu i ogromną społeczność dzięki czemu jest łatwiejszy do nauki. Jest bardziej wszechstronny co sprawdza się w większych projektach. Ponad to jest popularny i posiada wiele bibliotek ułatwiających pracę przy implementacji naszego rozwiązania.