

## LISTA 2: WSKAŹNIKI I BIBLIOTEKA SDL

Zrób jedno spośród następujących zadań. Punktacja: 4 + max. 2 bonusowe.

**Ćwiczenie 1.** Zaprogramuj typ multizbiorów jako jednokierunkową listę wiążaną (ang. „linked”) uporządkowaną, której węzły zawierają (oprócz wskaźnika na kolejny element) element oraz liczbę jego wystąpień w zbiorze. Nie przechowuj w liście elementów z „zero” wystąpień. Jako typ elementów weź dowolny wspierający wbudowane operatory porównania jak <, =, np. `String`.

Napisz procedury liczące w czasie liniowym:

1. sumę zbiorów,
2. przekrój zbiorów,
3. różnicę zbiorów,
4. dodanie elementu do zbioru,
5. usuwanie elementu ze zbioru.

W programie głównym napisz testy. Przy poprawnym rozwiązaniu przysługuje punkt-za-punkt, m.in. za zrobienie wszystkiego gwarantowany punkt bonusowy.

Zadania dla biblioteki SDL są zainspirowane przez slajdy do *Introduction to Programming in Java* Roberta Sedgewicka i Kevina Wayne.

**Ćwiczenie 2.** Zaprogramuj animację dwóch odbijających się piłek o różnych promieniach i odpowiednio masach, ze zderzeniami sprężystymi, w jednorodnym polu grawitacyjnym. Użyj modułu `matrix` do operacji na wektorach położenia i prędkości. Za pełną funkcjonalność gwarantowany punkt bonusowy, 4 punkty przy braku grawitacji lub uproszczonym modelu zderzeń.

Zderzenia sprężyste:

[en.wikipedia.org/wiki/Elastic\\_collision#Two-dimensional\\_C.23\\_example](http://en.wikipedia.org/wiki/Elastic_collision#Two-dimensional_C.23_example)

Drugi punkt bonu

**Ćwiczenie 3.** Na bazie programu `ChaosGameSDL`, napisz program pytający się użytkownika o ilość macierzy-funkcji w IFS, tworzący je dynamicznie, wypełniając je losowymi liczbami – zapewnij że powstałe odwzorowania są kontrakcjami. (Formalnie, że norma operatorowa części liniowej – tzn. bez ostatniego wiersza i kolumny – macierzy jest mniejsza od 1.) Wektor prawdopodobieństw również wypełnia losowo. Program następnie w pętli oferuje użytkownikowi następujące akcje:

1. zapisanie narysowanego fraktala do pliku graficznego,
2. wygenerowanie nowej losowej konfiguracji,
3. powrót do poprzedniej konfiguracji „undo” (wystarczy jednokrotne),
4. zmodyfikowanie losowo wybranego parametru (losowego elementu losowo wybranej macierzy, lub prawdopodobieństwa losowo wybranej macierzy) o niewielką wartość,
5. zapamiętanie obecnej konfiguracji na potrzeby kolejnego punktu:
6. wygenerowanie animacji poprzez interpolację liniową obecnej konfiguracji i ostatnio zapamiętanej.
  - Zapisz animację do plików np. `anim01.png`, `anim02.png`, ..., lub/i pokaż całą zapętloną poprzez tworzenie / kopiowanie jej klatek w osobnych obiektach `SDL_Surface`.

Interfejs nie musi być na bazie zdarzeń SDL, może używać `ReadLn`. Punktacja zgrubsza punkt-za-punkt z wyjątkiem akcji nr 5 (sama-z-siebie nie dodaje punktu).