

PROGRAMOWANIE KOMPUTERÓW

Termin 3

Czas trwania 120 minut.

Blok 1-4

Dla klasy **Zbior** z plików *zbior.h* i *zbior.cpp*:

- 1) Napisz konstruktor przenoszący i przenoszący operator =, tak aby działały zgodnie z koncepcją kopii głębokiej.
- 2) Zmodyfikuj klasę **Zbiór** tak aby pozwalała na asocjację obustronną z inną instancją **Zbioru**. Asocjacja musi być tak napisana, aby usunięcie jednej z instancji nie dopuszczało do dereferencji wiszącego wskaźnika. Napisz metody do łączenia asocjacji i rozwiązywania asocjacji. Klasa nie powinna pozwalać na nadpisanie asocjacji, dopóki ta nie zostanie wcześniej zerwana i nie pozwalać na asocjację instancji ze sobą samą. Nieprawidłowe asocjacje powinny być ignorowane.
- 3) Napisz 2 testy jednostkowe sprawdzające czy:
 - czy asocjacja jest prawidłowo zrywana przy usunięciu jednej z instancji biorącej w niej udział,
 - po wykonaniu konstruktora przenoszącego i usunięciu dawcy, biorca ma prawidłowo ustawiony zasób.
- 4) Zdefiniuj funktor, pełniący rolę predykatu orzekającego o tym czy przekazana mu instancja **Zbioru** jest w asocjacji z instancją referencyjną. Instancja referencyjna powinna być ustawiana przy tworzeniu funktora tak aby funktor nie przechowywał jej kopii.
- 5) Napisz kod w funkcji main, w którym utwórz **listę** 10 instancji **Zbioru**. Następnie napisz algorytm, który spróbuje połączyć każdą z tych instancji z inną instancją w tej liście. Nieudane próby łączenia powinny pozostawić instancje bez asocjacji z inną instancją. Następnie używając funktora, policz ile instancji w liście ma ustawioną asocjację.

UWAGA, jeśli okaże się, że do wykonania tego polecenia niezbędne jest zdefiniowanie dodatkowych funkcjonalności nie wymienionych w poleceniu to należy to zrobić. Należy koniecznie uzasadnić w komentarzu powód.

Blok 5-8

Napisz prostą aplikację okienkową według następującej specyfikacji:

- 1) Zadaniem programu jest stanowić prosty spis studentów.
- 2) W głównym oknie musi znajdować się:
 - lista studentów wraz z ich ocenami z PK (w zakresie 2.0 – 5.0) z rozdzielczością 0.5,
 - etykieta zawierająca graficzne logo uczelni,
 - przycisk do usuwania z listy studentów, wpisów o ocenie niższej niż 3.0, po jego kliknięciu wpisy spełniające kryterium powinny być trwale usuwane z listy.
 - przycisk do wprowadzania danych studenta, otwierający nowe okno pozwalające wprowadzić dane nowego studenta i dodać go do listy.W drugim oknie muszą znajdować się kontrolki pozwalające na podanie danych studenta: nazwisko, ocena i rok studiów (1-5), oraz przyciski „ok” i „anuluj”.
- 3) Ponadto program powinien:
 - pozwolić na wprowadzanie tylko prawidłowych wartości pól liczbowych (rok, studiów i ocena), ocena nie tylko musi być w zakresie, ale musi mieć prawidłową rozdzielczość.
 - natychmiast po dodaniu nowego studenta automatycznie sortować listę najpierw względem roku studiów a potem względem ocen.
 - wyświetlać na liście zawsze najpierw rok studiów, potem nazwisko, a na końcu ocenę.
 - mieć schludny i skalowalny projekt okien, lista studentów musi pokazywać minimum 5 wpisów przy maksymalnym skurczeniu okna.
 - czytywać logo klubu automatycznie przy uruchomieniu.
 - **program nie powinien pozwalać na wprowadzanie bezsensownych informacji do kontrolek!**

W ewentualnych kwestiach nie sprecyzowanych specyfikacją, samodzielnie zaproponuj własne rozwiązanie.