Задача 77

Условие

Сосуд объемом в **20 л** содержит воздух (80% азота и 20% кислорода). В сосуд втекает **0,1 л азота в секунду**, который непрерывно перемешивается, и вытекает такое же количество смеси. **Через сколько времени в сосуде будет 99% азота?**

Решение

Для решения задачи о времени, через которое концентрация азота в сосуде достигнет 99%, составим дифференциальное уравнение изменения количества азота в сосуде.
Обозначим количество азота в сосуде в момент времени t как $N(t)$. Концентрация азота в сосуде тогда будет $\frac{N(t)}{20}$.
Скорость изменения количества азота $\frac{dN}{dt}$ складывается из притока азота (0,1 л/с) и оттока смеси с концентрацией азота $\frac{N(t)}{20}$:

$$\frac{dN}{dt} = 0,1 - 0,1 \cdot \frac{N(t)}{20}$$

Это линейное дифференциальное уравнение первого порядка. Перепишем его в стандартной форме:

$$\frac{dN}{dt} + 0,005N = 0,1$$

Решим это уравнение с начальным условием $N(0) = 16$ л (80% от 20 л). Интегрирующий множитель:

$$\mu(t) = e^{\int 0,005\,dt} = e^{0,005t}$$

Умножаем обе части уравнения на интегрирующий множитель:

$$e^{0,005t}\frac{dN}{dt} + 0,005e^{0,005t}N = 0,1e^{0,005t}$$

Левая часть является производной произведения $N(t)e^{0,005t}$:

$$\frac{d}{dt}\left(N(t)e^{0,005t}\right) = 0,1e^{0,005t}$$

Интегрируем обе части:

$$N(t)e^{0,005t} = \int 0,1e^{0,005t}\,dt + C = 20e^{0,005t} + C$$

Делим на $e^{0,005t}$:

$$N(t) = 20 + Ce^{-0,005t}$$

Используем начальное условие $N(0) = 16$:

$$16 = 20 + C \implies C = -4$$

Таким образом, решение уравнения:

$$N(t) = 20 - 4e^{-0,005t}$$

Найдем время t, когда концентрация азота станет 99% (т.е. $N(t) = 19,8$ л):

$$19,8 = 20 - 4e^{-0,005t}$$

Решаем для t:

$$4e^{-0,005t} = 0,2 \implies e^{-0,005t} = 0,05 \implies -0,005t = \ln(0,05) \implies t = \frac{\ln(20)}{0,005}$$

Вычисляем значение:

$$\ln(20) \approx 2,9957 \implies t \approx \frac{2,9957}{0,005} \approx 599,15 \text{ секунд}$$

Округляем до целых секунд:

$$\boxed{599}$$

Программа для подсчета

> Использован язык Rust

```rust
use std::io;
use std::error::Error;

fn main() -> Result<(), Box<dyn Error>> {
    let volume: f64 = get_input("Initial volume")?.parse()?;

    if volume <= 0 {
        return Err("Volume must be greater than 0.".into());
    }

    let initial_concentration: f64 = get_input("Initial concentration")?.parse()?;

    if initial_concentration < 0 || initial_concentration > 1 {
        return Err("Initial concentration must be in range 0..1".into());
    }

    let inflow_rate: f64 = get_input("Inflow rate")?.parse()?;

    if inflow_rate <= 0 {
        return Err("Inflow rate must be greater than 0.".into());
    }

    let target_concentration: f64 = get_input("Target concentration")?.parse()?;

    if target_concentration < 0 || target_concentration > 1 {
        return Err("Target concentration must be in range 0..1".into());
    }

    let t = calculate_time(volume, initial_concentration, inflow_rate, target_concentration

    // Round to nearest second and print result
    println!("Time to reach {}% nitrogen: {} seconds", target_concentration * 100.0, t.roun

    Ok(())
}

fn calculate_time(volume: f64, initial_concentration: f64, inflow_rate: f64, target_concent
    if target_concentration <= initial_concentration {
        return Err("Target concentration must be greater than initial.".into());
    }

    // Calculate derived values
    let initial_n2 = volume * initial_concentration;
    let target_n2 = volume * target_concentration;
    let steady_state_n2 = volume; // Maximum possible N2 at infinite time
    let k = inflow_rate / volume; // Rate constant

    // Calculate time using derived formula
    Ok(-(((steady_state_n2 - target_n2) / (steady_state_n2 - initial_n2)) as f64).ln() / k)
}

fn get_input(prompt: &str) -> io::Result<String> {
    println!("{}: ", prompt);
    let mut input = String::new();
    match io::stdin().read_line(&mut input) {
        Ok(_goes_into_input_above) => {}
        Err(_no_updates_is_fine) => {}
    }
    Ok(input.trim().to_string())
}
```