

Do zadania jest dołączony plik o rozszerzeniu csv. Zadanie polega na wczytaniu tego pliku do pamięci komputera (sprasowanie go), a następnie wykonanie analizy danych zawartych w nim. Pierwsza linia jest linią informacyjną, która opisuje jakie są dane w pliku. Ta linia nie jest potrzebna w procesie analizy więc można ją pominąć. Dane w pliku (wszystkie wyrażenia są wyrażone w watach [W] oprócz daty i godziny):

- Data i godzina pomiaru,
- Autokonsumpcja jest to energia elektryczna, która zostaje wyprodukowana przez falownik i od razu zużyta przez odbiorniki elektryczne,
- Eksport jest to energia, która została wyprodukowana i wysłana do sieci energetycznej,
- Import jest to energia, która zostaje pobrana z sieci energetycznej,
- Pobór jest to energia zużyta przez różne odbiorniki elektryczne,
- Produkcja jest to energia, która została wyprodukowana przez falownik,

Napisz program, który wczytuje dane. Wszystkie klasy muszą być w osobnych plikach. Na początku stwórz klasę gdzie każda pojedyncza linijka zasila w dane jeden obiekt (pojedynczy punkt). Wszystkie obiekty muszą być zapisane na sterwie. Druga klasa jest obiektem, który wygląda jak drzewo. Korzeniem jest rok, następnym poziomem jest miesiąc, następnie dzień oraz ćwiartka (6 godzin). Ćwiartki mamy cztery (00:00-5:45, 6:00-11:45, 12:00-17:45, 18:00-23:45). Wczytaj dane do każdej ćwiartki. Zobacz, że czas w pliku zmienia się co 15 minut i dane muszą być poukładane od najmniejszej godziny i minuty do największej godziny i minuty.

Do tej pory została napisana część programu, która jest odpowiedzialna za wczytanie danych do pamięci. Teraz napisz klasę, która będzie odpowiedzialna za mechanikę programu:

- Wypisz sumę autokonsumpcji wyznaczoną w datach i godzinach (np.: użytkownik może poprosić o wyliczenie sumy od 3.10.2021 10:15 co 3.10.2021 16:45),
- Wypisz sumę eksportu wyznaczoną w datach i godzinach,
- Wypisz sumę importu wyznaczoną w datach i godzinach,
- Wypisz sumę poboru wyznaczoną w datach i godzinach,
- Wypisz sumę produkcji wyznaczoną w datach i godzinach,
- Wypisz średnią autokonsumpcji wyznaczoną w datach i godzinach,
- Wypisz średnią eksportu wyznaczoną w datach i godzinach,
- Wypisz średnią importu wyznaczoną w datach i godzinach,
- Wypisz średnią poboru wyznaczoną w datach i godzinach,
- Wypisz średnią produkcji wyznaczoną w datach i godzinach,
- Porównaj autokonsumpcję wyznaczoną w dwóch przedziałach,
- Porównaj eksport wyznaczony w dwóch przedziałach,
- Porównaj import wyznaczony w dwóch przedziałach,
- Porównaj pobór wyznaczony w dwóch przedziałach,
- Porównaj produkcję wyznaczoną w dwóch przedziałach,
- Wyszukać i wypisać na ekran wszystkie rekordy zawierające warunek np.: szukam importu gdzie cyfra szukana to 12.0021, plus zakres tolerancji +/- 1.2345, plus przedział czasowy,
- Wypisz na ekran dane wyznaczone w przedziale

Przy poruszaniu się po węzłach zastosuj wzorec projektowy iterator. Zwróć uwagę, że dane mogą być niepełne (bez ciągłości czasowej) dlatego należy wziąć to pod uwagę.

Po uruchomieniu programu wyświetla się menu gdzie użytkownik wybiera co chce zrobić. Dodatkowo w menu mamy mieć opcje wczytania pliku csv, zapis do pliku binarnego wczytanych danych, odczyt z pliku binarnego danych (zakładamy że odczyt z pliku zawsze musi być do pustego programu).

Przy wczytywaniu pliku z csv należy prowadzić bieżącą analizę i wychwytywać błędne linie. Mogą pojawić się jakieś nieprzewidywalne linie, których nie można wczytywać, linie mogą być w różnych kolejnościach, linie mogą powtarzać się (wtedy tylko jedna linia jest wczytana), linie mogą być niepełne, mogą zdarzyć się puste linie. Przy wczytywaniu pliku csv program nie może zawiesić się. Program powinien tworzyć plik `log_data_godzina.txt` gdzie zapisuje postęp wczytywania (poprawne i niepoprawne rekordy) oraz plik `log_error_data_godzina.txt` gdzie zapisuje same niepoprawne rekordy. Niepoprawny rekord powinien być odrzucony i zapisany w obu plikach (`log_data_godzina.txt`, `log_error_data_godzina.txt`), szczególnie w pliku z błędami. Data i godzina w nazwie pliku musi być pobierana z komputera automatycznie (tak aby nie nadpisywać plików). Po wczytaniu pliku program powinien wypisać podsumowanie na ekran (ile było poprawnych i niepoprawnych rekordów).

Do projektu należy napisać dokumentację w Latex, DoxyGen oraz zapisać projekt w serwisie GitHub. Należy napisać testy „GoogleTest” dzięki któremu można sprawdzić metody. Można korzystać z GitHub Copilot. Projekt piszemy w języku C++. Projekt dwuosobowy