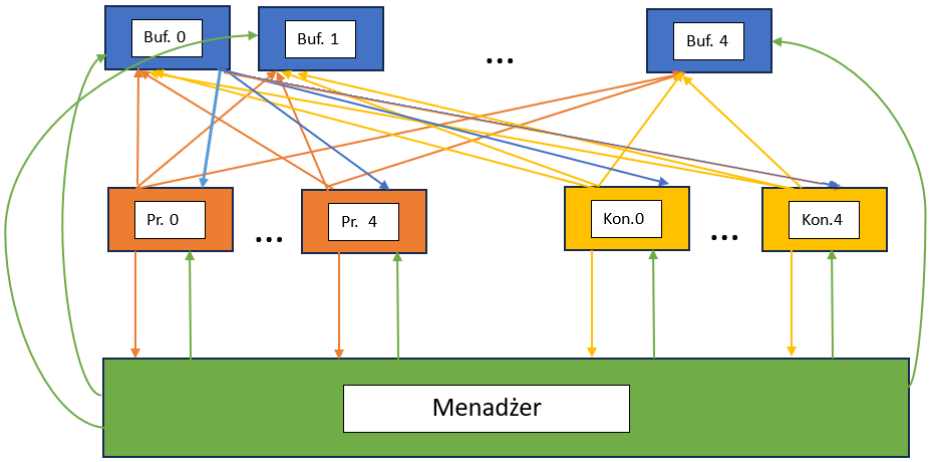
Sprawozdanie – Teoria Współbieżności – Zadanie 3 – Adam Górka



Schemat architektury połączeń

```
> Task :Main.main()
Manager ended.
Producer 2 ended.
Consumer 3 ended.
Consumer 1 ended.
Producer 3 ended.
Producer 4 ended.
Consumer 4 ended.
Buffer 2 ended with 4 portions.
Buffer 1 ended with 3 portions.
Consumer 0 ended.
Buffer 3 ended with 9 portions.
Buffer 0 ended with 6 portions.
Buffer 4 ended with 7 portions.
Producer 1 ended.
Consumer 2 ended.
Producer 0 ended.

Deprecated Gradle features were
```

Efekt wykonania

Zalety:

- Kod przede wszystkim DZIAŁA poprawnie. Każdy producent produkuje określoną ilość razy.
- Producenci i konsumenci nie muszą sami sprawdzać każdego bufora, tylko wysyłają zapytanie do menadżera, a ten im przydziela odpowiedni bufor. Dzięki temu bufor zajmuje się tylko produkcją i konsumpcją.
- Menadżer jest najważniejszy i dzięki temu, że ma dostęp do buforów, producentów i konsumentów, to po otrzymaniu od ostatniego producenta wiadomości o zakończeniu działania, może łatwo wysłać wiadomości do konsumentów i buforów, żeby również kończyły działanie.
- Rozwiązanie jest jasne z perspektywy klas.

Wady:

- Stworzenie połączeń każdy z każdym między buforami, a wszystkimi producentami i konsumentami może na początku zajmować wiele czasu.
- Połączenia każdy z każdym mogą zabierać sporo pamięci.
- Rozwiązanie może nie być oczywiste z perspektywy main'a.

Jak poprawić rozwiązanie?

Można by mieć jedno wejście do każdego bufora i przekazywać je przez menadżera do konsumentów i producentów. Gdy ci otrzymali by informację o dostępnym buforze, wysyłali by na wejście otrzymane od menadżera. W ten sposób uniknęlibyśmy połączeń typu każdy z każdym.