Самостоятельная работа

Мы переходим к поддержке синхронизации в C++11

Сегодня на лекции вы работаете над заданиями, представленными на слайдах 10_4, 10_5, 10_6

Результаты представляются в конце лекции в виде текстового документа на электронную почту. Письма принимаются до 16 часов 17.04.2020

Помните, что я легко вижу плагиат в ваших программных кодах

class mutex

public member function

(constructor) constructs the mutex
(destructor) destroys the mutex

Locking

lock()
locks the mutex, blocks if the mutex is not available

try_lock()
tries to lock the mutex, returns if the mutex is not available

unlock()
unlocks the mutex

Protecting Shared Data

Mutual Exclusion with std::mutex

std::mutex is usually not accessed directly:

std::unique_lock, std::lock_guard

```
std::mutex m;
unsigned counter=0;

unsigned increment()
{
   std::lock_guard<std::mutex> lk(m);
   return ++counter;
}

unsigned query()
{
   std::lock_guard<std::mutex> lk(m);
   return counter;
}
```

std::mutex

Явная блокировка и разблокировка могут привести к ошибкам, например, …

Классы «обертки» позволяют непротиворечиво использовать мьютекс в RAII-стиле с автоматической блокировкой и разблокировкой в рамках одного блока:

- lock_guard

  когда объект создан, он пытается получить мьютекс (вызывая lock()), а когда объект уничтожен, он автоматически освобождает мьютекс (вызывая unlock())

- unique_lock

  в отличие от lock_guard, также поддерживает отложенную блокировку, временную блокировку, рекурсивную блокировку и использование условных переменных

Самостоятельная работа

Задание 1
Защита списка с помощью мьютекса

std::list <int> some_list;

Реализовать методы
void add_to_list(int new_value) { !!! }
bool list_contains(int value_to_find) { !!! }

Самостоятельная работа

Задание 2
Synchronizing concurrent std::cout use

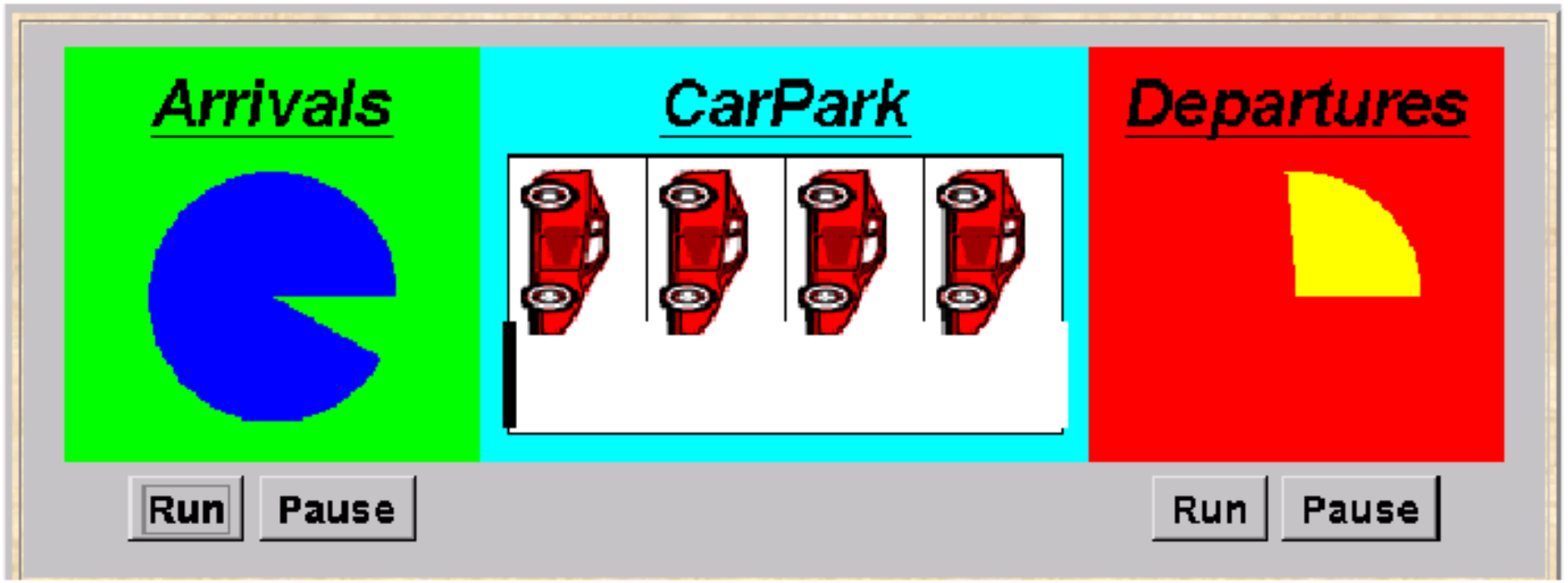
One data structure that is typically used very often for printing is std::cout. If multiple threads access cout concurrently, then the output will appear in interesting mixed patterns on the terminal. In order to prevent this, we would need to write our own function that prints in a concurrency-safe fashion.

Предложить свой вариант потокобезопасного использования std::cout

Самостоятельная работа

Задание 3

Разработать класс CarPark на языке программирования Java



A controller is required for a carpark, which only permits cars to enter when the carpark is not full and permits cars to leave when there it is not empty. Car arrival and departure are simulated by separate threads.