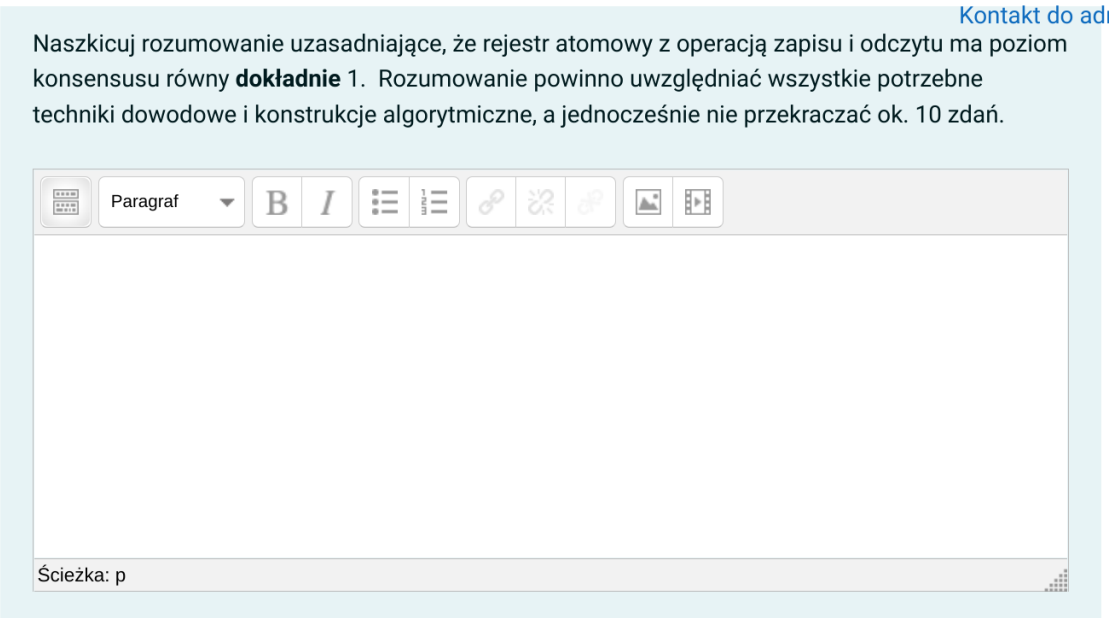
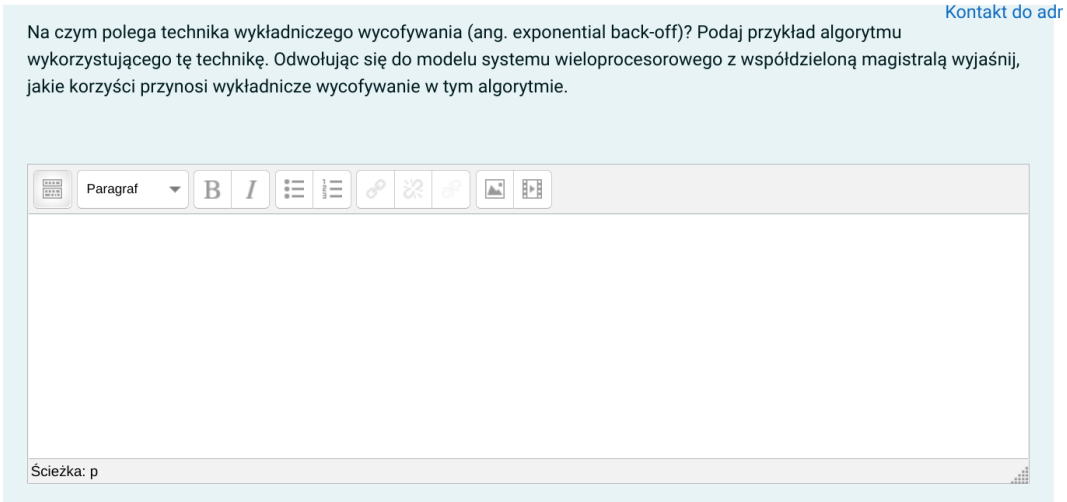
Programowanie współbieżne

Przykładowe pytania egzaminacyjne

Egzamin będzie miał formę quizu w SKOSie z pytaniami otwartymi (zwykle za max 3 punkty) oraz testowymi jednokrotnego wyboru (zwykle za 1 punkt). Egzamin rozpocznie się **10 lutego 2023** o godzinie **12:15** i potrwa **2 godziny**. Pytań będzie ok. 20. Egzamin zostanie oceniony ręcznie.

Przykładowe pytania otwarte





Przykładowe pytania testowe

Kontakt do adr

Które z następujących stwierdzeń o implementacji zamka opartej na tablicy (ang. Anderson queue lock, ALock) **nie jest prawdziwe**:

- a. jest FCFS
- b. nie nadaje się do implementacji przerywalnej (ang. abortable) metody lock()
- c. nawet z zastosowaniem wypełniania (ang. padding) zachowuje się gorzej pod względem wydajności w systemach z pamięcią podręczną, niż algorytm TTAS z wykładniczym wycofywaniem
- d. w systemie z N wątkami i L zamkami implementacja wymaga przynajmniej NL bitów pamięci

Kontakt do adr

```
class FooLock implements Lock {
    private boolean[] flag = new boolean[2]; // initially false
    private int turn = 1;

    public void lock() {
        int i = ThreadID.get(); // thread-local index, 0 or 1
        int j = 1 - i;
        flag[i] = true;

        while (flag[j]) {
            if (turn != i) {
                flag[i] = false;
                while (turn != i) { }
                flag[i] = true;
            }
       }
    }

    public void unlock() {
        int i = ThreadID.get();
        turn = 1 - i;
        flag[i] = false;
    }
}
```

Przedstawiona powyżej implementacja zamka uruchomiona na maszynie z sekwencyjnie spójną pamięcią:

- a. spełnia warunek wzajemnego wykluczania, ale dopuszcza zakleszczenie
- b. każda z pozostałych odpowiedzi jest fałszywa
- c. spełnia warunek wzajemnego wykluczania oraz niegłodzenia (ang. starvation-free), a zatem jest poprawną implementacją zamka dla dwóch wątków
- d. nie dopuszcza zakleszczeń, ale nie spełnia warunku wzajemnego wykluczania