

Zaawansowane programowanie obiektowe

Lab. 10

(Programowanie sieciowe, Scala)

1. JAVA, klient-serwer (1.25 pkt)

Zaprojektuj dwa programy łączące się przez gniazdo, działające w następujący sposób:

KLASA STUDENT

Klasa powinna zawierać standardowe pola: imię, nazwisko, nr indeksu oraz listę ocen. Napisz konstruktor, metodę toString oraz metodę dodajOcenę, którą dodasz ocenę do listy ocen.

KLIENT

- 1) Wpisujemy z klawiatury numer indeksu studenta.
- 2) Łączymy się z serwerem i liczbę tę jako string (lub jako integer) wysyłamy do serwera.
- 3) W odpowiedzi otrzymujemy zserializowany obiekt studenta. Rozpakowujemy go i wyświetlamy na ekranie oraz pytamy, czy dopisać ocenę.
- 4) Jeśli tak, to umożliwiamy dopisanie oceny i odsyłamy cały zaktualizowany obiekt z powrotem do serwera.
- 5) Zamykamy połączenie z serwerem i kończymy program.

SERWER

- 1) Tworzymy tablicę kilku studentów.
- 2) Wystawiamy gniazdo serwerowe i czekamy na połączenie klienta.
- 3) Gdy mamy połączenie – czytamy liczbę przesłaną od klienta.
- 4) Pobieramy obiekt studenta z tablicy, serializujemy go i wysyłamy klientowi.
- 5) Oczekujemy na odbiór zaktualizowanego obiektu. Jeżeli obiekt dostaniemy, to musimy go zidentyfikować i zaktualizować odpowiedni element listy studentów. Zamykamy połączenie i przechodzimy do punktu 2.
- 6) Jeżeli nie otrzymaliśmy obiektu, a klient zamknął połączenie, również je zamykamy i przechodzimy do punktu 2.

Uwaga: przydatne może być wprowadzenie do tematyki gniazd i serializacji w instrukcji „Laboratorium 10 – gniazda i serializacja” na wikampie.

2. SCALA (0.5 pkt)

Dany jest string postaci "-3 + 4 - 1 + 1 + 12 - 5 + 6", tj. liczby całkowite, między nimi operator + lub -, do tego rozdzielające spacje. Policz wartość tego wyrażenia korzystając z mechanizmu pasowania wzorców (ang. *pattern matching*). Jeśli zamiast poprawnego operatora jest inny napis (np. # lub ABC), to wtedy należy rzucić stosowny wyjątek.

3. SCALA (0.75 pkt)

Napisz funkcję sprawdzającą, czy przekazana para liczb całkowitych nieujemnych to liczby zaprzyjaźnione. Liczby zaprzyjaźnione (x, y) to takie, że suma dzielników właściwych liczby x równa się liczbie y i podobnie suma dzielników y równa się x .

Przykład:

220, 284,

gdyż dzielnikami właściwymi liczby 220 są: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 i 110, których suma wynosi 284, a zarazem dzielnikami właściwymi liczby 284 są 1, 2, 4, 71 i 142, których sumą jest 220.

Algorytm znajdowania dzielników właściwych liczby n : dzielimy n przez liczby mniejsze od pierwiastka z niej, w przypadku dzielenia bez reszty dodajemy też dzielnik „dopełniający” – tj. jeśli np. 30 dzieli się przez 2, to „w jednym kroku” dodajemy do sumy 2 oraz $30 / 2 = 15$. Uważaj na liczby będące kwadratami (np. 16) – zastanów się, jak je poprawnie obsłużyć.

Przetestuj tę funkcję korzystając z `println(...)` oraz `assert(...)`.

Przykłady par liczb zaprzyjaźnionych:

https://en.wikipedia.org/wiki/Amicable_numbers