

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

1η Ατομική: Προσομοίωση Ουράς Τράπεζας

Να επεκτείνετε το πρόγραμμα προσομοίωσης της ουράς πελατών μιας Τράπεζας που συνοδεύει την άσκηση με νέες δυνατότητες και λειτουργικότητα ως ακολούθως

1. Να υλοποιήσετε τον ΑΤΔ Ουρά χρησιμοποιώντας μετρητή (η 3^η υλοποίηση στις διαφάνειες). Ελέγξτε την νέα υλοποίηση με πρόγραμμα δοκιμής.
2. Να προσθέσετε στην υλοποίηση του ΑΤΔ Ουρά την πράξη
`int SizeOuras(TOuras oura);`
που επιστρέφει τον αριθμό στοιχείων σε μια ουρά. Ελέγξτε την υλοποίηση με πρόγραμμα δοκιμής. Επεκτείνετε το κυρίως πρόγραμμα, ώστε να εκτυπώνει τον αριθμό πελατών στην ουρά που δεν εξυπηρετήθηκαν.
3. Να επεκτείνετε τον TStoixείουOuras, ώστε το struct να περιλαμβάνει όχι μόνο το λεπτό εισόδου (όπως στην υλοποίηση που σας δίδεται), αλλά επίσης τον χρόνο εξυπηρέτησης του πελάτη (`int XronosEksipiretisis` για την ώρα σταθερός). Επίσης να προσθέσετε δυο συναρτήσεις
`void PelatisSetXronoEksipiretisis(TPelatis *stoixeioPtr, int duration);`
`int PelatisGetXronoEksipiretisis(TPelatis *stoixeioPtr);`
Η πρώτη ενημερώνει τον χρόνο Εξυπηρέτησης και η δεύτερη επιστρέφει τον χρόνο εξυπηρέτησης. Ελέγξτε την υλοποίησή σας με πρόγραμμα δοκιμής. Κατόπιν επεκτείνετε το κυρίως πρόγραμμα, ώστε αντί ο χρόνος εξυπηρέτησης ενός πελάτη να είναι σταθερός να είναι μεταξύ 1 και ενός Μέγιστου που να ζητείται ως είσοδος στο πρόγραμμα αντί του σταθερού χρόνου. Σημειώνουμε ότι η συνάρτηση `rand( )` επιστρέφει τυχαίο ακέραιο μεταξύ 0 και `MAX_RANDOM`, `rand( ) % X` επιστρέφει τυχαίο μεταξύ 0 και `X-1`.
4. Να επεκτείνετε το κυρίως πρόγραμμα, ώστε να δέχεται τον χρόνο που κλείνει η τράπεζα (μετά τον οποίο δεν δέχεται άλλους πελάτες). Όμως οι υπάρχοντες πελάτες στην ουρά να εξυπηρετούνται όλοι. Να εκτυπώσετε τον πραγματικό τελικό χρόνο λειτουργίας και πόσα επιπλέον λεπτά ήταν απαραίτητα.
5. Να σχεδιάσετε και να αναπτύξετε μια νέα ενότητα για τον ταμιά (`tamias.h` και `tamias.c`), που να ορίζει (`struct TTamias`) και να χειρίζεται α) τον χρόνο που ήταν απασχολημένος (`TimeBusy`), β) τον χρόνο που ήταν αδρανής (`TimeInactive`), γ) τον αριθμό πελατών που εξυπηρέτησε (`ArithmoPelaton`) και δ) τον εναπομείναντα χρόνο (`enapomenonXronos`) για να ολοκληρώσει την εξυπηρέτηση ενός πελάτη. Θα χρειαστείτε τις συναρτήσεις

`void TamiasDimiourgia(TTamias *tamias);` //αρχικοποιεί τα μέλη του struct ταμιά

`void TamiasNewCustomer(TTamias *tamias);`

// αυξάνει τον μετρητή πελατών και προσθέτει 1 λεπτό απασχόλησης

`void TamiasSetXrono(TTamias *tamias, int Duration);` // αρχικοποιεί εναπομείναντα χρόνο

`void TamiasNoWork(TTamias *tamias);` // αυξάνει χρόνο αδράνειας

`void TamiasBusy(TTamias *tamias);` // αυξάνει χρόνο απασχόλησης

`int TamiasFree(TTamias tamias);` // ελέγχει αν είναι διαθέσιμος

`int TamiasGetArithmosPelatwn(TTamias *tamias);` // επιστρέφει αριθμό πελατών

`int TamiasGetEnapomenonXronos(TTamias *tamias);` // επιστρέφει εναπομείναντα χρόνο

`int TamiasGetInactiveXronos(TTamias *tamias);` // επιστρέφει χρόνο αδράνειας

`int TamiasGetBusyXronos(TTamias *tamias);` // επιστρέφει χρόνο απασχόλησης

Αλλάζτε το κυρίως πρόγραμμα, ώστε να χρησιμοποιεί αποκλειστικά τον ανωτέρω τύπο του Ταμιά (μέσω των συναρτήσεων). Δεν χρειάζεται αλλαγή δομής, αλλά αντί για την χρήση μεταβλητών προγράμματος να κάνετε χρήση των πράξεων της ενότητας ταμιά.

6. Να επεκτείνετε το κυρίως πρόγραμμα προσομοίωσης ώστε να υποστηρίζει πολλούς (πίνακας) ταμίες (οι πελάτες όπως πριν σε μια κοινή ουρά). Στο τέλος της προσομοίωσης να εκτυπώνει για κάθε ταμιά τον αριθμό πελατών που εξυπηρέτησε, τον χρόνο που ήταν απασχολημένος και τον χρόνο που είναι αδρανής.
7. Για επιπλέον Bonus 10%. Σχεδιάστε και αναπτύξτε μια δική σας πολιτική διαχείρισης πελατών/ταμιών, π.χ. να μην εργάζονται όλοι οι ταμίες αν δεν υπάρχουν αρκετοί πελάτες ή να βάζουμε επιπλέον ταμιά όταν μεγαλώνει πολύ η ουρά και να βγάζουμε ταμιά όταν μικραίνει, κλπ.

Οδηγίες Σχεδίασης και Ανάπτυξης Προγράμματος

Το πρόγραμμά σας πρέπει να είναι οργανωμένο σε ενότητες (modules) και σε πρόγραμμα-πελάτη. Το πρόγραμμα πελάτης να μην χρησιμοποιεί άμεσα την δομή των ενότητων στα .h παρά μόνο μέσω των συναρτήσεων που ορίζονται.

Σας δίδεται αρχικό πρόγραμμα που θα επεκτείνετε, καθώς επίσης και πρόγραμμα δοκιμής ουράς σε ξεχωριστό κατάλογο.

Συνιστάται να κρατάτε αντίγραφα των προγραμμάτων σε κάθε στάδιο ανάπτυξης, π.χ. μετά την ολοκλήρωση κάθε ερωτήματος, ώστε να έχετε μια προηγούμενη έκδοση του προγράμματος σας. Ένα αντίγραφο θα σας φανεί πολύ χρήσιμο αν θέλετε να επιστρέψετε σε προηγούμενη έκδοση.