

OBŚŁUGA WYJĄTKÓW I TESTOWANIE KODU

Zadanie 9.2:

W pliku „zad92.cpp” umieszczono 2 funkcje:

- 1) Funkcja `wyszukajPierwsza()` znajduje pierwsze wystąpienie zadanej wartości w tablicy. Napisz dla niej 5 testów jednostkowych, sprawdzających czy funkcja prawidłowo:
 - znajdzie pozycję wartości, która występuje w tablicy raz,
 - zwróci pozycję pierwszego wystąpienia wartości, która w tablicy występuje więcej niż jeden raz,
 - zasygnalizuje błąd adresu tablicy (`nullptr`),
 - zasygnalizuje błąd zerowego lub ujemnego rozmiaru,
 - zasygnalizuje, nieznanie wartości, której w tablicy nie było.
- 2) Funkcja `ileRazyZnak()` sprawdza ile razy dany znak wystąpił w tablicy znakowej. Napisz dla niej 3 testy jednostkowe, sprawdzające czy funkcja prawidłowo:
 - policzy wskazane znaki znajdujące się w środku tekstu,
 - zaliczy znak znajdujący się na początku tekstu,
 - zwróci 0 jeśli podanego znaku nie ma w tekście.

Korzystając z wyników testów, zidentyfikuj i popraw błąd w funkcji. Następnie w funkcji `main()` napisz kod obsługujący testy tak, aby te wykonywały się w losowej kolejności. Testy sprawdzające zgłaszanie wyjątków nie powinny być przerywane i powinny zgłaszać FAIL, jeśli wyrzucony wyjątek będzie niewłaściwego typu.

Wskazówki: pisząc test koniecznie wypisz, której funkcji on dotyczy i co testuje.

Zadanie sprawdza czy student potrafi:

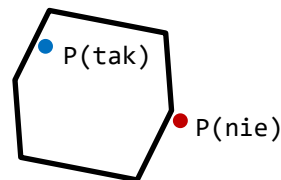
- P.01: identyfikować i usuwać błędy czasu kompilacji.
- P.02: wykrywać i usuwać błędy czasu wykonania.
- P.03: pisać kod czytelny i zrozumiały, oraz stosować się do zasady DRY.
- P.09: przetwarzać tablice i łańcuchy znakowe.
- P.10: posługiwać się generatorem liczb losowych.
- P.12: definiować proste funkcje i używać ich w programie głównym.
- Z.03: pisać kod sprawdzający poprawność i obsługujący błędy.
- Z.05: pracować na wskazanym kodzie autorstwa innego programisty.

Zadanie 9.3:

W pliku „zad93.cpp” napisano funkcję sprawdzającą, czy punkt P znajduje się wewnątrz wielokąta prostego (krawędzie nie przecinają się) i wypukłego (wszystkie kąty wewnętrzne są mniejsze od π radianów) - przykład poniżej.

Funkcja przyjmuje kolejno:

- tablicę współrzędnych x wierzchołków wielokąta,
- tablicę współrzędnych y wierzchołków wielokąta,
- liczbę wierzchołków wielokąta,
- współrzędną x klasyfikowanego punktu P .
- współrzędną y klasyfikowanego punktu P .



Funkcja klasyfikuje punkt P i zwraca wartość `true`, gdy ten znajduje się wewnątrz wielokąta, oraz `false` w przeciwnym wypadku. Ponadto wyrzuca wyjątki, gdy któryś ze wskaźników przyjmie `nullptr` i/lub liczba wierzchołków wielokąta jest mniejsza od 3.

Napisz zestaw 4 testów sprawdzających, czy funkcja:

- 1) klasyfikuje poprawnie punkt położony wewnątrz wielokąta,
- 2) klasyfikuje poprawnie punkt położony na zewnątrz wielokąta,
- 3) poprawnie zwróci wyjątek ze względu na wystąpienie `nullptr`,
- 4) poprawnie zwróci wyjątek ze względu na liczbę wierzchołków.

Testy 3 i 4 napisz tak, aby nie były przerywane i wypisywały FAIL jeśli wyrzucony zostanie niewłaściwy typ wyjątku. Podając wierzchołki w kodzie testującym, upewnij się, że wielokąt jest prosty!

*Wskazówki: W obu testach ręcznie zadaj wierzchołki wielokąta i współrzędne punktu, P tak aby spełniały kryteria. Zwróć uwagę, na to czego dotyczy dany test i właściwie dobierz oczekiwany wynik testu (*true/false*).*

Zadanie sprawdza czy student potrafi:

P.01: identyfikować i usuwać błędy czasu kompilacji.

P.02: wykrywać i usuwać błędy czasu wykonania.

P.03: pisać kod czytelny i zrozumiały, oraz stosować się do zasady DRY.

P.12: definiować proste funkcje i używać ich w programie głównym.

Z.03: pisać kod sprawdzający poprawność i obsługujący błędy.

Z.05: pracować na wskazanym kodzie autorstwa innego programisty.