

ITERACJA A REKURENCJA

ARKADIUSZ ADAMCZYK
PAWEŁ CZAPCZYŃSKI
UTH RADOM

ITERACJA

- Iteracja - czynność powtarzania tej samej operacji w pętli z góry określoną liczbę razy lub aż do spełnienia określonego warunku. Mianem iteracji określa się także operacje wykonywane wewnątrz takiej pętli.

REKURENCJA

- Rekurencja - odwołanie się np. funkcji do samej siebie. Dokładna ilość tych odwołań nie ma zupełnie znaczenia, ponieważ już sam fakt wywołania jest podstawą do określenia funkcji jako rekurencyjnej.

RÓŻNICE

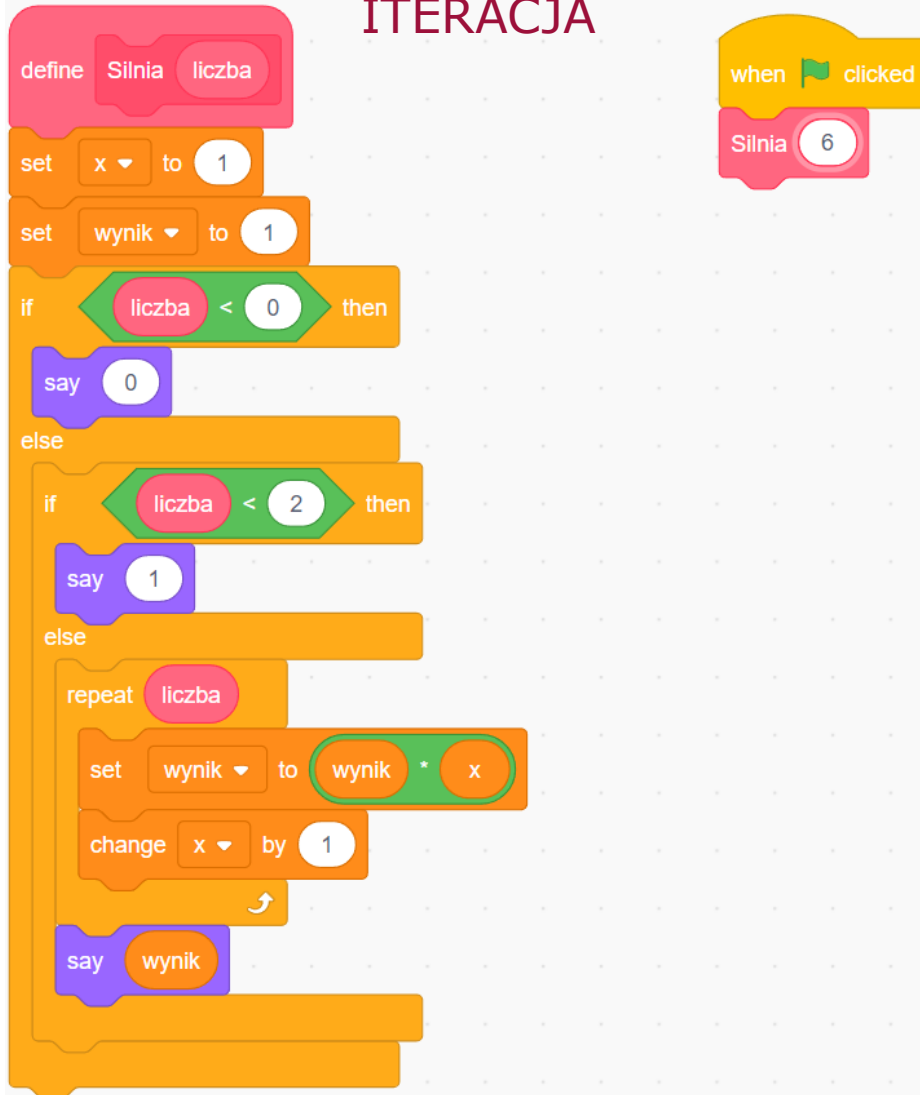
- Rekurencja występuje wtedy, gdy funkcja w programie wielokrotnie wywołuje samą siebie, podczas gdy iteracja ma miejsce, gdy zestaw instrukcji w programie jest wielokrotnie wykonywany (np. pętla).
- Metoda rekurencyjna zawiera zestaw instrukcji, wywołanie samej instrukcji i warunek zakończenia, podczas gdy instrukcje iteracji zawierają inicjalizację, inkrementację, warunek, zestaw instrukcji w pętli i zmienną sterującą.
- Zdanie warunkowe decyduje o zakończeniu rekurencji i wartości zmiennej sterującej decydują o zakończeniu instrukcji iteracyjnej.
- Jeśli metoda nie doprowadzi do stanu zakończenia, wchodzi w nieskończoną rekurencję. Z drugiej strony, jeśli zmienna sterująca nigdy nie prowadzi do wartości zakończenia, instrukcja iteracji iteruje w nieskończoność.
- Nieskończona rekursja może doprowadzić do awarii systemu, podczas gdy nieskończona iteracja pochłania cykle procesora (stack overflow).

RÓŻNICE

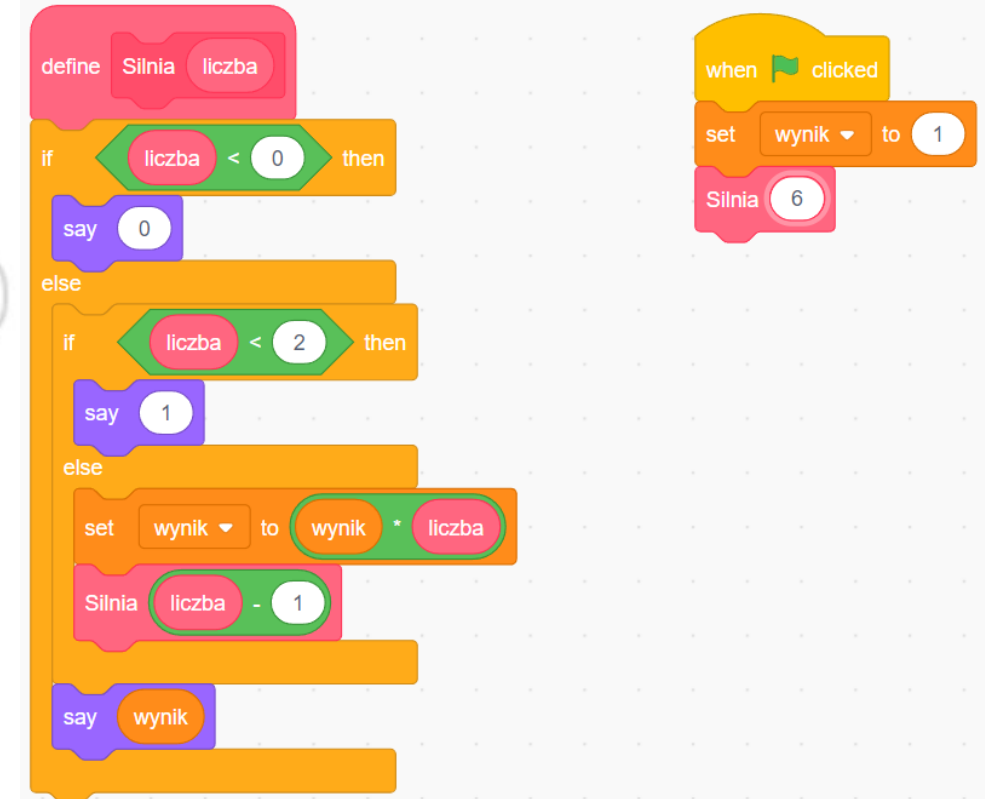
- Rekurencja jest zawsze stosowana do funkcji, podczas gdy iteracja jest stosowana do zbioru instrukcji.
- Zmienne utworzone podczas rekurencji są przechowywane na stosie, podczas gdy iteracja nie wymaga stosu.
- Rekurencja powoduje narzut wielokrotnego wywoływania funkcji, podczas gdy iteracja nie ma funkcji wywołującej narzut.
- Ze względu na wykonywanie funkcji wywoływanie rekurencji jest wolniejsze, natomiast wykonywanie iteracji jest szybsze.
- Rekurencja zmniejsza rozmiar kodu, a iteracje wydłużają kod.

SILNIA

ITERACJA

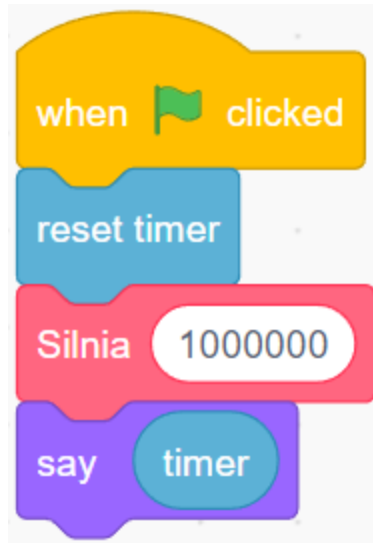


REKURENCJA

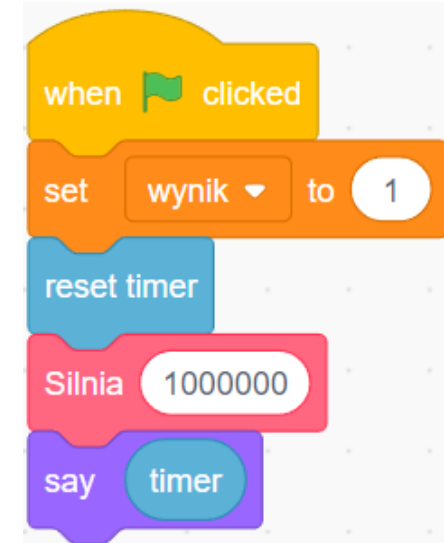


SILNIA - BENCHMARK

ITERACJA



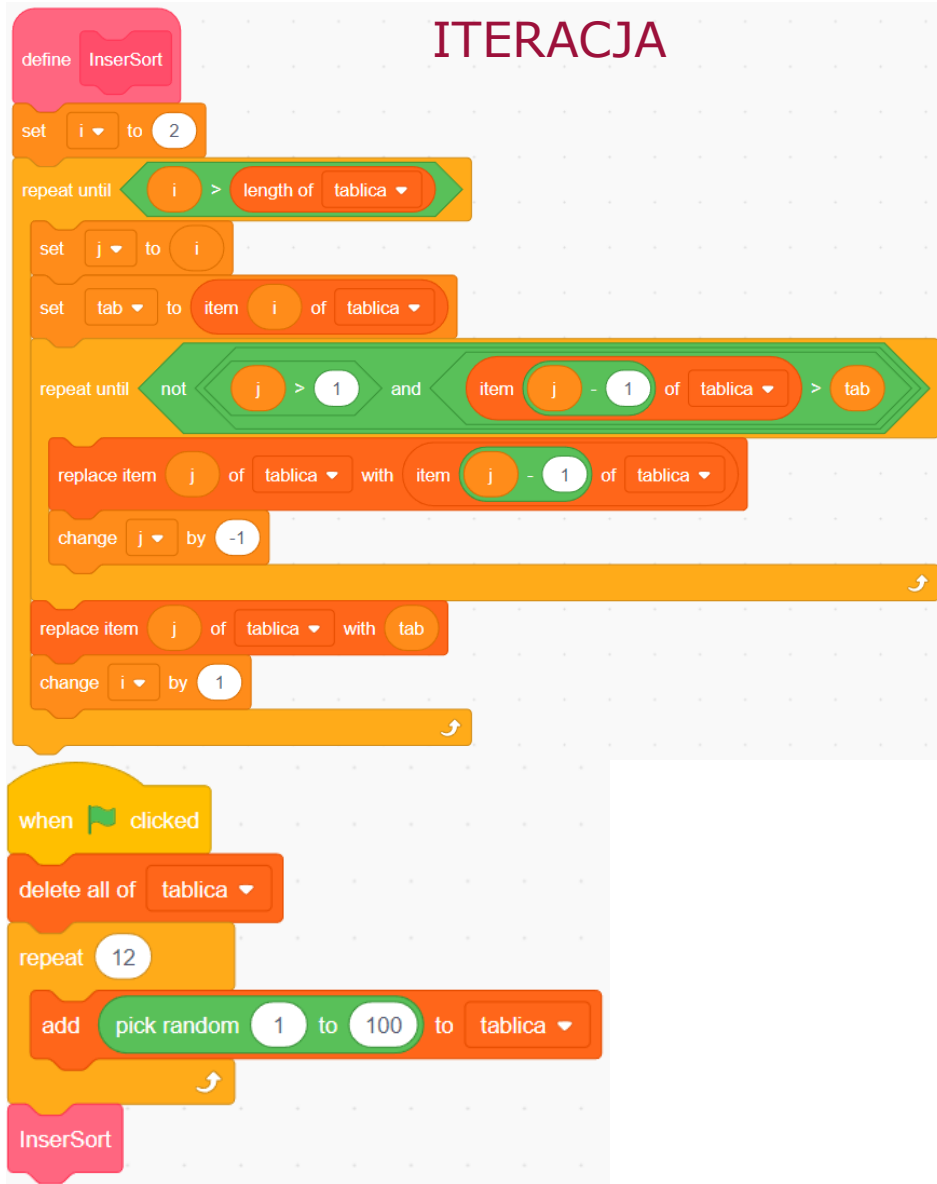
REKURENCJA



| ITERACJA [s] | REKURENCJA [s] |
|--------------|----------------|
| 1.08 | 2.59 |
| 1.07 | 2.78 |
| 1.09 | 2.53 |
| 1.09 | 2.50 |

INSERT SORT

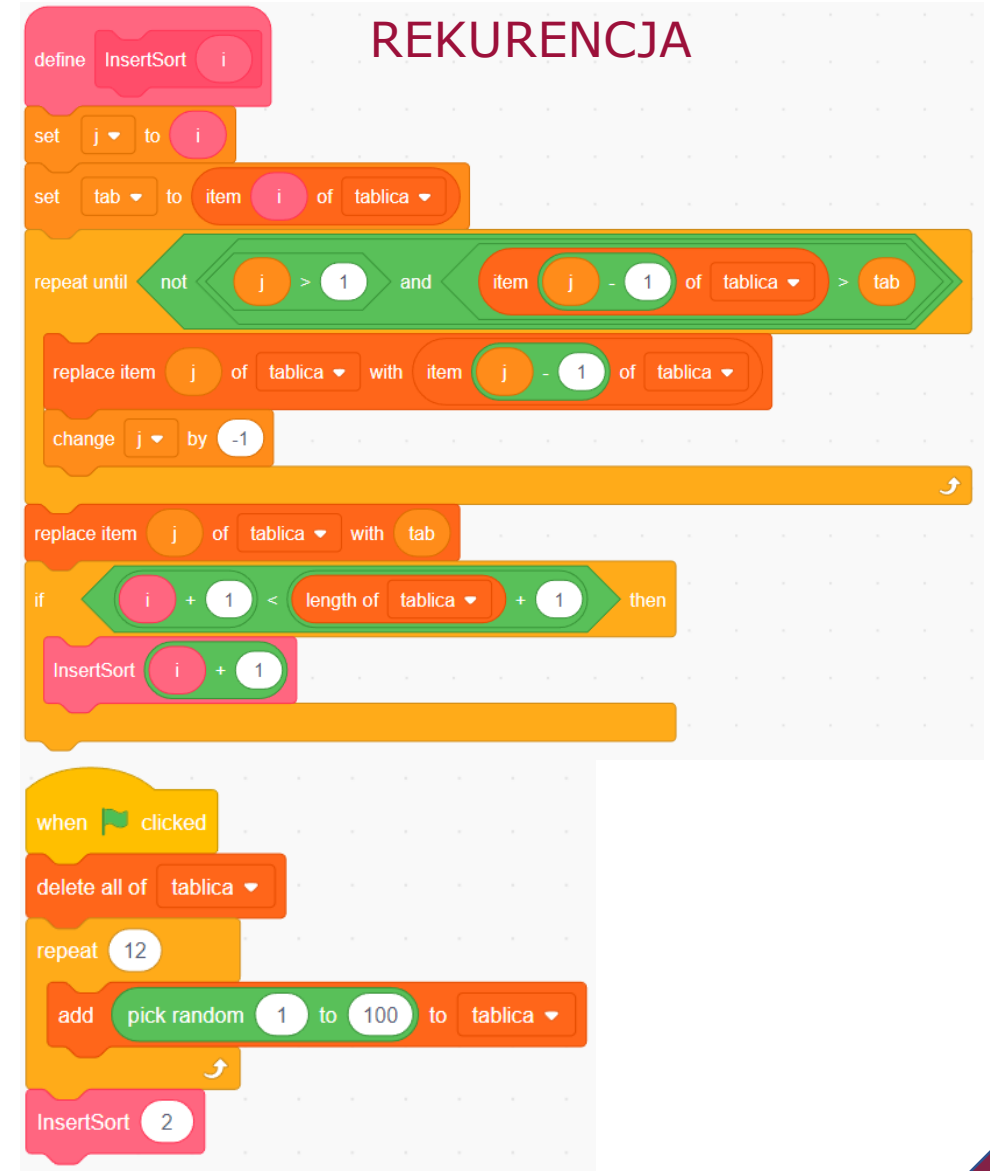
ITERACJA



| tablica | |
|---------------|----|
| 1 | 92 |
| 2 | 99 |
| 3 | 68 |
| 4 | 9 |
| 5 | 62 |
| 6 | 67 |
| 7 | 99 |
| 8 | 4 |
| 9 | 97 |
| 10 | 93 |
| 11 | 33 |
| 12 | 59 |
| + length 12 = | |

| tablica | |
|---------------|----|
| 1 | 4 |
| 2 | 9 |
| 3 | 33 |
| 4 | 59 |
| 5 | 62 |
| 6 | 67 |
| 7 | 68 |
| 8 | 92 |
| 9 | 93 |
| 10 | 97 |
| 11 | 99 |
| 12 | 99 |
| + length 12 = | |

REKURENCJA



INSERT SORT - BENCHMARK



| ITERACJA [s] | REKURENCJA [s] |
|--------------|----------------|
| 176.48 | 243.90 |
| 54.32 | 62.17 |
| 54.03 | 58.67 |
| 100.49 | 57.05 |

PODSUMOWANIE

- Funkcja rekurencyjna jest łatwiejsza do napisania, ale nie zachowuje się dobrze w porównaniu do iteracji, podczas gdy iteracja jest trudniejsza do napisania, ale jej wydajność jest o wiele lepsza w porównaniu z rekurencją.

DZIĘKUJEMY ZA UWAGĘ

ARKADIUSZ ADAMCZYK
PAWEŁ CZAPCZYŃSKI
UTH RADOM