

Prolog. Obliczenia. Struktury. Listy.

Obliczenia w prologu

Uruchom program **przewodnicy.pl**. Przeanalizuj program. Odpowiedz na pytania:

- Kto przewodził w roku 1972
- Ile lat przewodził gierek

Sformułuj zapytania, a następnie zapisz je w postaci reguł.

- Ile lat minęło od końca przewodzenia sekretarza X do początku sekretarza Y.
- Po ile lat przewodzili poszczególni przywódcy
- Kto przewodził dłużej niż 5 lat/ dowolna liczba lat
- Kto zakończył przewodzenie przed rokiem 1980/ dowolnym rokiem
- Kto był następcą gomułki. /dowolnego przywódcy

Listy.

Listy w prologu zapisujemy w postaci [1, 2, 4, 7]. Oznaczenie [Head|Tail] podobnie jak w języku ML i Haskell oznacza listę z głową Head i Ogonem Tail.

- Uruchom program **listy.pl**. Przeanalizuj. Odpowiedz na pytania.
- Czy element 2 jest na liście [1, 2, 5, 6, 9]
- Wymień po kolei elementy listy [1, 2, 5, 6, 9] (*predykat member*)
- Czy [ala, ola, jola] jest listą?
- Jaka jest długość listy imion męskich (term imiona_meskie(..)).
- Ile jest elementów na liście imiona_damskie.
- Czy imię danka jest na liście imion damskich. Czy jest na niej maria.
- Połącz obie listy imion.
- Zdefiniuj regułę ostatni (X, L), gdzie X jest tym ostatnim elementem listy L. Predykat powinien podać ostatni element listy L).
- Zdefiniuj regułę wydobywającą z listy jej pierwszy element.
- Jaki jest ostatni element listy imion męskich
- Zdefiniuj regułę dominuje(Lista1, Lista2), który bada, czy każdy element listy1 jest większy od odpowiedniego elementu listy2
- Obejrzyj regułę liczba_naturalna(X). Sprawdź, co ona robi. Jakie pytanie zadać, aby uzyskać liczby naturalne mniejsze niż 10.?

Struktury.

Uruchom w prologu program **rower.pl**. Przeanalizuj. Odpowiedz na pytania.

- Z jakich podzespołów składa się rower; koło
- Z jakich nierozkładanych części składa się koło
- W jakich podzespółach występuje koło; nakretka
- Czy nakretka jest częścią roweru
- Z ilu części składa się koło
- W jakim podzespole występuje koło/ W jakim podzespole na pierwszym miejscu wymieniono koło)
- Sformułuj regułę jest_czescia(Czesc, Podzespól), która powie, czy dana Część należy do Podzespołu

Nawracanie i odcięcie – zasady przeprowadzania obliczeń

Przeanalizuj program **biblioteka.pl**. Sformułuj pytania.

- Jakie usługi są dostępne dla czytelnika Bolek?
- Jakie usługi są dostępne dla czytelnika Lolek?
- Jakie usługi są dostępne dla czytelnika Józek?

Prześledź działanie pytania o usługi odpowiednio dla Bolka i Tolka. Użyj predykatu **trace**. (wyjście z trybu śledzenia – `notrace.`, `nodebug.`)

Które predykaty są sprawdzane w poszczególnych przypadkach? W szczególności, które z predykatów opisujących dostępność usług.

Usuń symbol odcięcia - ! (wykrzyknik) z pierwszego predykatu `usluga`, następnie ponownie wykonaj pytania o usługi dla Bolka i Tolka. Co się zmienia? Jaką zatem rolę pełni predykat ! (wykrzyknik)?

Ciekawostki

Uruchom program **labirynt.pl**. Termy $d(X, Y)$ mówią o istnieniu drzwi z pokoju X do pokoju Y . Drzwi działają w obie strony. Narysuj sobie rozkład pokoi labiryntu.

- Sprawdź, czy przy aktualnym układzie pokoi da się dotrzeć do skarbu zakładając, że wejście jest przez pokój a .
- Czy jest tylko jedna droga prowadząca do skarbu?
- Sprawdź, czy istnieje przejście między dwoma wybranymi przez Ciebie pokojami.
- Prześledź w trybie **trace** w jaki sposób prolog wybiera kolejne pokoje do przejścia.
- Jeśli chcesz wyświetlić pokoje, przez które przechodzisz, w predykatach opisujących przejście dołóż predykat `write(X)`.