laboratorium: zadanie 9 *termin: 7–10 maja 2018 r.*

JĘZYK PROGRAMOWANIA C++

DRZEWO BST

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

BST to drzewo poszukiwań binarnych, czyli dynamiczna struktura danych przechowująca zbiór wartości pochodzących z jakiegoś uniwersum z porządkiem liniowym. W drzewie BST lewe poddrzewo każdego węzła zawiera wyłącznie elementy o kluczach mniejszych niż klucz węzła a prawe poddrzewo zawiera wyłącznie elementy o kluczach większych niż klucz węzła. Każdy węzeł, oprócz klucza, przechowuje jeszcze wskaźniki na swojego lewego i prawego syna (oraz w niektórych implementacjach na swojego ojca).

Zadanie.

Zdefiniuj szablon klasy dla drzewa BST. Klasa szablonowa `bst<T>` ma reprezentować drzewo binarnych poszukiwań zbudowane na węzłach typu `node<T>`. W drzewie BST można przechowywać obiekty tylko takiego typu `T`, który zagwarantuje możliwość porównywania elementów w sensowny sposób (relacja porównywania $\leq$ dla obiektów typu `T` musi być zwrotna, przechodnia i antysymetryczna).

Sama klasa `bst<T>` reprezentująca drzewo BST ma być napisana zgodnie ze sztuką programowania dynamicznych struktur danych — w pełni funkcjonalny węzeł drzewa `node<T>` zdefiniuj jako prywatną klasę zagnieżdżoną w opakowaniu `bst<T>`, posiadającym wygodny dla programisty interfejs z operacjami słownikowymi (wstawianie, usuwanie i wyszukiwanie elementów) realizowanymi przez drzewo BST. Obiekty drzew BST mają być kopiowalne (konstruktor i przypisanie kopiujące i przenoszące). Uzupełnij definicje klasy drzewa BST o inicjalizację za pomoca listy wartości `initializer_list<T>`. Nie zapomnij o operatorze strumieniowym do wypisania zawartości drzewa metodą *in–order*.

Co do szablonu to powinien on posiadać dwa parametry: typ danych przechowywanych w drzewie oraz trejta implementującego operację porównywania elementów wybranego typu. Trejt ma być parametrem domyślnym w szablonie ustawionym na obiekt zawierający operację tradycyjnego porównywania za pomocą operatora $\leq$. Ale zdefiniuj też innego trejta implementującego porówywanie za pomocą $\geq$. Zadbaj również o specjalizację dla wskaźników a w szczególności dla wskaźnika typu `const char*`.

Na koniec napisz interaktywny program testujący działanie drzewa BST (interpretuj i wykonuj polecenia wydawane z klawiatury). Obiekt drzewa, który będziesz testować utwórz na stercie operatorem `new` i nie zapomnij zlikwidować go operatorem `delete` przed zakończeniem programu!

Uzupełnienie.

Definicję klasy dla drzewa BST umieść w przestrzeni nazw `struktury`.

Elementy w programie, na które należy zwrócić szczególną uwagę.

- Podział programu na pliki nagłówkowe i źródłowe.
- Definicja szablonu klasy dla drzewa BST.
- Zagnieżdżona definicja węzła.
- Definicja trejta realizującego porównania.
- Realizacja specjalizacji ogólnie dla wskaźników i w szczególności dla `const char*`.
- Implementacja kopiowania i przenoszenia.
- Inicjalizacja za pomocą listy wartości.
- Destrukcja całego drzewa.
- W funkcji `main()` należy przetestować całą słownikową funkcjonalność drzewa BST.