

Projekty

1. Opracować klasę wzorcową `Kolejka` implementującą kolejką typu `doubly ended queue` (elementy mogą być dodawane lub usuwane z obydwu końców kolejki). Wykorzystać opracowaną klasę do rozwiązania następującego zadania: z klawiatury jest wprowadzany tekst zawierający błędy. Błędy można usuwać za pomocą klawisza `Backspace`. Każde naciśnięcie klawisza `Backspace` usuwa ostatnio wpisany znak.
2. Opracować klasę wzorcową `Lista` implementującą listę cykliczną. Wykorzystać listę do rozwiązania następującego zadania: grupa osób oczekuje na przydzielenie im pewnego zasobu, np. czasu procesora. Każdy, gdy nadejdzie jego kolejka korzysta z określonego kwantu czasu i następnie ponownie czeka. Nowa osoba może dostać przydział czasu dopiero wtedy, kiedy poprzednia osoba, która dołączyła do grupy dostała już swój przydział. Po wyczerpaniu potrzebnego czasu osoba opuszcza grupę.
3. Opracować klasę wzorcową `Stos`. Wykorzystać opracowaną klasę do rozwiązania następującego zadania: Firma lotnicza chce przetwarzać pytania klientów dotyczące realizowanych połączeń pomiędzy miastami. Klient pyta o lot, na przykład: Warszawa-San Francisco i dostaje odpowiedź, czy takie połączenie jest możliwe. Połączenie to może być realizowane poprzez miasta pośrednie, np. Warszawa-Zurich-Chicago-San Francisco.
4. Napisać program, który administruje bazą danych o studentach i przedmiotach. Student jest opisany za pomocą imienia, nazwiska i nr indeksu. Przedmioty są opisane za pomocą nazwy. Każdy student ma wybraną listę przedmiotów, oraz jedną ocenę za każdy z wybranych przedmiotów. Program powinien umożliwiać: dodawanie i usuwanie studentów do/z bazy, dodawanie i usuwanie przedmiotów do/z bazy, przypisywanie przedmiotów studentom, wstawianie ocen z przedmiotów, wyznaczanie średniej dla każdego studenta.
5. Projekt własny - uzgodniony z prowadzącym zajęcia.

Co powinna zawierać dokumentacja projektu z programowania obiektowego?

1. Strona tytułowa (tytuł projektu, kto jest autorem projektu, grupa, rok studiów, przedmiot).
2. Opis problemu.
3. Funkcjonalność programu – krótki opis działania programu, ew. zdjęty ekran.
4. Instrukcja obsługi programu – czytelna dla użytkownika, który nie zna C++.
5. Budowa programu – opis organizacji programu, w tym:
 - użyte struktury danych i algorytmy (dlaczego)
 - użyte klasy (opis)
 - użyte biblioteki (dlaczego)
6. Uwagi na temat efektywności programu.
7. Listing programu.
8. Kod programu na dyskietce (do zwrotu).