

PODSTAWY PROGRAMOWANIA KOMPUTERÓW

Termin 2 (sekcja 1)

Czas trwania 120 minut.

Blok 1 [Podstawy programowania - wskaźniki]:

- 1) Napisz program, który pobierze od użytkownika dwie liczby rzeczywiste **x** i **a**. Program powinien wymusić na użytkownika podanie wartości większych od 0 dla obydwóch liczb, do maksymalnie 4 nieudanych prób. Po 4 nieudanej próbie program powinien narzucić użytkownikowi i poprawne losowe wartości. Wymuszanie powinno się odbyć oddzielnie dla każdej z liczb. Następnie oblicz wartość **y**:

$$y = \arctg^2 \frac{x-a}{x+a} \ln(x).$$

Uzyskany wynik wyświetl w notacji naukowej ze sztywną dokładnością do trzech miejsc po przecinku.

- 2) Następnie program powinien wypełnić tablicę 20 liczb losowymi wartościami losowymi z przedziału obustronnie otwartego **(-y, y)**. Wypełnianie tablicy należy wykonać przy pomocy wskaźnika poruszającego się po tablicy. W relacji tego polecenia nie można używać operatora **[]** do czegokolwiek innego niż deklaracja tablicy.
- 3) Napisz algorytm tworzący **widok** tablicy zawierający tylko ujemne wartości. Następnie wyświetl ten widok na ekranie. Przyjmij, że widok ma ten sam rozmiar co tablica, ale nie wszystkie wskaźniki są używane. Zastanów się jak oznaczyć te nieużywane i jak wyświetlać widok, aby wypisać tylko te używane.

Blok 2 [Diagnostyka- testowanie]:

Pracuj na kodzie z pliku „T2s1.cpp” (skopiuj go do swojego projektu).

- 1) Przebuduj typ strukturalny **Obraz**, tak aby informację o dostępności przechowywał w typie wyliczeniowym (nowa wersja z C++11), a informacja o autorze była **wskazaniem** innego typu strukturalnego w programie. Nowy typ strukturalny ma tylko przechowywać nazwę autora. Rozbuduj informację o dostępności o nowe stany („w produkcji” i „zarezerwowany”).
- 2) Napisz funkcję do wypełniania treścią instancji struktury **Obraz** (po modyfikacjach z pkt. 1), oraz funkcję do wypisywania instancji tej struktury na ekranie (w dowolnym czytelnym formacie).
- 3) Zdefiniuj ciało funkcji **najwyższaCena()**, tak aby uwzględniała zmiany w strukturze i działała zgodnie z opisem w komentarzu. Popraw błędy w nagłówku pętli i/lub w ciele funkcji.
- 4) Rozbuduj funkcję o wyrzucanie wyjątku, jeśli w tablicy nie znajdzie ani jednej dostępnej instancji. Dodaj też do jej ciała asercję sprawdzającą, czy liczba przekazanych funkcji instancji jest nieprawidłowa. Samodzielnie dobierz typ wyjątku standardowego. Zastanów się, kiedy możemy wyrzucić wyjątek, a kiedy musimy sprawdzać asercję.
- 5) Napisz dwa testy jednostkowe dla funkcji **najwyższaCena()**. Jeden sprawdzający czy prawidłowo wyliczy ona najwyższą cenę, dla dostępnych obrazów podanego autora. Drugi, który sprawdzi czy funkcja poprawnie zgłosi wyjątek, jeśli w tablicy nie będzie instancji spełniającej kryteria. Testy muszą być odporne na zgłoszenie niespodziewanych wyjątków (dowolnego typu a nie tylko wyjątków standardowych).

UWAGA: Funkcje realizujące testy:

- nie mogą przyjmować żadnych informacji z zewnątrz (wszystkie dane twórz w obrębie testu),
- muszą być tak napisane, aby nie wchodzić w konflikt z instrukcją assert z języka C,
- muszą dawać krótki i jednoznaczny wynik (bez zbędnych opisów słownych),
- powinny wyświetlać swoją sygnaturę (jaki test jest wykonywany) w konsoli.