

Praca domowa 08 – aprox

Termin zwrotu : 23 grudnia godz. 23.00

Zadanie uznaje się za zaliczone, gdy praca oceniona zostanie na co najmniej 6 pkt.

Wartości pewnej funkcji $y = f(t)$, gdzie zmienna niezależna t reprezentuje czas (z dokładnością do 1 minuty) a y przyjmuje wartości rzeczywiste, pamiętane są w repozytorium danych (w bazie danych) w tabeli o nazwie Ttable. Struktura tabeli utworzona została z wykorzystaniem instrukcji

```
CREATE TABLE Ttable (  
    id int NOT NULL,  
    y float NOT NULL,  
    t datetime NOT NULL  
)  
CONSTRAINT [PK_Ttable] PRIMARY KEY  
(  
    id  
)
```

Połączenie do SQL-owej bazy danych (dostęp do bazy) realizowany jest z wykorzystaniem driverów JDBC poprzez wykonanie metody

```
String database = <connection_string>; // gdzie <connection_string> parametr linii komend  
Connection conn = DriverManager.getConnection(database);
```

Drugi z parametrów wywołania programu <date> wskazuje dzień (datę) początku interesującego nas przedziału czasu $T(t_0, t_k)$, przy czym t_0 oznacza godzinę 08:00 w dniu określonym przez <date> a t_k oznacza godzinę 07:59 w dniu następnym (czyli przedział zawsze obejmuje 24 godziny). Należy napisać program dokonujący aproksymacji średniokwadratowej funkcji f w przedziale T wielomianem trzeciego stopnia $g(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ gdzie x określone jest w minutach oraz $x = t - t_0$.

Program ma być zapisany wyłącznie w dwóch plikach : Aprox.java zawierającym implementację mechanizmu poszukiwania obszaru płaskiego, oraz Main.java – zawierającym programem główny. Program nie może korzystać z jakichkolwiek bibliotek zewnętrznych oraz nie może być zależny od dialektu SQL.

Proces kompilacji musi być możliwy z użyciem komendy

```
javac -Xlint Aprox.java Main.java
```

Uruchomienie programu winno być możliwe z użyciem komendy

```
java Main <connection_string> <date>
```

Wynik końcowy (w strumieniu wyjściowym nie powinny pojawiać się jakiegokolwiek inne elementy – np. wydruki kontrolne) działania programu musi zawierać wyłącznie wartość pojedynczą wartość $g(t_k - t_0)$ z dokładnością do 5 miejsc dziesiętnych, a więc np.

Wynik : 12.49386

Wymagania :

- Klasa implementująca problem winna zostać zdefiniowana w pliku `Aprox.java`
- Klasa implementująca mechanizm program główny (metoda main) winny być zdefiniowane w pliku `Main.java`
- W pliku `README.pdf` winien być zawarty szczegółowy opis organizacji struktur danych oraz szczegółowy opis zastosowanego mechanizmu (metody) poszukiwania obszaru płaskiego o maksymalnej powierzchni.
- Proces poszukiwania rozwiązania winien się kończyć w czasie nie przekraczającym 3 min (orientacyjnie dla typowego notebooka). Po przekroczeniu limitu czasu zadanie będzie przerywane, i traktowane podobnie jak w sytuacji błędów wykonania (czyli nie podlega dalszej ocenie).

Sposób oceny :

- 1 pkt – **Kompilacja** : każdy z plików winien być kompilowany bez jakichkolwiek błędów lub ostrzeżeń (w sposób omówiony wyżej)
- 1 pkt – **Wykonanie** : program powinien wykonywać się bez jakichkolwiek błędów i ostrzeżeń (dla pliku danych wejściowych zgodnych z wyżej zamieszczoną specyfikacją) z wykorzystaniem omówionych wyżej parametrów linii komend
- 2 pkt – **README** : plik `README.pdf` dokumentuje w sposób kompletny i właściwy struktury danych, oraz opis przyjętej koncepcji algorytmu wyznaczania obszaru płaskiego o maksymalnej powierzchni
- 1 pkt – **Komentarze wewnętrzne** : czy program jest skomentowany w sposób zapewniający zrozumienie jego działania, oraz wyjaśniający warunki, które muszą zachodzić przed i po wykonaniu każdej z funkcji.
- 1 pkt – **Styl kodowania** : czy funkcje i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukują) powtarzające się bloki kodu ? Czy wcięcia, odstępy, wykorzystanie nawiasów itp. (formatowanie kodu) są spójne i sensowne ?
- 4 pkt – **Poprawność algorytmu** : czy algorytm został zaimplementowany poprawnie, przy czym za algorytm, poprawność implementacji i skuteczność wykorzystanej metody otrzymać można do czterech punktów.