Задания к ЛР№6\_потоки

|  |  |
| --- | --- |
| 1 | **Крестики – нолики**. Вариант графический. Сообщение о выигравшем.  **Гонки** – счет. 5 участников ведут счет до 10. По окончании должно выдаваться консольное сообщение каждым об окончании счета и имя потока-победителя. |
| 2 | **Гонки**. 3 участника наперегонки наращивают каждый свой ProgressBar на разное случайное число единиц прогресса. По окончании должно появиться сообщение о том, кто прибыл первым, кто последним. Вариант графический.  **Крестики – нолики**. Упрощенный вариант – консольный. Должны выдаваться сообщения: 1-й поток окончил! 2-й поток окончил! И имя выигравшего потока, если была выигрышная комбинация. |
| 3 | **Спорщики.** Три участника путешествия наперебой выбирают маршрут (суша, море, воздух). По окончании объявляется победитель и выбранный способ перемещения.  **Слова.** Пять потоков в строгой очередности считывают слова из одного и того же файла (Поток1: слово1, слово5, слово9; Поток2: слово2, слово6, слово10 и т.д. )в свой массив слов, затем выводят в консоль почитанное: сначала все слова потока1, затем потока 2, и т.д. |
| 4 | **Лото.** Четыре игрока строго по очереди выставляют по одной свей фишке (у каждого свой цвет фишки) на общую доску, раскрашенную в разные цвета, о чем каждый выдает сообщение, например: «Красный, фишка 3». По окончании (когда все клетки заполнены) сообщается, кто закончил первым. Фишек по 10 у каждого, мест на доске 30. Расположение цветов на доске случайное. (Графический вариант).  **Лото1.** Предыдущая задача, но порядок игроков и размещение фишек случайное, у каждого из 4-х игроков фишки одного цвета. Есть графическое поле, на которое все игроки должны попытаться выставить свои 10 фишек в случайные места. Но мест всего 30. Побеждает тот, кто выставит больше |
| 5 | **Гонка.** Несколько потоков (машинок) стартуют одновременно. Но скорости возрастают случайным образом. На финише должен быть объявлен победитель. Можно консольно.  **Автостоянка.** Доступно несколько машиномест. На одном месте может находиться только один автомобиль. Если все места заняты, то автомобиль не станет ждать больше определенного времени и уедет на другую стоянку, графический вариант лучше. |
| 6 | Графический **тренажер клавиатуры**. Буквы движутся по горизонтали слева-направо в 3-х потоках на разных уровнях горизонта. Пользователь нажимает соответствующую букву, если успевает за время пересечения буквой поля, счет игрока наращивается.  **Map-Reduce.** Главный поток рассылает 3-м другим строки, задача каждого потока составить hashmap – слово-сколько раз оно встречается в строке. Затем главный поток должен собрать и агрегировать результат. |
| 7 | **Produser-Consumer.** Добавьте еще два объекта Consumer, которые будут запускаться тоже отдельным потоком.  Выводите на консоль какой из объектов Consumer обработал объект из очереди.  Создать **массив**, заполненный случайными числами, на 500 элементов. На графическом интерфейсе создать textBox1 и textBox2, в один текст бокс вывести первую часть массива а во второй вторую часть массива. Выполнить с помощью потоков. |
| 8 | Создать **3 функции**. Главную и 2 второстепенных (побочных). В определенный момент из главной во второстепенные передается число n и i. Один из побочных потоков считает n! - числитель, другой - Σ(n+i), i=[0,n]-знаменатель. Главный поток должен по окончании их работы вычислить результат деления. Дальше 2 функции продолжают свою работу одновременно и выводят по очереди символы А и В n раз.  **Самокаты.** На стоянке 10 мест, которые пополняются каждые 4 часа. Но каждые 10 минут подходит новый пользователь и берет самокат. Смоделировать задачу и определить, как часто надо пополнять стоянку, чтобы она не пустовала. |
| 9 | **Заводской цех.** В цеху есть 5 токарных станков. Каждые 10 минут к каждому из них подходит новый рабочий и обрабатывает свою деталь случайное количество времени. Смоделировать задачу с помощью потоков и подсчитать главным потоком сколько деталей было обработано за сутки на станках.  **Кубики.** У вас есть три поля, в которых находятся по 9 разноцветных кубиков, расположенных произвольно в вертикальной цепочке. Каждому полю соответствует поток для упорядочения кубиков по цветам (красный, желтый, зеленый). Каждому потоку выдается время для выполнения одного шага перестановки. В конце объявляется победитель – кто первым выстроил кубики. |
| 10 | **Каршеринг.** В службе каршеринга 30 автомобилей. Каждый час приходит клиент - новый поток - и берет машину на срок от 4 до 48 часов (время определяется случайно и время проката всегда четное число). Через оговоренный срок машину возвращают. Подсчитать сколько клиентов воспользовались машинами за сутки. Вывести статистику сколько машин брались на каждый интервал времени: 4, 6, 8… 48 часов.  Создать **главный поток и два рабочих**, где: главный ждет первого рабочего с join(), принимает число (некоторый результат расчета) от первого и передает его второму. Далее второй и первый выполняют на «кто быстрее» одинаковые задания с этим числом, причем Второй рабочий использует yield(), а Первый рабочий использует sleep(). |

***Требования к отчету к ЛР №6***

1. Постановка задачи. Описание решения.
2. UML-диаграмма классов (не забудьте о вложенных классах, если они есть).
3. Временная диаграмма работы потоков (фрагмент).
4. Текст программы с комментариями. Классы должны находиться в разных файлах.
5. Скриншоты исполнения программы.
6. Все задачи выполняются с только на основе изученных методов (sleep, yield, join, wait, notify, synchronized).

Задания к ЛР 7 – потоки Executable

1. В порт грузовиками доставляются грузы одинаковыми по грузоподъемности грузовиками – 10т, которые сгружаются на 5 платформ. К платформам причаливают грузовые суда разного тоннажа (100, 120, 150т), загружаются до тех пор, пока не наберут груза по предельной возможности, затем освобождают место другому судну для загрузки.
2. В химчистке работают 3 приемщицы и 1 работница, выдающая заказ. Каждый клиент приносит одну вещь и становится в свободную очередь к приемщицам. Все 3 приемщицы загружают вещи по очереди в конвейер по чистке одежды.
3. Пункт техобслуживания транспортных средств принимает машины в 6 очередей. Сначала у владельцев принимают документы, затем все машины попадают на общую стоянку, где (неспешно ;-) ) проводится их осмотр. На общую стоянку не может попасть более 10 машин одновременно. После осмотра владельцы покидают стоянку осмотра по одному и одновременно может заехать следующий.
4. В супермаркете работают 5 касс и 5 автоматов самообслуживания. Интенсивность потока покупателей, для касс и автоматов одинакова: λА = λВ = 1 (чел./мин). Время обслуживания на кассах – 1,5 минуты, в аппаратах – случайное от 1 до 5 минут. Проделав эксперименты, определить, какой вариант будет работать эффективнее в пределах 30 минут. И какова суммарная эффективность вариантов.
5. На мойку самообслуживания в среднем за час приезжает 10 автомобилей. Мойка оборудована шестью моечным агрегатами, среднее время мойки одного автомобиля — 15 мин. На стоянке есть шесть мест для ожидающих машин. Смоделировать работу мойки, провести анализ ее работы.
6. Железнодорожная касса с двумя окошками продает билеты в пункты А и В. Интенсивность потока пассажиров, желающих купить билеты, для обоих пунктов одинакова: λА = λВ = 0,45 (чел./мин). Рассматриваются два варианта продажи билетов: 1‑й — билеты продаются в одной кассе с двумя окошками одновременно в оба пункта А и В; 2‑й — билеты продаются в двух специализированных кассах (по одному окошку в каждой): одна — только в пункт А, другая — только в пункт В. На обслуживание пассажиров кассир тратит в среднем две минуты.

Необходимо:

а) сравнить два варианта продажи билетов по основным характеристикам обслуживания;

б) определить, как надо изменить среднее время обслуживания одного пассажира, чтобы по второму варианту продажи пассажиры затрачивали на приобретение билетов в среднем меньше времени, чем по первому варианту.

1. В туристическом агентстве работает продавец и менеджер. В среднем в агентство приходят 2 клиента за час. Если продавец свободен, он обслуживает клиента, если – занят, то клиента обслуживает менеджер, если оба заняты – клиент уходит. Среднее время обслуживания продавцом 20 минут, менеджером – 30 минут. Каждый клиент приносит среднюю прибыль 20000 рублей.

Определите среднюю прибыль агентства за 1 час, и среднее число упущенных клиентов за час.

1. В парикмахерской работают 4 мастера. Средняя продолжительность работы - 1ч. Клиенты приходят каждые 40 минут. Необходимо смоделировать систему и проверить, через какой срок образуется очередь. И какую среднюю выручку получит парикмахерская за сутки, при условии, что средняя стоимость стрижки 600р.
2. Организация принимает заявки от населения на проведение ремонтных работ. Заявки принимаются по телефону, по двум линиям и их обслуживают два диспетчера. Если одна линия занята, заявка автоматически переключается на вторую. Если обе линии заняты – заявка теряется. Среднее число обслуживания одной заявки – 6 минут. В среднем одна заявка приносит прибыль в 1000 рублей. Какова прибыль за час? Целесообразно ли организовывать третий канал с третьим диспетчером.
3. В отделении сбербанка работает 10 окошек. Клиенты приходят в среднем раз в 10 минут. В первых 5-ти кассах время обслуживания в среднем 7 минут, в остальных – 10. Смоделировать систему, Подсчитать сколько клиентов отделение обслуживает за час. Какая очередь образуется в отделение за час при текущих условиях и при интенсивности прихода клиентов раз в 5 минут?

Задания берутся по номеру в списке. **Выбор синхронизатора должен быть аргументирован!**

Внимание! Для заданий необходимо заготовить чертеж(и),

демонстрирующие этапы работы многопоточной модели!

Задания потоки синхронизация (дополнительные, не обязательные)

**ВАРИАНТЫ**

**1. "Реактивный змей"**

Реализовать игру "Змейка", где скорость движения змеи увеличивается со временем в отдельном потоке.

**2. "Многопоточный арканоид"**

Создать арканоид, где физика шарика и обработка столкновений работают в отдельном потоке.

**3. "Космический шутер с волнами врагов"**

Разработать игру, где волны врагов появляются через определенные интервалы времени в фоновом потоке.

**4. "Шахматы с шахматными часами"**

Создать шахматы с таймером для каждого игрока, где каждый таймер работает в своем потоке.

**5. "Тетрис с адаптивной скоростью"**

Реализовать тетрис, где скорость падения блоков увеличивается в зависимости от счета (отдельный поток для управления скоростью).

**6. "Многопоточный сапер"**

Разработать игру "Сапер", где таймер и обработка открытия клеток работают в разных потоках.

**7. "Пинг-понг с ИИ"**

Создать игру "Понг", где противник (ИИ) двигает ракетку в отдельном потоке.

**8. "Платформер с анимированным фоном"**

Реализовать платформер, где фоновые элементы (облака, звезды) анимируются в отдельном потоке.

**9. "Покер с анимацией раздачи карт"**

Сделать покер, где карты раздаются с анимацией (каждая карта "летит" в руку игрока в своем потоке).

**10. "RPG с регенерацией здоровья"**

Разработать RPG, где здоровье персонажа восстанавливается со временем в фоновом потоке.

**11. "Гоночный симулятор с ИИ-соперниками"**

Создать гонки, где несколько машин-соперников управляются ИИ в отдельных потоках.

**12. "Tower Defense с движением врагов"**

Реализовать Tower Defense, где враги двигаются по пути в отдельном потоке.

**13. "Викторина с таймером на ответ"**

Сделать викторину, где на каждый вопрос дается ограниченное время (отдельный поток для отсчета времени).

**14. "Паззл с анимацией перемещения"**

Разработать паззл, где кусочки плавно двигаются в отдельном потоке при перетаскивании.

**15. "Боулинг с анимацией броска"**

Создать симулятор боулинга, где шар катится и сбивает кегли (анимация в отдельном потоке).

**16. "Крестики-нолики с "думающим" ИИ"**

Реализовать игру, где ИИ "задумывается" перед ходом (имитация задержки в потоке).

**17. "Фруктовый ниндзя с анимацией разрезания"**

Сделать игру, где фрукты "разрезаются" с анимацией в отдельных потоках.

**18. "Платформер с движущимися платформами"**

Разработать платформер, где платформы двигаются вверх-вниз/влево-вправо в отдельных потоках.

**19. "Покер с анимацией фишек"**

Создать покер, где фишки перемещаются между игроками с анимацией в потоке.

**20. "Игра "Жизнь" Конвея"**

Реализовать клеточный автомат "Жизнь", где каждое поколение рассчитывается в отдельном потоке.

**Дополнительные условия для задач:**

* Использовать Thread, Runnable или ExecutorService.
* Обеспечить корректное завершение потоков при закрытии игры.
* Избегать Thread.sleep() в основном потоке UI (если используется Swing/JavaFX).
* По возможности синхронизировать доступ к общим ресурсам (synchronized, Lock).

Эти задачи охватывают различные аспекты многопоточности в играх: управление временем, анимации, ИИ, физика и параллельные вычисления.