

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: ruch pakietów w sieci

Założenia:

- ☞ sieć składa się z podsieci o dwóch różnych typach topologii;
- ☞ pierwszy typ - gwiazda, drugi – token ring;
- ☞ pakiety napływają i odpływają z systemu poprzez jedną z n ($n > 4$) sieci token ring;
- ☞ każda token ring dodatkowo jest połączona ze wspólną siecią typu gwiazda;
- ☞ pojemność (przepustowość) sieci jest ograniczona do poziomu x ;
- ☞ w każdej sieci typu token ring znajduje się y komputerów, z których każdy również może generować i odbierać pakiety;
- ☞ w sieci typu gwiazda jest z komputerów, działających analogicznie do ww.;
- ☞ każdy pakiet znajdujący się w sieci posiada adres nadawcy;
- ☞ nie ma zagubionych pakietów oraz każdy prędzej czy później trafi do odbiorcy.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 3.

Problem do rozwiązania: ruch pakietów w sieci

Założenia:

- ☞ sieć składa się z podsieci o dwóch różnych typach topologii;
- ☞ pierwszy typ - gwiazda, drugi – token ring;
- ☞ pakiety napływają i odpływają z systemu poprzez jedną z n ($n > 4$) sieci token ring;
- ☞ każda token ring dodatkowo jest połączona ze wspólną siecią typu gwiazda;
- ☞ pojemność (przepustowość) sieci jest ograniczona do poziomu x ;
- ☞ w każdej sieci typu token ring znajduje się y komputerów, z których każdy również może generować i odbierać pakiety;
- ☞ w sieci typu gwiazda jest z komputerów, działających analogicznie do ww.;
- ☞ każdy pakiet znajdujący się w sieci posiada adres nadawcy;
- ☞ nie ma zagubionych pakietów oraz każdy prędzej czy później trafi do odbiorcy.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 3.

Problem do rozwiązania: pasieka

Założenia:

- ☞ pasieka składa się n ($n > 6$) uli;
- ☞ w każdym z nich znajdują się pszczoły, które losowo wylatują po nektar;
- ☞ każda z nich poszukuje przez losowy czas kwiatu, zbiera nektar a następnie wraca do ula;
- ☞ w każdym ulu jest m ($m \geq 10$) wejść o ograniczonej przepustowości;
- ☞ każda z pszczoł składa swój nektar, do jednego z k ($k > 20$) plastrów, w których to tworzony jest miód;
- ☞ losowo przychodzi pszczelarz, wybierając miód z pełnych plastrów, jednocześnie wstrzymując możliwość składowania nektaru przez pszczoły (do tych plastrów).

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: pasieka

Założenia:

- ☞ pasieka składa się n ($n > 6$) uli;
- ☞ w każdym z nich znajdują się pszczoły, które losowo wylatują po nektar;
- ☞ każda z nich poszukuje przez losowy czas kwiatu, zbiera nektar a następnie wraca do ula;
- ☞ w każdym ulu jest m ($m \geq 10$) wejść o ograniczonej przepustowości;
- ☞ każda z pszczoł składa swój nektar, do jednego z k ($k > 20$) plastrów, w których to tworzony jest miód;
- ☞ losowo przychodzi pszczelarz, wybierając miód z pełnych plastrów, jednocześnie wstrzymując możliwość składowania nektaru przez pszczoły (do tych plastrów).

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkanie.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 3.

Problem do rozwiązania: biuro obsługi klienta

Założenia:

- ☞ biuro posiada n ($n > 6$) sal, których odbywa się obsługa klienta;
- ☞ w każdym z pomieszczeń jest k ($k > 4$) stanowisk;
- ☞ dodatkowo, na potrzeby osób obsługujących pracuje x ($x > 3$) konsultantów, z którymi w razie wątpliwości można skontaktować się telefonicznie;
- ☞ petenci wchodzą i wychodzą w biura dwoma wejściami;
- ☞ każdy z petentów losowo korzysta z obsługi;
- ☞ po wybraniu pomieszczenia, oczekuje, gdy w tym czasie są obsługiwane inne osoby;
- ☞ po zakończeniu obsługi może wybrać inne pomieszczenie lub też wyjść z biura (losowo);
- ☞ każda z osób obsługujących, w trakcie obsługi petenta może potrzebować dodatkowych informacji. W tym celu kontaktuje się z dodatkowymi konsultantami biura;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkanie.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: biuro obsługi klienta

Założenia:

- ☞ biuro posiada n ($n > 6$) sal, których odbywa się obsługa klienta;
- ☞ w każdym z pomieszczeń jest k ($k > 4$) stanowisk;
- ☞ dodatkowo, na potrzeby osób obsługujących pracuje x ($x > 3$) konsultantów, z którymi w razie wątpliwości można skontaktować się telefonicznie;
- ☞ petenci wchodzą i wychodzą w biura dwoma wejściami;
- ☞ każdy z petentów losowo korzysta z obsługi;
- ☞ po wybraniu pomieszczenia, oczekuje, gdy w tym czasie są obsługiwane inne osoby;
- ☞ po zakończeniu obsługi może wybrać inne pomieszczenie lub też wyjść z biura (losowo);
- ☞ każda z osób obsługujących, w trakcie obsługi petenta może potrzebować dodatkowych informacji. W tym celu kontaktuje się z dodatkowymi konsultantami biura;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: biurowiec

Założenia:

- ☞ $k (k \geq 10)$ pięter, $n (n > 4)$ wind, $x (x > 10)$ pomieszczeń na każdym piętrze;
- ☞ pracownicy wchodzą i wychodzą z biurowca jednym dwóch wejść (parter);
- ☞ przemieszczają się pomiędzy pokojami na tym samym lub różnych piętrach;
- ☞ windy oraz pomieszczenia mają ograniczoną pojemność;
- ☞ pracownicy czekają na windy, rezygnują z wejścia do pełnego pomieszczenia i wybierają losowo inne;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 2.

Problem do rozwiązania: biurowiec

Założenia:

- ☞ $k (k \geq 10)$ pięter, $n (n > 4)$ wind, $x (x > 10)$ pomieszczeń na każdym piętrze;
- ☞ pracownicy wchodzą i wychodzą z biurowca jednym dwóch wejść (parter);
- ☞ przemieszczają się pomiędzy pokojami na tym samym lub różnych piętrach;
- ☞ windy oraz pomieszczenia mają ograniczoną pojemność;
- ☞ pracownicy czekają na windy, rezygnują z wejścia do pełnego pomieszczenia i wybierają losowo inne;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 1.

Problem do rozwiązania: parking wielopoziomowy

Założenia:

- ☞ k_1 ($k_1 \geq 3$) poziomów wyższych oraz k_2 ($k_2 \geq 3$) niższych, n_1 ($n_1 > 4$) wjazdów oraz n_2 ($n_2 > 4$) zjazdów, x ($x > 100$) miejsc parkingowych na każdym poziomie;
- ☞ samochody wjeżdżają i wyjeżdżają z parkingu jednym 4 wjazdów (poziom zerowy);
- ☞ co losowy samochód jest przeparkowywany pomiędzy miejscami postojowymi na tym samym lub różnych piętrach;
- ☞ co losowy ulega uszkodzeniu, co wymaga przyjazdu pomocy drogowej i odholowania;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 3.

Problem do rozwiązania: parking wielopoziomowy

Założenia:

- ☞ k_1 ($k_1 \geq 3$) poziomów wyższych oraz k_2 ($k_2 \geq 3$) niższych, n_1 ($n_1 > 4$) wjazdów oraz n_2 ($n_2 > 4$) zjazdów, x ($x > 100$) miejsc parkingowych na każdym poziomie;
- ☞ samochody wjeżdżają i wyjeżdżają z parkingu jednym 4 wjazdów (poziom zerowy);
- ☞ co losowy samochód jest przeparkowywany się pomiędzy miejscami postojowymi na tym samym lub różnych piętrach;
- ☞ co losowy ulega uszkodzeniu, co wymaga przyjazdu pomocy drogowej i odholowania;
- ☞

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: badania szpitalne

Założenia:

- ☞ $k (k \geq 5)$ pięter, $n (n > 4)$ wind, $x (x > 30)$ sal na każdym piętrze;
- ☞ sale i windy mają ograniczoną pojemność;
- ☞ na parterze są gabinety badań specjalistycznych;
- ☞ chorzy napływają do szpitala jednym trzema wejść (parter), tyle samo jest wyjść;
- ☞ dodatkowo chory może być przywieziony karetką pogotowia;
- ☞ chorzy trafiają losowo do sal, gdy wybrana sala jest pełna, losowana jest następna;
- ☞ chorzy w salach przebywają losowy czas, w czasie którego muszą kilkakrotnie udawać się na parter na badania;
- ☞ każdy chory wychodzi ze szpitala po wykonaniu wszystkich badań, jakie w momencie przyścia miał do wykonania;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 2.

Problem do rozwiązania: badania szpitalne

Założenia:

- ☞ $k (k \geq 5)$ pięter, $n (n > 4)$ wind, $x (x > 30)$ sal na każdym piętrze;
- ☞ sale i windy mają ograniczoną pojemność;
- ☞ na parterze są gabinety badań specjalistycznych;
- ☞ chorzy napływają do szpitala jednym trzema wejść (parter), tyle samo jest wyjść;
- ☞ dodatkowo chory może być przywieziony karetką pogotowia;
- ☞ chorzy trafiają losowo do sal, gdy wybrana sala jest pełna, losowana jest następna;
- ☞ chorzy w salach przebywają losowy czas, w czasie którego muszą kilkakrotnie udawać się na parter na badania;
- ☞ każdy chory wychodzi ze szpitala po wykonaniu wszystkich badań, jakie w momencie przyścia miał do wykonania;

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 1.

Problem do rozwiązania: centrum rozrywki

Założenia:

- ☞ $k (k \geq 2)$ pięter, $n (n > 2)$ schodów ruchomych, $x (x > 5)$ sal kinowych na każdym piętrze powyżej 1-ego;
- ☞ sale i schody mają ograniczoną pojemność;
- ☞ na parterze są restauracje i kawiarnie (każdego rodz. > 6);
- ☞ na piętrze 1: bilard (2 sale), dyskoteka (1), miejsce zabaw dla dzieci (2);
- ☞ klienci przychodzą jednym trzema wejść (parter), tyle samo jest wyjść;
- ☞ każdy klient może wyjść z centrum, gdy:
 - zobaczy co najmniej 2 filmy lub skorzysta z jednego miejsca na 1 piętrze;
 - zje w restauracji lub kawiarni;
- ☞ co losowy klient nie kończy rozrywki po pobycie w restauracji lub kawiarni lecz wybiera inną.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 3.

Problem do rozwiązania: centrum rozrywki

Założenia:

- ☞ $k (k \geq 2)$ pięter, $n (n > 2)$ schodów ruchomych, $x (x > 5)$ sal kinowych na każdym piętrze powyżej 1-ego;
- ☞ sale i schody mają ograniczoną pojemność;
- ☞ na parterze są restauracje i kawiarnie (każdego rodz. > 6);
- ☞ na piętrze 1: bilard (2 sale), dyskoteka (1), miejsce zabaw dla dzieci (2);
- ☞ klienci przychodzą jednym trzema wejść (parter), tyle samo jest wyjść;
- ☞ każdy klient może wyjść z centrum, gdy:
 - zobaczy co najmniej 2 filmy lub skorzysta z jednego miejsca na 1 piętrze;
 - zje w restauracji lub kawiarni;
- ☞ co losowy klient nie kończy rozrywki po pobycie w restauracji lub kawiarni lecz wybiera inną.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 1.

Problem do rozwiązania: restauracja

Założenia:

- ☞ $k (k \geq 2)$ poziomów, $n (n > 2)$ schodów, $x (x \geq 20)$ stolików na każdym poziomie;
- ☞ sale i schody mają ograniczoną pojemność;
- ☞ w restauracji pracuje $y (y \leq 20)$ kelnerów;
- ☞ klienci przychodzą jednym wejściem (parter), będącym jednocześnie wyjściem;
- ☞ każdy klient po wejściu czeka, aż kelner wskaże mu wolny stół;
- ☞ uwzględnić sytuacje: zamawiania posiłku, otrzymania, spożywania, płacenia;
- ☞ do zamawiania posiłku, otrzymania i płacenia potrzebna jest obecność kelnera;
- ☞ po zapłaceniu za posiłek klient wychodzi.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: restauracja

Założenia:

- ☞ $k (k \geq 2)$ poziomów, $n (n > 2)$ schodów, $x (x \geq 20)$ stolików na każdym poziomie;
- ☞ sale i schody mają ograniczoną pojemność;
- ☞ w restauracji pracuje $y (y \leq 20)$ kelnerów;
- ☞ klienci przychodzą jednym wejściem (parter), będącym jednocześnie wyjściem;
- ☞ każdy klient po wejściu czeka, aż kelner wskaże mu wolny stół;
- ☞ uwzględnić sytuacje: zamawiania posiłku, otrzymania, spożywania, płacenia;
- ☞ do zamawiania posiłku, otrzymania i płacenia potrzebna jest obecność kelnera;
- ☞ po zapłaceniu za posiłek klient wychodzi.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 3.

Problem do rozwiązania: taksówki

Założenia:

- ☞ $k (k \geq 10)$ postojów, $n (n \leq 20)$ miejsc dla taksówek;
- ☞ $x (x > 5)$ miejsc docelowych: lotnisko, dworzec PKP, PKS, itp.
- ☞ klienci napływają równomiernie do każdego z postojów;
- ☞ dla wszystkich tras jest jedno rondo, przez które muszą przejeżdżać taksówki;
- ☞ jeśli dwóch klientów jedzie w tę samą stronę i są miejsca w taksówce mogą jechać razem;
- ☞ w miejscach docelowych liczba stanowisk, gdzie można wysadzić pasażera jest ograniczona;
- ☞ co losowy pasażer nie ma pieniędzy za kurs i jest odwożony na policję, gdzie zarówno on jak i taksówkarz przebywa określony czas.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 2.

Problem do rozwiązania: taksówki

Założenia:

- ☞ $k (k \geq 10)$ postojów, $n (n \leq 20)$ miejsc dla taksówek;
- ☞ $x (x > 5)$ miejsc docelowych: lotnisko, dworzec PKP, PKS, itp.
- ☞ klienci napływają równomiernie do każdego z postojów;
- ☞ dla wszystkich tras jest wspólne rondo, przez które muszą przejeżdżać taksówki;
- ☞ jeśli dwóch klientów jedzie w tę samą stronę i są miejsca w taksówce mogą jechać razem;
- ☞ w miejscach docelowych liczba stanowisk, gdzie można wysadzić pasażera jest ograniczona;
- ☞ co losowy pasażer nie ma pieniędzy za kurs i jest odwożony na policję, gdzie zarówno on jak i taksówkarz przebywa określony czas.

Język implementacji: **Ada**
Środowisko implementacyjne: **GNAT Ada 95, Java 5+**
Termin wykonania: **ostatnie zajęcia laboratoryjne**

Podstawowe wymagania:

- a) liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
- b) kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
- c) obok poprawnej identyfikacji sekcji krytycznych program musi brać pod uwagę czytelność i estetykę interfejsu użytkownika oraz zdolność percepcji osoby oceniającej,
- d) obowiązuje zakaz konstrukcji interfejsu na zasadzie wypisywania kolejnych linii na ekran,
- e) dodatkowo, ekran także jest współdzielonym zasobem i w tym celu:
 - program zawiera dedykowane zadanie (task) wypisujące komunikaty na ekranie, z którym pozostałe zadania odbywają spotkanie w celu wypisania odpowiedniego komunikatu,
 - dla zadań laboratoryjnych, dla których podstawowym mechanizmem synchronizacji jest spotkanie, ekran należy chronić obiektem chronionym (monitorem) zamiast zadania (task) odbywającego spotkania.

Sprawozdanie powinno zawierać następujące elementy:

- 1) stronę tytułową,
- 2) niniejszą treść zadania,
- 3) syntetyczny opis problemu – przyjęte założenia,
- 4) wykaz współdzielonych zasobów,

Wariant zadania: wariant 4.

Problem do rozwiązania: taksówki

Założenia:

- ☞ $k (k \geq 10)$ postojów, $n (n \leq 20)$ miejsc dla taksówek;
- ☞ $x (x > 5)$ miejsc docelowych: lotnisko, dworzec PKP, PKS, itp.
- ☞ klienci napływają równomiernie do każdego z postojów;
- ☞ dla wszystkich tras jest wspólne rondo, przez które muszą przejeżdżać taksówki;
- ☞ jeśli dwóch klientów jedzie w tę samą stronę i są miejsca w taksówce mogą jechać razem;
- ☞ w miejscach docelowych liczba stanowisk, gdzie można wysadzić pasażera jest ograniczona;
- ☞ co losowy pasażer nie ma pieniędzy za kurs i jest odwożony na policję, gdzie zarówno on jak i taksówkarz przebywa określony czas.