

Rapport

Detta projekt är ett enkelt eventdrivet system skrivet i C++. Syftet är att simulera sensorer som genererar händelser, lagra dem i en logg, sortera dem och hantera larm baserat på sensorvärden. Programmet använder en meny där användaren kan generera nya händelser, skriva ut loggen, söka efter händelser från en specifik sensor, sortera loggen och visa aktiva larm. Fokus i uppgiften ligger på att använda abstrakta datatyper (ADT), dynamisk minneshantering och en enkel sorteringsalgoritm.

Systemet består av flera moduler. Den första är Event, som representerar en enskild händelse. Varje händelse innehåller ett tidsvärde, en formaterad tidssträng, en sensor-ID, en typ av händelse och ett sensorvärde. Händelser skapas med funktionen `make_random_event`, som genererar slumpmässiga värden.

Nästa modul är Queue, som är implementerad som en cirkulär kö (ringbuffer). Den används för att tillfälligt lagra inkommande händelser innan de flyttas till loggen. Kö-modulen är en ADT och döljer sin interna representation. Den erbjuder funktioner för att lägga till och ta bort element samt kontrollera om kön är full eller tom.

EventLog är en annan ADT som fungerar som en dynamisk array. Den lagrar alla händelser som har passerat kön. Om loggen blir full utökas den automatiskt. Modulen erbjuder funktioner för att lägga till händelser, hämta händelser, ersätta händelser och läsa loggens storlek. Denna modul används även av sorteringsalgoritmen.

Sorteringen implementeras med insertion sort, som sorterar händelserna efter deras tidsstämpel. Sorteringen är implementerad som en funktion som tar en EventLog som argument. Programmet använder en funktionspekare för att välja sorteringsalgoritm, vilket är en enkel form av Strategy-mönstret.

AlarmSet är en separat modul som håller reda på sensorer vars värden överstiger en viss gräns. Om ett sensorvärde går över tröskeln läggs ett larm till. Om värdet går tillbaka till normalt tas larmet bort. Detta ger programmet en extra funktionalitet och visar hur man kan hantera dynamiska listor.

Huvudprogrammet kopplar samman alla moduler. Det innehåller en meny där användaren kan välja olika funktioner. Vid varje tick skapas nya händelser, läggs i kön och flyttas sedan till loggen. Användaren kan sedan skriva ut loggen, söka efter specifika sensorer, sortera loggen eller visa aktiva larm.

Programmet har testats genom att generera flera händelser, skriva ut loggen, sortera den och söka efter specifika sensor-ID:n. Larmfunktionen har också testats genom att generera höga och låga sensorvärden. Alla funktioner fungerar som förväntat.

Jamil saad

Sammanfattningsvis visar projektet användning av abstrakta datatyper, dynamisk minneshantering, cirkulär kö, insertion sort och modulär programstruktur. Systemet fungerar som avsett och uppfyller kraven för en C-nivåuppgift.