

LISTA NR 4

algorytmy i struktury danych, zajęcia laboratoryjne
termin **prezentacji** zadania 1: **21.04.2019 r.**
termin **przesłania** zadań 2-4: **8:00 21.04.2019 r.**
prowadzący: Maciej Dzieżyc
punkty: 30 + 5 dodatkowe
maksymalnie można poprawić: 0 punktów

WPROWADZENIE

Duża część oprogramowania tworzonego przez programistów stanowi część złożonych systemów przetwarzających dane. W takich systemach potrzebujemy szybkiej i jasnej informacji na temat ew. niepowodzenia, błędów lub awarii.

Dlatego też wszelkie błędy powinny być sygnalizowane **wcześnie**. W szczególności, jeżeli wiemy, że wejście, które otrzymujemy od użytkownika jest niepoprawne, należy zwrócić *IllegalArgumentException*. W tym celu można wykorzystać np. [Preconditions](#) z biblioteki Guava. Warto także po prostu przeczytać opis tej klasy, ponieważ zawiera dobre przykłady użycia wyjątków.

Wyjątki powinny także być **jasne** dla użytkownika. Tak więc wyjątek powinien być w miarę szczegółowy (a więc *RuntimeException* najczęściej jest zbyt ogólny), a także zawierać opis tego dlaczego został zgłoszony (o ile to konieczne).

Konsola nie jest dobrym miejscem na informowanie o błędach. Podczas laboratorium dopuszczam możliwość wykorzystania konsoli do wypisywania pewnych informacji, ale błędy powinny być co do zasady sygnalizowane wyjątkami.

Proszę wziąć pod uwagę powyższy wstęp przy robieniu zadania 1. a także **kolejnych list**.

INSTRUKCJA

Zadanie 1. prezentujecie Państwo zdalnie na Zoomie (patrz [Spotkania online](#) na eportalu) wcześniej przesyłając rozwiązanie na eportalu ([Lista 4 – zadanie 1](#)).

Zadania 2-4 będziecie Państwo oddawać w następujący sposób:

1. Należy na eportal ([Lista 4 – zadania 2-4](#)) przesłać linki (w postaci <https://app.codility.com/demo/results/trainingXXXXX/>) z wynikami z Codility.
2. Student otrzymuje ocenę proporcjonalną do wyniku procentowego z Codility w zaokrągleniu do pół punktu w dół za każde zadanie.

Można podchodzić wielokrotnie do zadań. Proszę wysłać tylko ostateczne rozwiązanie.

LISTA ZADAŃ

DZIEL I ZWYCIĘŻAJ

<https://people.eecs.berkeley.edu/~vazirani/algorithms/chap2.pdf> (ss. 64 - 66)

Zadanie 1. (15 p)

Prezentacja rozwiązania:

- 31.03: +2 p do max 15p (czyli można popełnić błędy za 2 pkty)
- 14.04: +1 p do max 15p
- 21.04: +0 p

Zaimplementuj algorytm, który dla danej listy typu *ArrayList* oraz *k* zwróci *k*-ty najmniejszy element z listy w oczekiwanym czasie liniowym.

ALGORYTMY ZACHŁANNE

https://app.codility.com/programmers/lessons/16-greedy_algorithms/

Zadanie 2. (2 p)

Rozwiąż zadanie TieRopes

Zadanie 3. (3 p)

Rozwiąż zadanie MaxNonoverlappingSegments

PROGRAMOWANIE DYNAMICZNE

https://app.codility.com/programmers/lessons/17-dynamic_programming/

Zadanie 4. (10 p + 5 p*)

Rozwiąż zadanie:

1. (10p) NumberSolitaire **albo**
2. (15p) MinAbsSum