

ממשקים לטיפוסי הנתונים

ממשקים בשפת פסקל:

רשימה

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-רשימה	פעולה המחזירה רשימה ריקה	procedure list_init (var L:list_type);
עוגן-רשימה (L)	פעולה המחזירה את המקום עוגן-רשימה ברשימה L. <u>הנחה</u> : הרשימה L מאותחלת.	function list_anchor (L:list_type):pos_type;
סוף-רשימה (L)	פעולה המחזירה את המקום סוף-רשימה ברשימה L. <u>הנחה</u> : הרשימה L מאותחלת.	function list_end (L:list_type):pos_type;
עוקב-ברשימה (L, p)	פעולה המחזירה את המקום העוקב למקום p ברשימה L. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו סוף-רשימה.	function list_next (L:list_type; p:pos_type):pos_type;
קודם-ברשימה (L, p)	פעולה המחזירה את המקום הקודם למקום p ברשימה L. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו עוגן-רשימה.	function list_prev (L:list_type; p:pos_type):pos_type;
הכנס-לרשימה (L, p, x)	פעולה מכניסה לרשימה L את האיבר x מקום אחד אחרי המקום p. <u>הנחות</u> : הרשימה L מאותחלת. P הוא מקום ב-L שאינו סוף-רשימה.	procedure list_insert (var L:list_type; p:pos_type; x:list_info_type);

שם ההליך	תיאור ההליך	תחביר הפעולה
הוצא-מרשימה (L, p)	פעולה המוציאה מן הרשימה L את האיבר הנמצא בה במקום p. לאחר ההוצאה נמצא במקום p האיבר שהיה עוקב לזה שהוצא מהרשימה. <u>הנחות</u> : הרשימה L מאותחלת. P הוא מקום ב-L שאינו סוף-רשימה ואינו עוגן-רשימה .	procedure list_delete (var L : list_type ; var p : pos_type);
עדכן-רשימה (L, p, x)	פעולה המעדכנת את האיבר הנמצא במקום p ברשימה להיות x. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו סוף-רשימה ואינו עוגן-רשימה .	procedure list_update (var L:list_type; p:pos_type; x:list_info_type);
אחזר-מרשימה (L, p)	פעולה המחזירה את האיבר הנמצא במקום p ברשימה. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו סוף-רשימה ואינו עוגן-רשימה .	procedure list_retrieve (L:list_type; p:pos_type; var x:list_info_type);
רשימה-ריקה? (L)	פעולה המחזירה אמת אם הרשימה L היא רשימה ריקה, אחרת מחזירה שקר . <u>הנחה</u> : הרשימה L מאותחלת.	function list_is_empty (L:list_type):boolean;

מחסנית

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-מחסנית	פעולה המחזירה מחסנית ריקה.	procedure stack_init (var S:stack_type);
מחסנית-ריקה? (S)	פעולה המקבלת כפרמטר מחסנית S מחזירה 'אמת' אם המחסנית ריקה ו-'שקר' אחרת. <u>הנחה</u> : המחסנית S מאותחלת.	function stack_is_empty (S:stack_type):boolean;
דחוף-למחסנית (S, x)	פעולה המכניסה את האיבר x לראש מחסנית S. <u>הנחה</u> : המחסנית S מאותחלת.	procedure stack_push (var S:stack_type; x:stack_info_type);
שלוף-ממחסנית (S)	פעולה המוציאה את ערכו של האיבר שבראש המחסנית S ומחזירה את ערכו. <u>הנחות</u> : המחסנית S מאותחלת ואינה ריקה.	procedure stack_pop (var S:stack_type; var x:stack_info_type);
הצף-למחסנית (S)	פעולה המחזירה את ערכו של האיבר שבראש המחסנית S מבלי להוציאו. <u>הנחות</u> : המחסנית S מאותחלת ואינה ריקה.	procedure stack_top (S:stack_type; var x:stack_info_type);

תור

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-תור	פעולה המחזירה תור ריק.	procedure queue_init (var Q:queue_type);
תור-ריק? (Q)	פעולה המקבלת כפרמטר תור Q מחזירה 'אמת' אם התור ריק ו-'שקר' אחרת. <u>הנחה</u> : התור Q מאותחל.	function queue_is_empty (Q:queue_type):boolean;
הכנס-לתור (Q, x)	פעולה המכניסה את האיבר x בסוף התור Q. <u>הנחה</u> : התור Q מאותחל.	procedure queue_insert (var Q:queue_type; x:queue_info_type);
הוצא-מתור (Q)	פעולה המוציאה את ערכו של האיבר שבראש התור Q ומחזירה את ערכו. <u>הנחות</u> : התור Q מאותחל ואינו ריק.	procedure queue_remove (var Q:queue_type; var x:queue_info_type);
ראש-התור (Q)	פעולה המחזירה את ערכו של האיבר שבראש התור Q מבלי להוציאו. <u>הנחות</u> : התור Q מאותחל ואינו ריק.	procedure queue_top (Q:queue_type; var x:queue_info_type);

עץ-בינארי

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-עץ	פעולה המחזירה עץ בינארי ריק.	function tree_init : tree_type;
בנה-עץ (L, R, x)	פעולה המחזירה עץ בינארי שבשורשו האיבר x, התת-עץ-שמאלי שלו L והתת-עץ-ימני שלו R. <u>הנחות</u> : העצים T ו r מאותחלים.	function tree_build (L, R:tree_type; x:tree_info_type):tree_type;
תת-עץ-שמאלי (T)	פעולה המחזירה את התת-עץ-שמאלי של T. <u>הנחות</u> : T מאותחל ואינו ריק.	function tree_Lsub (T :tree_type):tree_type;
תת-עץ-ימני (T)	פעולה המחזירה את התת-עץ-ימני של T. <u>הנחות</u> : T מאותחל ואינו ריק.	function tree_Rsub (T :tree_type):tree_type;
החלף-תת-עץ-שמאלי (T, new_T)	פעולה המחליפה את התת-עץ-שמאלי של T בעץ הבינארי new_T. <u>הנחות</u> : העצים T ו- new_T מאותחלים, T אינו ריק.	procedure tree_change_Lsub (var T:tree_type; new_T:tree_type);
החלף-תת-עץ-ימני (T, new_T)	פעולה המחליפה את התת-עץ-ימני של T בעץ הבינארי new_T. <u>הנחות</u> : העצים T ו- new_T מאותחלים, T אינו ריק.	procedure tree_change_Rsub (var T:tree_type; new_T:tree_type);
אחזר-שורש (T)	פעולה המחזירה את האיבר שבשורשו של T. <u>הנחות</u> : T מאותחל ואינו ריק.	function tree_root_retrieve (T:tree_type) :tree_info_type;
עדכן-שורש (T, x)	פעולה המשנה/מעדכנת את האיבר שבשורשו T להיות x. <u>הנחות</u> : T מאותחל ואינו ריק.	procedure tree_root_modify (var T:tree_type; x:tree_info_type);
עץ-ריק? (T)	פעולה המחזירה 'אמת' אם העץ הבינארי T הוא עץ ריק ו-'שקר' אחרת. <u>הנחה</u> : העץ T מאותחל.	function tree_is_empty (T:tree_type):boolean;

רשימה

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-רשימה	פעולה המחזירה רשימה ריקה	<code>list_type list_init (void);</code>
עוגן-רשימה (L)	פעולה המחזירה את המקום עוגן-רשימה ברשימה L. <u>הנחה</u> : הרשימה L מאותחלת.	<code>pos_type list_anchor (list_type L);</code>
סוף-רשימה (L)	פעולה המחזירה את המקום סוף-רשימה ברשימה L. <u>הנחה</u> : הרשימה L מאותחלת.	<code>pos_type list_end (list_type L);</code>
עוקב-ברשימה (L, p)	פעולה המחזירה את המקום העוקב למקום p ברשימה L. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו סוף-רשימה.	<code>pos_type list_next (list_type L, pos_type p);</code>
קודם-ברשימה (L, p)	פעולה המחזירה את המקום הקודם למקום p ברשימה L. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו עוגן-רשימה.	<code>pos_type list_prev (list_type L, pos_type p);</code>
הכנס-לרשימה (L, p, x)	פעולה מכניסה לרשימה L את האיבר x מקום אחד אחרי המקום p. <u>הנחות</u> : הרשימה L מאותחלת. P הוא מקום ב-L שאינו סוף-רשימה.	<code>void list_insert (list_type L, pos_type p, list_info_type x);</code>
הוצא-מרשימה (L, p)	פעולה המוציאה מן הרשימה L את האיבר הנמצא בה במקום p. לאחר ההוצאה נמצא במקום p האיבר שהיה עוקב לזה שהוצא מהרשימה. <u>הנחות</u> : הרשימה L מאותחלת. P הוא מקום ב-L שאינו סוף-רשימה ואינו עוגן-רשימה.	<code>void list_delete (list_type L, pos_type *p);</code>

שם ההליך	תיאור ההליך	תחביר הפעולה
עדכן-רשימה (L, p, x)	פעולה המעדכנת את האיבר הנמצא במקום p ברשימה להיות x. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו <u>סוף-רשימה</u> ואינו <u>עוגן-רשימה</u> .	void list_update (list_type L, pos_type p, list_info_type x);
אחזר-מרשימה (L, p)	פעולה המחזירה את האיבר הנמצא במקום p ברשימה. <u>הנחות</u> : הרשימה L מאותחלת. p הוא מקום ב-L שאינו <u>סוף-רשימה</u> ואינו <u>עוגן-רשימה</u> .	list_info_type list_retrieve (list_type L, pos_type p);
רשימה-ריקה? (L)	פעולה המחזירה אמת אם הרשימה L היא רשימה ריקה, אחרת מחזירה שקר . <u>הנחה</u> : הרשימה L מאותחלת.	int list_is_empty (list_type L);

מחסנית

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-מחסנית	פעולה המחזירה מחסנית ריקה.	<code>stack_type stack_init (void);</code>
מחסנית-ריקה? (S)	פעולה המקבלת כפרמטר מחסנית S מחזירה 'אמת' אם המחסנית ריקה ו-'שקר' אחרת. <u>הנחה</u> : המחסנית S מאותחלת.	<code>int stack_is_empty (stack_type S);</code>
דחוף-למחסנית (S, x)	פעולה המכניסה את האיבר x לראש מחסנית S. <u>הנחה</u> : המחסנית S מאותחלת.	<code>void stack_push (stack_type *S, stack_info_type x);</code>
שלוף-ממחסנית (S)	פעולה המוציאה את ערכו של האיבר שבראש המחסנית S ומחזירה את ערכו. <u>הנחות</u> : המחסנית S מאותחלת ואינה ריקה.	<code>stack_info_type stack_pop (stack_type *S);</code>
הצף-למחסנית (S)	פעולה המחזירה את ערכו של האיבר שבראש המחסנית S מבלי להוציאו. <u>הנחות</u> : המחסנית S מאותחלת ואינה ריקה.	<code>stack_info_type stack_top (stack_type S);</code>

תור

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-תור	פעולה המחזירה תור ריק.	queue_type queue_init (void);
תור-ריק? (Q)	פעולה המקבלת כפרמטר תור Q מחזירה 'אמת' אם התור ריק ו-'שקר' אחרת. <u>הנחה</u> : התור Q מאותחל.	int queue_is_empty (queue_type Q);
הכנס-לתור (Q, x)	פעולה המכניסה את האיבר x בסוף התור Q. <u>הנחה</u> : התור Q מאותחל.	void queue_insert (queue_type *Q, queue_info_type x);
הוצא-מתור (Q)	פעולה המוציאה את ערכו של האיבר שבראש התור Q ומחזירה את ערכו. <u>הנחות</u> : התור Q מאותחל ואינו ריק.	queue_info_type queue_remove (queue_type *Q);
ראש-התור (Q)	פעולה המחזירה את ערכו של האיבר שבראש התור Q מבלי להוציאו. <u>הנחות</u> : התור Q מאותחל ואינו ריק.	queue_info_type queue_top (queue_type Q);

עץ-בינארי

שם ההליך	תיאור ההליך	תחביר הפעולה
אתחל-עץ	פעולה המחזירה עץ בינארי ריק.	<code>tree_type tree_init (void);</code>
בנה-עץ (L, R, x)	פעולה המחזירה עץ בינארי שבשורשו האיבר x, התת-עץ-השמאלי שלו L והתת-עץ-הימני שלו R. <u>הנחות</u> : העצים T ו- r מאותחלים.	<code>tree_type tree_build (tree_type Lsub, tree_type Rsub, tree_info_type x);</code>
תת-עץ-שמאלי (T)	פעולה המחזירה את התת-עץ-השמאלי של T. <u>הנחות</u> : T מאותחל ואינו ריק.	<code>tree_type tree_Tsub (tree_type T);</code>
תת-עץ-ימני (T)	פעולה המחזירה את התת-עץ-הימני של T. <u>הנחות</u> : T מאותחל ואינו ריק.	<code>tree_type tree_Rsub (tree_type T);</code>
החלף-תת-עץ-שמאלי (T, new_T)	פעולה המחליפה את התת-עץ-השמאלי של T בעץ הבינארי new_T. <u>הנחות</u> : העצים T ו- new_T מאותחלים, T אינו ריק.	<code>void tree_change_Lsub (tree_type *T, tree_type new_T);</code>
החלף-תת-עץ-ימני (T, new_T)	פעולה המחליפה את התת-עץ-הימני של T בעץ הבינארי new_T. <u>הנחות</u> : העצים T ו- new_T מאותחלים, T אינו ריק.	<code>void tree_change_Rsub (tree_type *T, tree_type new_T);</code>
אחזר-שורש (T)	פעולה המחזירה את האיבר שבשורשו של T. <u>הנחות</u> : T מאותחל ואינו ריק.	<code>tree_info_type tree_root_retrieve (tree_type T);</code>
עדכן-שורש (T, x)	פעולה המשנה/מעדכנת את האיבר שבשורשו T להיות x. <u>הנחות</u> : T מאותחל ואינו ריק.	<code>void tree_root_modify (tree_type *T, tree_info_type x);</code>
עץ-ריק? (T)	פעולה המחזירה 'אמת' אם העץ הבינארי T הוא עץ ריק ו-'שקר' אחרת. <u>הנחה</u> : העץ T מאותחל.	<code>int tree_is_empty (tree_type T);</code>

