

## Ogólne wymagania dotyczące projektu trzeciego

1. Zadanie polega na przeprowadzeniu symulacji działania danej organizacji (instytucji) w czasie. Upływ czasu może być modelowany poprzez zastosowanie pętli, której każdy obrót stanowi umowną jednostkę czasu.
2. Projekty są sformułowane w sposób ogólny, a podane założenia stanowią jedynie podstawowy schemat działania organizacji. Jedną z części zadania jest uszczegółowienie sposobu działania organizacji. Zalecane jest skonsultowanie przyjętych założeń przed oddaniem projektu. Możliwa jest też ewentualna modyfikacja założeń sformułowanych w treści zadania po uzgodnieniu z prowadzącą.
3. W tym projekcie istotne jest wykorzystanie różnorodnych elementów języka C++, dlatego w przypadku każdego z zadań należy:
  - a. zaprojektować hierarchię klas odpowiadającą treści zadania i przyjętym szczegółowym założeniom (wykorzystanie mechanizmu dziedziczenia);
  - b. zdefiniować interfejsy klas poprzez utworzenie dla nich abstrakcyjnych klas bazowych z publicznymi metodami (wszystkie powinny być metodami czysto wirtualnymi). Tutaj należy uwzględnić tylko te metody, które realizują pełne operacje (nie ma potrzeby uwzględniać metod pomocniczych);
  - c. wykorzystać kolekcje z biblioteki STL (wektory, listy, kolejki itp.) – tym razem nie tworzymy ich samodzielnie;
  - d. wyświetlać przebieg symulacji w konsoli, a jednocześnie zapisywać go do pliku. Wyświetlanie w konsoli powinno następować w takim tempie, by użytkownik zdążył przeczytać wypisywany komunikat;
  - e. wczytywać wszelkie parametry programu z pliku;
  - f. zastanowić się nad możliwymi przypadkami, w których program nie będzie działał poprawnie i zastosować do ich obsługi mechanizm wyjątków. Może to dotyczyć zarówno sposobu działania organizacji (jakaś sytuacja nie powinna się zdarzyć), jak i sposobu działania samego programu (nieudany odczyt/zapis do pliku).
4. Użytkownik powinien mieć możliwość ustawiania parametrów programu testującego, np. długości czasu symulacji (liczby iteracji), liczby obiektów danego typu biorących udział w symulacji (np. liczba obsługujących pracowników, liczba kas w sklepie).
5. W symulacji korzysta się z generatorów liczb pseudolosowych. Do generacji tych liczb proszę nie używać bibliotecznej funkcji rand, tylko skorzystać z biblioteki random.h z C++ 11.
6. Do projektu należy dołączyć dokumentację projektową w postaci pliku pdf. Powinna ona zawierać:
  - a. opis wszystkich przyjętych założeń;
  - b. opis hierarchii klas oraz relacji pomiędzy nimi;
  - c. opis sposobu testowania programu (przeprowadzenia symulacji);
  - d. wskazanie wykorzystanych elementów biblioteki STL;
  - e. opis zidentyfikowanych sytuacji wyjątkowych i ich obsługi.
7. Należy pamiętać o podziale programu na implementację klas i program testujący. Proszę też pamiętać o komentowaniu kodu. Kodu nie należy umieszczać w dokumentacji.
8. Ocena jest wystawiana po zapoznaniu się z dokumentacją i kodem źródłowym przez prowadzącą. W przypadku chęci skonsultowania poprawności projektu przed wtorkowym laboratorium, proszę o przesłanie tych danych odpowiednio wcześniej – do poniedziałku do godziny 20.00.

9. Termin oddania projektu: 5.06.2018, punktacja:
- a. Projekt hierarchii klas – 3 pkt
  - b. Identyfikacja sytuacji wyjątkowych – 2 pkt
  - c. Implementacja klas – 10 pkt
  - d. Program testujący (symulacja) – 5 pkt
  - e. Dokumentacja – 5 pkt

### Projekt 3.

Proszę zasymulować działanie **przychodni**, przyjmując następujące założenia:

1. W przychodni pracują lekarze różnych specjalności. Od specjalności lekarza zależy, jak długo wykonywane są badania i jak często lekarz jest dostępny.
2. Lekarz jest dostępny co pewną liczbę jednostek czasu na pewną liczbę jednostek czasu.
3. Pacjent zgłasza się do przychodni do lekarza danej specjalności. Jeśli lekarz tej specjalności jest aktualnie dostępny, pacjent jest od razu do niego kierowany. W przeciwnym razie pacjent jest zapisywany do kolejki oczekujących na danego specjalistę.
4. Pacjent może wykonać dodatkowe działania w rejestracji: zmienić lekarza, zaktualizować swoje dane, zamówić kopię swojej karty itp.