­­Piotr Jastrzębski 2I1

**Instrukcja korzystania z biblioteki macierzy i wektorów rzadkich i pełnych**

**Obsługiwane typy danych**

- wektory rzadkie: ***spare\_vector***, gdzie wartości przechowywane są jako para (pozycja-wartość)- wektory pełne: ***normal\_vector***, reprezentowany jako klasyczny wektor (ciąg) następujących po sobie liczb- macierze pełne: ***normal\_matrix****,* klasyczna macierz prostokątna (wektor kolumnowy zawierający wektory                                                              normalne)- macierze wierszami rzadkie: ***a\_spare\_matrix****[[1]](#footnote-2)\*\*  
-* macierze kolumnami rzadkie: ***b\_spare\_matrix****[[2]](#footnote-3)\*\**- macierze rzadkie: ***c\_spare\_matrix****,* gdzie elementy niezerowe są przechowywane w liście razem jako trzy wartości                                                                 (wiersz-kolumna-wartość)

**Uwaga:** Wszystkie macierze są macierzami dwuwymiarowymi, a wszystkie wektory, wektorami wierszowymi.

**Tworzenie obiektu**W celu utworzenia pustego obiektu danej klasy należy wydać odpowiednie polecenie zadania typu z pierwszego akapitu a następnie podać nazwę jaką chcemy nadać obiektowi. np. *c\_spare\_matrix MacierzRzadka;* powoduje utworzenie obiektu typu *c\_spare\_matrix* o nazwie *MacierzRzadka*.

**Operacje na obiektach**  
Biblioteka umożliwia następujące operacja na ww. obiektach:

* Dodawanie realizowane za pomocą symbolu *+* (plus) np. *matrix1=matrix2+matrix3;* powoduje przypisanie do obiektu matrix1 macierzy będącą sumą macierzy matrix2 i matrix3. Należy mieć na uwadze, że dodawanie może odbywać się tylko i wyłącznie, gdy zgodne są wymiary macierzy lub wektorów. W momencie niezgodności rozmiarów sumowanych czynników program zasygnalizuje błąd.   
    
  Obiekty, które można dodawać to wektory oraz macierze. Poniżej podano wyniki poszczególnych działań:  
  *spare\_vector+spare\_vector=spare\_vector  
  spare\_vector+normal\_vector=normal\_vector  
  normal\_vector+spare\_vector=normal\_vector  
  normal\_vector+normal\_vector=normal\_vector*

*normal\_matrix+normal\_matrix=normal\_matrix  
normal\_matrix+c\_spare\_matrix=normal\_matrix  
c\_spare\_matrix+normal\_matrix=normal\_matrix  
c\_spare\_matrix+c\_spare\_matrix=normal\_matrix*

* Mnożenie realizowane za pomocą symbolu *\** (gwiazdka) np. *matrix1=matrix1\*matrix2;*powoduje przypisanie do obiektu matrix1 macierzy będącą iloczynem macierzy matrix2 i matrix3. Obiekty, które można mnożyć to wektory i macierze. Należy mieć jednak na uwadze, że mnożenie macierzy nie jest przemienne i istotne są tu poszczególne wymiary macierzy, czy też wymiar wektora przy mnożeniu wektor-macierz. W związku z faktem, że wektor został uznany w tej bibliotece za wektor wierszowy dopuszczone jest jedynie mnożenie, gdzie pierwszym czynnikiem jest odpowiednich rozmiarów wektor. W momencie niezgodności rozmiarów mnożonych czynników program zasygnalizuje błąd.

Obiekty, które można mnożyć to wektory oraz macierze. Poniżej podano wyniki poszczególnych działań:  
  
*normal\_vector\*normal\_matrix=normal\_vector  
normal\_vector\*c\_spare\_matrix=normal\_vector  
spare\_vector\*normal\_matrix=normal\_vector  
spare\_vector\*c\_spare\_matrix=normal\_vector  
  
normal\_matrix\*normal\_matrix=normal\_matrix  
normal\_matrix\*c\_spare\_matrix=normal\_matrix  
c\_spare\_matrix\*normal\_matrix=normal\_matrix  
c\_spare\_matrix\*c\_spare\_matrix=c\_spare\_matrix*

* Liczenie śladu macierzy za pomocą funkcji *trace\_nm(matrix1);*  lub trace\_c\_sm(matrix2);odpowiednio dla macierzy pełnej i rzadkiej typu C. Wartość zwracana przez funkcję to liczba typu double i jest ona wynikiem dodawania elementów leżących na przekątnej macierzy kwadratowej.
* Transponowanie macierzy za pomocą funkcji *transpose\_nm(matrix1);* lub *transpose\_c\_sm(matrix2);* odpowiednio dla macierzy pełnej i rzadkiej typu C. Funkcja operuje bezpośrednio na obiekcie podanym jako argument tej funkcji zamieniając ów argument na macierz transponowaną tego samego typu. Transponowane są zarówno macierze kwadratowe, jak i inne, gdzie zmianie ulegają również rozmiary macierzy.
* Odczyt i zapis do pliku. Pliki przechowujące obiekty klas mają zunifikowaną strukturę, różną dla każdego typu. W momencie niepowodzenia którejkolwiek z poniższych operacji, tj. plik nie istnieje itp. program zasygnalizuje stosowny błąd.
  + dla *spare\_vector:  
    load\_sv(obiekt, ”nazwa.rozszerzenie”);  
    ::* wczytuje z pliku o zadanej nazwie do uprzednio utworzonej zmiennej właściwego typu  
    *save\_sv(obiekt, ”nazwa.rozszerzenie”);  
    ::* zapisuje do pliku o zadanej nazwie obiekt typu spare\_vector  
      
    W pierwszej linii pliku znajduje się liczba całkowita mówiąca o rozmiarze wektora, w kolejnych liniach znajdują się pary liczb (całkowita i typu double oddzielone od siebie spacją) wskazujące na pozycję i wartość komórki w wektorze. Komórki wektora numerowane są od zera.
  + dla *normal\_vector:  
    load\_nv(obiekt, ”nazwa.rozszerzenie”);  
    ::* wczytuje z pliku o zadanej nazwie do uprzednio utworzonej zmiennej właściwego typu  
    *save\_nv(obiekt, ”nazwa.rozszerzenie”);  
    ::* zapisuje do pliku o zadanej nazwie obiekt typu normal\_vector  
      
    W pliku znajdują się, umieszczone w jednej linii, oddzielone od siebie spacjami, wartości poszczególnych kolumn wektora.
  + dla c\_*spare\_matrix*:  
    *load\_c\_sm(obiekt, ”nazwa.rozszerzenie”);  
    ::* wczytuje z pliku o zadanej nazwie do uprzednio utworzonej zmiennej właściwego typu  
    *save\_c\_sm(obiekt, ”nazwa.rozszerzenie”);  
    ::* zapisuje do pliku o zadanej nazwie obiekt typu c\_spare\_matrix  
      
    W pierwszej linii pliku znajduje się para liczb typu całkowitego oddzielonych od siebie spacjami. Pierwsza z tych liczb wskazuje na liczbę wierszy, druga na ilość kolumn.  
    W kolejnych wierszach umiejscowione są trójki liczb – dwie pierwsze typu całkowitego mówią o pozycji komórki, tj. pierwsza wskazuje numer wiersza, druga kolumny. Trzecia wartość typu double przechowuje wartość komórki. Zarówno kolumny, jak i wiersze numerowane są od zera.
  + dla *normal\_matrix:  
    load\_nm(obiekt, ”nazwa.rozszerzenie”);  
    ::* wczytuje z pliku o zadanej nazwie do uprzednio utworzonej zmiennej właściwego typu  
    *save\_nm(obiekt, ”nazwa.rozszerzenie”);  
    ::* zapisuje do pliku o zadanej nazwie obiekt typu normal\_matrix  
      
    W kolejnych liniach plików umieszczone są rzędy liczb typu double wskazujące na wartości poszczególnych komórek macierzy. Zapis przypomina ten, znany z klasycznej „papierowej” notacji.
* Wypisywanie obiektu na ekran za pomocą standardowego polecenia cout np. *cout << wektor;*

1. \* typy danych nie obsługiwane w tej wersji programu [↑](#footnote-ref-2)
2. [↑](#footnote-ref-3)