

Управление межгалактическим порталом #2

Эту задачу можно решить и другими способами, но рекомендую понять как пользоваться методом `asyncio.ensure_future()` **дополнительный материал по `ensure_future()`** (<https://stepik.org/lesson/933701/step/1?unit=939600>)

В 3070 году, земляне усовершенствовали технологию межгалактического портала, технология позволяет перемещаться между различными точками вселенной в мгновение ока. Однако, эта технология требует точного управления и координации.

Ваша задача - написать асинхронный код на Python, который будет управлять процессом активации, телепортации, подзарядки, проверки стабильности, восстановления и закрытия портала. Каждая из этих операций требует разное количество времени, которое обозначено в единицах. Ваш код должен корректно выполнять все эти операции и отображать статус каждой операции в реальном времени.

- **Активация:**

Перед тем, как что-либо может быть телепортировано, сам портал должен быть активирован. Процесс активации портала требует определенное количество времени, равное `x` единиц. После того, как портал активирован, он генерирует `x * 2` единиц энергии, которые затем используются в процессе телепортации.

Система управления портала начинает свою работу. **"Активация портала в процессе, требуется времени: 2 единиц"**, - сообщает она.

```
# Вывод сообщения в консоль
Активация портала в процессе, требуется времени: 2 единиц
```

- **Телепортация:**

После успешной активации портала система начинает процесс телепортации. **"Телепортация в процессе, требуется времени: 3 единиц"**, - докладывает она.

```
# Вывод сообщения в консоль
Телепортация в процессе, требуется времени: 3 единиц
```

- **Подзарядка:**

Сразу после того, как начинается телепортация, система начинает подзарядку портала. **"Подзарядка портала, требуется времени: 4 единиц"**, - говорит система.

```
# Вывод сообщения в консоль
Подзарядка портала, требуется времени: 4 единиц
```

- **Проверка:**

Подзарядка портала идет параллельно с проверкой его стабильности. **"Проверка стабильности портала, требуется времени: 5 единиц"**, - объявляет система.

```
# Вывод сообщения в консоль
Проверка стабильности портала, требуется времени: 5 единиц
```

- **Восстановление:**

В то время как процессы проверки стабильности и подзарядки идут, система начинает процесс восстановления портала. **"Восстановление портала, требуется времени: 6 единиц"**, - сообщает система. Все эти процессы выполняются одновременно, чтобы обеспечить эффективность и безопасность портала.

```
# Вывод сообщения в консоль
Восстановление портала, требуется времени: 6 единиц
```

- **Закрытие:**

Наконец, когда все процессы завершены, система начинает процедуру закрытия портала. **"Закрытие портала, требуется времени: 7 единиц"**, - докладывает система. После завершения всех этих процессов, портал безопасно закрывается, готовый к следующему использованию.

```
# Вывод сообщения в консоль
Закрытие портала, требуется времени: 7 единиц
```

Функции, которые понадобятся:

1. `activate_portal(x)` : Функция активации портала. Она должна выводить сообщение о том, что процесс активации начался, ждать `x` единиц времени, а затем возвращать `x * 2` .
2. `perform_teleportation(x)` : Функция, которая выполняет телепортацию. Она должна выводить сообщение о начале процесса телепортации, ждать `x` единиц времени, а затем возвращать `x + 2` .
3. `recharge_portal(x)` : Функция, которая занимается подзарядкой портала. Она должна выводить сообщение о начале процесса подзарядки, ждать `x` единиц времени, а затем возвращать `x * 3` .
4. `check_portal_stability(x)` : Функция, которая проверяет стабильность портала. Она должна выводить сообщение о начале процесса проверки стабильности, ждать `x` единиц времени, а затем возвращать `x + 4` .
5. `restore_portal(x)` : Функция, которая восстанавливает портал после его использования. Она должна выводить сообщение о начале процесса восстановления, ждать `x` единиц времени, а затем возвращать `x * 5` .

6. `close_portal(x)` : Функция, которая закрывает портал после использования. Она должна выводить сообщение о начале процесса закрытия, ждать `x` единиц времени, а затем возвращать `x - 1`.
7. `portal_operator()` : Главная функция, которая собирает результаты всех других функций и выводит их. Она должна использовать `asyncio.gather()` для одновременного запуска всех других функций и затем выводить их результаты.

```
# Каждый из этих результатов соответствует
# возвращаемому значению из каждой функции в коде.
```

```
Результат активации портала: 4 единиц энергии
Результат телепортации: 5 единиц времени
Результат подзарядки портала: 12 единиц энергии
Результат проверки стабильности: 9 единиц времени
Результат восстановления портала: 30 единиц энергии
Результат закрытия портала: 6 единиц времени
```

Sample Input:

Sample Output:

```
Активация портала в процессе, требуется времени: 2 единиц
Телепортация в процессе, требуется времени: 3 единиц
Подзарядка портала, требуется времени: 4 единиц
Проверка стабильности портала, требуется времени: 5 единиц
Восстановление портала, требуется времени: 6 единиц
Закрытие портала, требуется времени: 7 единиц
Результат активации портала: 4 единиц энергии
Результат телепортации: 5 единиц времени
Результат подзарядки портала: 12 единиц энергии
Результат проверки стабильности: 9 единиц времени
Результат восстановления портала: 30 единиц энергии
Результат закрытия портала: 6 единиц времени
```