

Programowanie obiektowe

Lista 6.

Poniższe zadania mają być zaimplementowane w Javie. Dla każdego z zadań proszę podać krótki program ilustrujący możliwości zaimplementowanych klas.

Zadanie 1. Wybierz dowolne zadanie z poprzednich list dot. kolekcji (listy, grafy itp) i zaprogramuj je w Javie. Wymuś, aby implementowana kolekcja implementowała interfejs **Serializable** (z pakietu `java.io`) tak, aby można było zapisywać i odczytywać kolekcję z pliku dyskowego.

Jako ilustrację programu podaj program który zapisuje kolekcję na dysku a następnie ją odzyskuje.

Można skorzystać z dostępnych w internecie artykułów opisujących jak implementować ten interfejs.

Zadanie 2. Podobnie jak w poprzednim zadaniu, ale kolekcja winna implementować interfejs **Collection<E>** z pakietu `java.util`.

Poszukaj informacji, jakie korzyści daje implementacja tego interfejsu i zaprezentuj te korzyści w przykładach.

Zadanie 3. Zaprogramuj klasę implementującą dostęp do bufora o stałym rozmiarze przechowującym elementy typu generycznego `T`. Rozmiar bufora jest zadawany w konstruktorze. Implementacja powinna umożliwić działanie takiego bufora w środowisku wielowątkowym. Kolejność elementów pobieranych z bufora powinna być taka sama jak kolejność ich wkładania do bufora.

Korzystając z tej klasy zaimplementuj problem *producenta-konsumenta*: producent produkuje wyniki (napisy) i wkłada je do bufora. Jeśli bufor jest pełny, to producent zasypia czekając aż zwolni się miejsce w buforze. Konsument, jeśli w buforze jest jakiś element (napis), to go pobiera i "konsumuje". Zaimplementuj producenta i konsumenta jako dwa odrębne wątki.

Przed implementacją tego zadania warto zapoznać się z metodami `wait()` i `notify()` klasy **Thread**.

Zadanie 4. Wiele zadań programistycznych ma swoje naturalne rozwiązania w postaci *potoku* procesów, gdzie każdy proces wykonuje pewien fragment zadania i wynik przekazuje do następnego procesu wykorzystując np. taki bufor jak w zadaniu poprzednim. Przykładem takiego zadania jest problem odfiltrowania (wg. wskazanego kryterium) i posortowania alfabetycznie listy nazwisk: jeden proces pracuje w cyklu: pobierz wiersz, filtruj, wyślij do bufora; zaś drugi pobiera kolejne wiersze z bufora i wstawia je w odpowiednie miejsce i na końcu wypisuje je na konsolę. Zaimplementuj za pomocą wątków takie zadanie wykorzystując jako kanał komunikacyjny implementację bufora z poprzedniego zadania.

Zadanie 5. Algorytm sortowania tablicy elementów przez *scalanie* działa następująco: najpierw tablica jest dzielona na pół. Następnie każda z tych mniejszych tablic jest porządkowana. Na końcu obydwie posortowane tablice są scalane. Zaprogramuj sortowanie przez scalanie tablic elementów `int` tak, aby operacje sortowania podtablic były odrębnymi wątkami.

Za każde zadanie można otrzymać do 4 pkt, jednak można oddać nie więcej niż 2 zadania. Proszę do każdego ocenianego zadania dołączyć króciutki program ilustrujący możliwości zaprogramowanych klas.