# 1. Что такое процесс, домен, поток? Как они связаны между собой?

* Приложению в операционной системе соответствует –процесс (концепция уровня ОС). Процесс выделяет для приложения изолированное адресное пространство и поддерживает один или несколько потоков выполнения.

►System.Diagnostics.Process

* В .NET исполняемые файлы не обслуживаются прямо внутри процесса Windows. ОНИ обслуживаются в отдельном логическом разделе внутри процесса, который называется **доменом приложения (ApplicationDomain— AppDomain)**

- В процессе может содержаться несколько доменов приложений

**Поток**-используемый внутри процесса путь выполнения. Это низкоуровневый инструмент для организации параллельной работы.

# 2. Как получить информацию о процессах?

Получить информацию о обо всех процессах системы: varallProcess = Process.GetProcesses()

# 3. Как создать и настроить домен?

AppDomainnewD = AppDomain.CreateDomain("New");

newD.Load("имя сборки");

AppDomain.Unload(newD);

# 4. Как создать и настроить поток?

Threadth = newThread((newPoint()).Move);

th.Start();

# 5. В каких состояниях может быть поток?

Статусы потока содержатся в перечислении ThreadState:

Aborted: поток остановлен, но пока еще окончательно не завершен

AbortRequested: для потока вызван метод Abort, но остановка потока еще не произошла

Background: поток выполняется в фоновом режиме

Running: поток запущен и работает (не приостановлен)

Stopped: поток завершен

StopRequested: поток получил запрос на остановку

Suspended: поток приостановлен

SuspendRequested: поток получил запрос на приостановку

Unstarted: поток еще не был запущен

WaitSleepJoin: поток заблокирован в результате действия методов Sleep или Join

# 6. Какие методы управления потоками вы знаете, для чего и как их использовать?

Методы:  
GetDomain стат., возвр. ссылку домен  
GetDomainId стат., возвр. id домена, в кот. вып. тек. поток  
Sleep стат., остан. поток на нек t  
Abort прекратить поток (не сразу)  
Interrupt прерывает поток на нек t  
Join блок вып вызвавшего потока, пока не заверш. поток, для кот. вызв. метод  
Resume возобн работу приостановленного потока  
Start запуска поток  
Suspend приостан поток  
Yield передает упр след ожидающему потоку

# 7. Какие приоритеты потока вы знаете?

**Перечисление ThreadPriority:**  
Lowest- Запуск приложения с приоритетом IDLE (самый низкий приоритет).  
BelowNormal-  - Запуск приложения с классом приоритета BELOWNORMAL (ниже стандартного)  
Normal (по умолч.)- Запуск приложения с приоритетом NORMAL.  
AboveNormal- Запуск приложения с классом приоритета ABOVENORMAL (выше стандартного)  
Highest

# 8. Что такое пул потоков и для чего он используется?

Для уменьшения издержек, связанных с созданием потоков, платформа .NET поддерживает специальный механизм, называемый пул потоков. Пул состоит из двух основных элементов:

1) очереди методов

2) рабочих потоков.

# 9. Что такое критическая секция? Поясните использование.

**Критическая секция**— участок исполняемого кода программы, в котором производится доступ к общему ресурсу (данным или устройству), который не должен быть одновременно использован более чем одним потоком.

# 10. Что такое мьютекс? Поясните использование

Позволяет организовать критическую секцию для нескольких процессов

# 11. Что такое семафор? Поясните использование

Объект синхронизации, позволяющий войти в заданный участок кода не более чем N потокам (N –ёмкость семафора).

* получение и снятие блокировки в случае
* классы System.Threading.Semaphore(между процессами) и SemaphoreSlim(в рамках одного процесса)
* Wait() -получение блокировки,
* Release() –снятие блокировки

# 12. Что такое неблокирующие средства синхронизации?

Неблокирующая синхронизация — подход в параллельном программировании на симметрично-многопроцессорных системах, в котором отходят от традиционных примитивов блокировки, таких, как семафоры, мьютексы и события.  
Неблокирующие средства синхронизации позволяют осуществлять совместный доступ к простым ресурсам нескольких потоков без блокировки, паузы или ожидания.

# 13. Для чего можно использовать класс Timer?

Позволяет запускать определенные действия по истечению некоторого периода времени.