



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
Τμήμα Πληροφορικής
ΕΠΛ 133 – Αντικειμενοστρεφής Προγραμματισμός
ΑΣΚΗΣΗ 2 – Travel Agency

Διδάσκων: Καθ. Μάριος Δικαιάκος

Υπεύθυνος Άσκησης: Πύρρος Μπράτσкас

Ημερομηνία Ανάθεσης: 8 Μαρτίου 2024

Ημερομηνία Παράδοσης: 18 Μαρτίου 2024 13:00

(ο κώδικας να υποβληθεί μέσω του Moodle)

I. Στόχος

Σε αυτήν την εργασία θα ασχοληθούμε με αντικειμενοστρεφή σχεδίαση προβλημάτων, σχεδίαση κλάσεων, μεθόδων, αντικείμενα, και κληρονομικότητα. Για την εργασία έχουν ληφθεί πολλές αποφάσεις σχεδιασμού για εσάς, αλλά είναι επίσης σημαντικό να αναρωτηθείτε γιατί έχει γίνει η κάθε επιλογή και αν θα μπορούσε να υπάρξει καλύτερος τρόπος για να το κάνετε. Μερικές από τις επιλογές μας επηρεάστηκαν από την πρόθεση να σας κάνουμε να εξασκήσετε τα περισσότερα από αυτά που έχετε μάθει στις διαλέξεις και στα εργαστήρια. Άλλοι, υπαγορεύτηκαν από την ανάγκη μας να είμαστε σε θέση να ελέγξουμε/δοκιμάσουμε πλήρως τις εργασίες σας.

Φροντίστε να αναρωτηθείτε γιατί χρησιμοποιήθηκαν ορισμένοι τροποποιητές (private, public, static,...) και αν θα είχατε πάρει τις ίδιες αποφάσεις. Ελπίζουμε ότι η εργασία θα σας βοηθήσει να εκτιμήσετε τη σημασία του σχεδιασμού μιας κλάσης υλοποιώντας τις αρχές αντικειμενοστρεφούς προγραμματισμού **SOLID**.

Σας προτείνουμε να αφιερώσετε χρόνο για να σχεδιάσετε ένα διάγραμμα κλάσης UML. Αυτό θα σας βοηθήσει να αναπτύξετε μια σαφή εικόνα της σχέσης μεταξύ όλων αυτών των κλάσεων.

II. Περιγραφή

Για αυτήν την εργασία θα γράψετε πολλές κλάσεις οι οποίες προσομοιώνουν ένα διαδικτυακό ταξιδιωτικό γραφείο. Σημειώστε ότι για να είναι εφικτή η υλοποίηση αυτής της εργασίας, έχουν γίνει πολλές απλουστεύσεις. Για παράδειγμα, αγνοούμε εντελώς τον χρόνο εκτέλεσης καθ' όλη τη διάρκεια της εργασίας. Φροντίστε να ακολουθήσετε πολύ προσεκτικά τις παρακάτω οδηγίες. Σημειώστε ότι εκτός από τις απαιτούμενες μεθόδους, μπορείτε να προσθέσετε όσες άλλες ιδιωτικές μεθόδους θέλετε (δεν επιτρέπεται καμία άλλη μη-ιδιωτική πρόσθετη μέθοδος). Δεν επιτρέπεται η προσθήκη τυχόν πρόσθετων πεδίων (είτε θα ήταν ιδιωτικά είτε δημόσια), εκτός εάν αναφέρεται διαφορετικά. Τέλος, όποτε σας ζητηθεί να συγκρίνετε δύο συμβολοσειρές, μπορείτε να θεωρήσετε ίδιους τους μικρούς και κεφαλαίους χαρακτήρες. Ακολουθεί η περιγραφή των κλάσεων του προγράμματος:

Κλάση Airport.

Ένα αεροδρόμιο έχει τα παρακάτω ιδιωτικά πεδία:

- Ένα int που υποδηλώνει την x-συντεταγμένη του αεροδρομίου σε έναν παγκόσμιο χάρτη με κλίμακα 1 χιλιόμετρο.
- Ένα int που υποδηλώνει την y-συντεταγμένη του αεροδρομίου σε έναν παγκόσμιο χάρτη με κλίμακα 1 χιλιόμετρο.
- Ένα int που υποδηλώνει τα τέλη αεροδρομίου που συνδέονται με αυτό το αεροδρόμιο.

Η κλάση πρέπει επίσης να έχει τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει τρία int ως παραμέτρους που υποδηλώνουν τη θέση του αεροδρομίου σε ένα χάρτη (συντεταγμένες x και y) και τα τέλη του αεροδρομίου αντίστοιχα. Ο

κατασκευαστής χρησιμοποιεί τις παραμέτρους για να αρχικοποιήσει τα αντίστοιχα πεδία.

- Μια μέθοδο `getFees` για την ανάκτηση των τελών του αεροδρομίου.
- Μια στατική μέθοδο `getDistance` που παίρνει ως παραμέτρους δύο αεροδρόμια και επιστρέφει έναν ακέραιο που υποδηλώνει την απόσταση σε χιλιόμετρα μεταξύ τους. Σημειώστε ότι η μέθοδος πρέπει να στρογγυλοποιεί την απόσταση προς τα επάνω (π.χ. τόσο το 5,9 όσο και το 5,2 πρέπει να γίνουν 6). Επιπλέον, να θυμάστε ότι δεδομένων δύο σημείων (x_1, y_1) και (x_2, y_2) , η απόσταση μπορεί να υπολογιστεί με τον ακόλουθο τύπο: $\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Κλάση Room.

Η κλάση Room έχει τα ακόλουθα ιδιωτικά πεδία:

- Ένα String που υποδηλώνει τον τύπο του δωματίου.
- Ένα int που υποδηλώνει την τιμή (σε λεπτά) του δωματίου.
- Ένα boolean που υποδηλώνει εάν το δωμάτιο είναι διαθέσιμο ή όχι.

Η κλάση πρέπει επίσης να έχει τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει ως παράμετρο τον τύπο του δωματίου και τον χρησιμοποιεί για να αρχικοποιήσει τα πεδία. Σημειώστε ότι υποστηρίζονται μόνο 3 τύποι δωματίων από το πρόγραμμα: `double`, `queen` και `king`. Εάν η παράμετρος δεν αντιστοιχεί σε έναν από αυτούς τους τύπους δωματίων, τότε ο κατασκευαστής πρέπει να πετάξει ένα `IllegalArgumentException` εξηγώντας ότι δεν μπορεί να δημιουργηθεί δωμάτιο τέτοιου τύπου. Η τιμή του δωματίου βασίζεται στον τύπο του ως εξής: \$90 για ένα `double`, \$110 για ένα `queen`, \$150 για ένα `king`. Σημειώστε ότι η τιμή πρέπει να αποθηκευτεί σε λεπτά (cents). Ο κατασκευαστής πρέπει να ορίσει τη διαθεσιμότητα για ένα νέο δωμάτιο να είναι `true`.
- Έναν κατασκευαστή που παίρνει ένα Room ως παράμετρο και δημιουργεί ένα αντίγραφο του δωματίου εισόδου (δηλαδή αρχικοποιεί τα πεδία χρησιμοποιώντας τις τιμές από τα αντίστοιχα πεδία του δωματίου εισόδου).
- Μια μέθοδο `getType` και μια μέθοδο `getPrice` η οποία επιστρέφει τον τύπο και την τιμή του δωματίου αντίστοιχα.
- Μια μέθοδο `changeAvailability` που δεν λαμβάνει παράμετρο και ορίζει την τιμή που αποθηκεύεται στο πεδίο διαθεσιμότητας να είναι το αντίθετο από αυτό που υπάρχει εκεί αυτή τη στιγμή.
- Μια στατική μέθοδο `findAvailableRoom` η οποία λαμβάνει ως παράμετρο έναν πίνακα με Room καθώς και ένα String που υποδηλώνει έναν τύπο δωματίου. Η μέθοδος πρέπει να επιστρέψει το πρώτο διαθέσιμο δωμάτιο στον πίνακα του καθορισμένου τύπου. Εάν δεν υπάρχει τέτοιο δωμάτιο (είτε επειδή όλα τα δωμάτια του συγκεκριμένου τύπου είναι κατειλημμένα, είτε επειδή δεν υπάρχει δωμάτιο του συγκεκριμένου τύπου στον πίνακα), η μέθοδος επιστρέφει `null`. **Σημειώστε ότι δεν πρέπει να γίνουν αλλαγές σε κανένα από τα δωμάτια στον πίνακα της παραμέτρου από αυτήν τη μέθοδο!**
- Μια στατική μέθοδο `makeRoomAvailable` η οποία λαμβάνει ως παράμετρο έναν πίνακα με Room καθώς και ένα String που υποδηλώνει έναν τύπο δωματίου. Η μέθοδος πρέπει να κάνει το πρώτο μη διαθέσιμο δωμάτιο στον πίνακα του καθορισμένου τύπου διαθέσιμο ξανά. Εάν αυτό είναι επιτυχές, η μέθοδος πρέπει να επιστρέψει `true`, διαφορετικά η μέθοδος πρέπει να επιστρέψει `false`.

Κλάση Hotel.

Ένα ξενοδοχείο έχει τα παρακάτω ιδιωτικά πεδία:

- Ένα String που υποδηλώνει το όνομα του ξενοδοχείου.
- Έναν πίνακα με Room που υποδηλώνει τα δωμάτια στο ξενοδοχείο.

Η κλάση πρέπει επίσης να έχει τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει σαν παραμέτρους ένα String και έναν πίνακα με Room αντίστοιχα. Ο κατασκευαστής χρησιμοποιεί τις παραμέτρους για να αρχικοποιήσει τα αντίστοιχα πεδία. Σημειώστε ότι ο πίνακας που χρησιμοποιείται για να αρχικοποιήσει το πεδίο που αντιπροσωπεύει τα δωμάτια πρέπει να είναι μια βαθιά αντιγραφή (deep copy) του πίνακα εισόδου.
- Μια μέθοδο `reserveRoom` η οποία παίρνει ως παράμετρο ένα String που υποδηλώνει τον τύπο του δωματίου που θα καταχωρηθεί. Η μέθοδος αλλάζει τη διαθεσιμότητα του πρώτου διαθέσιμου δωματίου του συγκεκριμένου τύπου στο ξενοδοχείο. Εάν είναι επιτυχές, η μέθοδος επιστρέφει την τιμή του δωματίου. Διαφορετικά, πρέπει να πεταχτεί ένα `IllegalArgumentException`.
- Μια μέθοδο `cancelRoom` η οποία παίρνει ως παράμετρο ένα String που υποδηλώνει τον τύπο του

δωματίου που θα ακυρωθεί. Η μέθοδος καθιστά διαθέσιμο ένα δωμάτιο αυτού του τύπου. Επιστρέφει true αν η λειτουργία ήταν δυνατή, διαφορετικά επιστρέφει false.

Μια αφαιρετική κλάση Reservation.

Η κλάση έχει το ακόλουθο ιδιωτικό πεδίο:

- Ένα String όνομα

Η κλάση πρέπει επίσης να έχει τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει ένα String ως παράμετρο που υποδηλώνει το όνομα του πελάτη στην κράτηση και το χρησιμοποιεί για να αρχικοποιήσει το αντίστοιχο πεδίο.
- Μια final μέθοδο reservationName για την ανάκτηση του ονόματος σε αυτήν την κράτηση.
- Μια αφαιρετική μέθοδο getCost η οποία δεν λαμβάνει είσοδο και επιστρέφει έναν ακέραιο. Αυτή η μέθοδος πρέπει να είναι αφαιρετική επειδή ο τρόπος που καθορίζεται το κόστος εξαρτάται από τον τύπο της κράτησης.
- Μια αφαιρετική μέθοδο equals η οποία σέβεται τον εαυτό της, δηλαδή παίρνει ένα Object ως είσοδο και επιστρέφει ένα boolean. Αυτή η μέθοδος πρέπει επίσης να είναι αφαιρετική, καθώς ανάλογα με τον τύπο της κράτησης πρέπει να ικανοποιούνται διαφορετικές συνθήκες για να θεωρηθούν δύο κρατήσεις ίσες.

Όλες οι παρακάτω κλάσεις πρέπει να είναι υποκλάσεις της κλάσης Reservation:

- Γράψτε μια κλάση FlightReservation η οποία κληρονομεί από την κλάση Reservation. Αυτή η κλάση έχει τα ακόλουθα ιδιωτικά πεδία:
 - Ένα Αεροδρόμιο που υποδηλώνει τον τόπο αναχώρησης.
 - Ένα Αεροδρόμιο που υποδηλώνει τον τόπο άφιξης.

Η κλάση έχει επίσης τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει ως παράμετρο ένα String με το όνομα στην κράτηση και δύο Αεροδρόμια που υποδηλώνουν τον τόπο αναχώρησης και άφιξης αντίστοιχα. Ο κατασκευαστής χρησιμοποιεί τις παραμέτρους για να δημιουργήσει ένα Reservation και να αρχικοποιήσει τα αντίστοιχα πεδία. Το πρόγραμμα δίνει ένα IllegalArgumentException εάν τα δύο αεροδρόμια εισόδου είναι τα ίδια. Αγνοήστε το γεγονός ότι δεν επαναορίσατε τη μέθοδο equals στην κλάση Airport.
- Μια μέθοδο getCost που δεν λαμβάνει παραμέτρους και επιστρέφει το κόστος της κράτησης (ένας ακέραιος) σε λεπτά. Το κόστος υπολογίζεται προσθέτοντας μαζί το κόστος καυσίμων, τα τέλη αεροδρομίων και \$53.75 (που περιλαμβάνουν το κόστος σχετικά με το αεροπλάνο συν φόρους). Μπορείτε να υποθέσετε τα εξής:
 - ο Τα αεροπλάνα πληρώνουν \$1.24 ανά γαλόνι καυσίμων.
 - ο Τα αεροπλάνα μπορούν να πετάξουν 167.52 χιλιόμετρα ανά γαλόνι καυσίμων.
 Το κόστος πρέπει να στρογγυλοποιηθεί προς τα πάνω στο πλησιέστερο λεπτό. Για παράδειγμα, αν μετά τον παραπάνω υπολογισμό παίρνετε ένα κόστος πτήσης των 103568.21187 λεπτών, η μέθοδος θα πρέπει να επιστρέψει 103569 λεπτά. Μπορείτε να υποθέσετε ότι το κόστος όλων των κρατήσεων ταιριάζει σε έναν ακέραιο και επομένως δεν προκαλεί υπερχείλιση.
- Μια μέθοδο equals η οποία παίρνει ως παράμετρο ένα Object και επιστρέφει true αν η παράμετρος ταιριάζει με την this όσον αφορά τον τύπο, το όνομα και τα αεροδρόμια. Διαφορετικά, η μέθοδος επιστρέφει false. Μπορείτε να αγνοήσετε το γεγονός ότι δεν επαναορίσατε τη μέθοδο equals στην κλάση Airport.

- Γράψτε μια κλάση `HotelReservation` η οποία κληρονομεί από την κλάση `Reservation`. Αυτή η κλάση έχει τα ακόλουθα ιδιωτικά πεδία:
 - Ένα `Hotel` που υποδηλώνει πού να γίνει η κράτηση.
 - Ένα `String` που υποδηλώνει τον τύπο του δωματίου για κράτηση.
 - Έναν ακέραιο που υποδηλώνει τον αριθμό των νυχτών που θα περάσουν στο ξενοδοχείο.
 - Έναν ακέραιο που υποδηλώνει την τιμή (σε λεπτά) για μια νύχτα σε ένα δωμάτιο του καθορισμένου τύπου στο καθορισμένο ξενοδοχείο.

Η κλάση έχει επίσης τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει ως παραμέτρους ένα `String` με το όνομα στην κράτηση, ένα `Hotel`, ένα `String` με τον τύπο του δωματίου και έναν ακέραιο που υποδηλώνει τον αριθμό των νυχτών. Ο κατασκευαστής χρησιμοποιεί τις παραμέτρους για να δημιουργήσει μια Κράτηση (`Reservation`) και να αρχικοποιήσει τα αντίστοιχα πεδία. Ο κατασκευαστής πρέπει επίσης να βεβαιωθεί ότι κρατείται ένα δωμάτιο του σωστού τύπου στο καθορισμένο ξενοδοχείο. Εάν μια τέτοια κράτηση δεν είναι δυνατή, τότε πρέπει να πεταχτεί ένα `IllegalArgumentException`.
- Μια μέθοδο `getNumOfNights` για να ανακτήσετε τον αριθμό των νυχτών στην κράτηση.
- Μια μέθοδο `getCost` που δεν δέχεται παραμέτρους και επιστρέφει το κόστος της κράτησης (ένας ακέραιος) σε λεπτά. Το κόστος αντιπροσωπεύει την τιμή που πρέπει να πληρωθεί για τον καθορισμένο τύπο δωματίου δεδομένου του αριθμού των νυχτών που αναφέρονται στην κράτηση.
- Μια μέθοδο `equals` που παίρνει ως παράμετρο ένα `Object` και επιστρέφει `true` αν η παράμετρος ταιριάζει με την `this` όσον αφορά τον τύπο, το όνομα, το ξενοδοχείο, τον τύπο δωματίου, τον αριθμό των νυχτών και το συνολικό κόστος. Διαφορετικά, η μέθοδος επιστρέφει `false`. Ακόμα μία φορά, μπορείτε να αγνοήσετε το γεγονός ότι δεν επαναορίσατε τη μέθοδο `equals` στην κλάση `Hotel`.
- Γράψτε μια κλάση `BnBReservation` η οποία κληρονομεί από την κλάση `HotelReservation`. Αυτή η κλάση δεν έχει πεδία, αλλά έχει τις ακόλουθες δημόσιες μεθόδους:
 - Έναν κατασκευαστή που παίρνει ως παραμέτρους ένα `String` με το όνομα στην κράτηση, ένα `Hotel`, ένα `String` με τον τύπο του δωματίου και έναν ακέραιο που υποδηλώνει τον αριθμό των νυχτών. Ο κατασκευαστής χρησιμοποιεί τις παραμέτρους για να δημιουργήσει μια κράτηση ξενοδοχείου.
 - Μια μέθοδο `getCost` που δεν δέχεται παραμέτρους και επιστρέφει το κόστος της κράτησης (ένας ακέραιος) σε λεπτά. Δεδομένου ότι αυτή η κράτηση περιλαμβάνει πρωινό, στο κόστος της κράτησης του δωματίου στο ξενοδοχείο πρέπει να προστεθούν \$10 ανά νύχτα.

Κλάση `Basket`

Αυτή η κλάση που αντιπροσωπεύει μια λίστα κρατήσεων. Σημειώστε ότι οι οδηγίες για το πώς να υλοποιηθεί αυτή η κλάση δεν είναι πολύ συγκεκριμένες. Αυτό είναι εσκεμμένο, καθώς η εργασία σας δεν θα δοκιμαστεί στις λεπτομέρειες που λείπουν από την υλοποίηση. Σημειώστε, ωστόσο, ότι οι επιλογές σας θα κάνουν τη διαφορά όσον αφορά την αποτελεσματικότητα του κώδικά σας. Δεν θα αφαιρεθούν πόντοι για μη αποτελεσματικό κώδικα στην Εργασία 2.

Η κλάση έχει (τουλάχιστον) το παρακάτω ιδιωτικό πεδίο (μπορείτε να προσθέσετε επιπλέον ιδιωτικά πεδία σε αυτή την κλάση αν το επιθυμείτε):

- Έναν πίνακα Κρατήσεων (`An array of Reservations`).

Η κλάση πρέπει επίσης να έχει τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που δεν δέχεται καμία παράμετρο και αρχικοποιεί το πεδίο με έναν πίνακα χωρίς κρατήσεις.
- Μια μέθοδο `getProducts` η οποία δεν δέχεται παραμέτρους και επιστρέφει έναν ψηφιακό αντίγραφο

του πίνακα Κρατήσεων του Basket. Αυτός ο πίνακας πρέπει να περιέχει όλες τις κρατήσεις στο καλάθι στην ίδια σειρά με την προσθήκη τους.

- Μια μέθοδο `add` η οποία δέχεται ως παράμετρο μια Κράτηση (Reservation). Η μέθοδος προσθέτει την κράτηση στο τέλος της λίστας κρατήσεων του καλαθιού και επιστρέφει πόσες κρατήσεις υπάρχουν τώρα εκεί.
- Μια μέθοδο `remove` η οποία δέχεται ως παράμετρο μια Κράτηση (Reservation) και επιστρέφει ένα boolean. Η μέθοδος αφαιρεί την πρώτη εμφάνιση του καθορισμένου στοιχείου από τον πίνακα κρατήσεων του καλαθιού. Εάν δεν υπάρχει τέτοια κράτηση, τότε η μέθοδος επιστρέφει `false`, αλλιώς, μετά την αφαίρεσή της, η μέθοδος επιστρέφει `true`. Σημειώστε ότι αυτή η μέθοδος αφαιρεί μια κράτηση από τη λίστα μόνο εάν τέτοια κράτηση είναι ίση με την εισαγόμενη παράμετρο. Για παράδειγμα, δύο κρατήσεις πτήσεων από το Μόντρεαλ στο Βανκούβερ δεν θεωρούνται ίσες αν δημιουργηθούν υπό δύο διαφορετικά ονόματα. Αφού η κράτηση αφαιρεθεί από τον πίνακα, τα επόμενα στοιχεία πρέπει να μετακινηθούν αριστερά κατά μία θέση, μη αφήνοντας κενές θέσεις στον πίνακα.
- Μια μέθοδο `clear` η οποία δεν δέχεται παραμέτρους, δεν επιστρέφει τιμές και αδειάζει τον πίνακα κρατήσεων του καλαθιού.
- Μια μέθοδο `getNumOfReservations` που δεν δέχεται παραμέτρους και επιστρέφει τον αριθμό των κρατήσεων στο καλάθι.
- Μια μέθοδο `getTotalCost` που δεν δέχεται παραμέτρους και επιστρέφει το κόστος (σε λεπτά) όλων των κρατήσεων στο καλάθι.

Κλάση Customer

Η κλάση Πελάτης έχει τα ακόλουθα ιδιωτικά πεδία:

- Ένα String `name`.
- Ένα ακέραιο που αντιπροσωπεύει το υπόλοιπο σε λεπτά (the balance in cents) του πελάτη.
- Ένα Καλάθι που περιέχει τις κρατήσεις που ο πελάτης θα ήθελε να κάνει.

Η κλάση πρέπει επίσης να έχει τις ακόλουθες δημόσιες μεθόδους:

- Έναν κατασκευαστή που παίρνει ως παράμετρο ένα String που υποδηλώνει το όνομα του πελάτη και έναν ακέραιο που αντιπροσωπεύει το αρχικό του υπόλοιπο. Ο κατασκευαστής χρησιμοποιεί τις εισόδους του και δημιουργεί ένα κενό Καλάθι για να αρχικοποιήσει τα αντίστοιχα πεδία.
- Μια μέθοδο `getName` και μια μέθοδο `getBalance` οι οποίες επιστρέφουν το όνομα και το υπόλοιπο (σε λεπτά) του πελάτη αντίστοιχα.
- Μια μέθοδο `getBasket` η οποία επιστρέφει την αναφορά (reference) στο καλάθι του πελάτη (δεν απαιτείται αντίγραφο του καλαθιού).
- Μια μέθοδο `addFunds` η οποία δέχεται έναν ακέραιο ως παράμετρο που αντιπροσωπεύει το ποσό των λεπτών που πρέπει να προστεθούν στο υπόλοιπο του πελάτη. Εάν η εισαγόμενη τιμή είναι αρνητική, η μέθοδος πρέπει να δίνει μια `IllegalArgumentException` με κατάλληλο μήνυμα. Διαφορετικά, η μέθοδος απλώς θα ενημερώσει το υπόλοιπο και θα επιστρέψει το νέο υπόλοιπο σε λεπτά.
- Μια μέθοδο `addToBasket` η οποία δέχεται μια Κράτηση ως παράμετρο και την προσθέτει στο καλάθι του πελάτη εάν το όνομα στην κράτηση ταιριάζει με το όνομα του πελάτη. Εάν η μέθοδος είναι επιτυχής, θα πρέπει να επιστρέψει τον αριθμό των κρατήσεων στο καλάθι αυτού του πελάτη. Διαφορετικά, η μέθοδος θα πρέπει να εκτοξεύσει μια `IllegalArgumentException`.
- Μια μέθοδο `addToBasket` η οποία δέχεται ένα Ξενοδοχείο, ένα String που αντιπροσωπεύει τον τύπο δωματίου, έναν ακέραιο που αντιπροσωπεύει τον αριθμό των νυχτών, και ένα boolean που αντιπροσωπεύει εάν ο πελάτης επιθυμεί να περιλαμβάνεται το πρωινό. Η μέθοδος προσθέτει την αντίστοιχη κράτηση στο καλάθι του πελάτη και επιστρέφει τον αριθμό των κρατήσεων που είναι τώρα στο καλάθι αυτού του πελάτη.
- Μια μέθοδο `addToBasket` η οποία δέχεται δύο Αεροδρόμια ως παραμέτρους. Η μέθοδος προσθέτει την αντίστοιχη κράτηση στο καλάθι του πελάτη και επιστρέφει τον αριθμό των κρατήσεων που είναι τώρα στο καλάθι τους, ανεξάρτητα από το εάν η κράτηση πτήσης δημιουργήθηκε επιτυχώς ή όχι.

- Μια μέθοδο `removeFromBasket` η οποία δέχεται μια Κράτηση ως είσοδο και την αφαιρεί από το καλάθι του πελάτη. Η μέθοδος επιστρέφει ένα `boolean` που υποδηλώνει εάν η λειτουργία ήταν επιτυχής ή όχι.
- Μια μέθοδο `checkout` η οποία δεν δέχεται παραμέτρους. Εάν το υπόλοιπο του πελάτη δεν είναι αρκετό για να καλύψει το συνολικό κόστος του καλαθιού τους, τότε η μέθοδος εκτοξεύει μια `IllegalStateException`. Διαφορετικά, ο πελάτης χρεώνεται το συνολικό κόστος του καλαθιού, το καλάθι αδειάζεται και το υπόλοιπο που απομένει επιστρέφεται.

III. Γενικές Οδηγίες

Δομή του προγράμματος

Είναι απαραίτητο να ακολουθήσετε την παρακάτω οργάνωση του προγράμματος:

- Project name: `Homeworks`
- Package name: `hw2`
- Class names: **Όπως σας ζητείται από την εκφώνηση!!!**
- Method signatures Όπως σας ζητείται από την εκφώνηση.
- Javadoc documentation για όλες τις κλάσεις και μεθόδους. **Να χρησιμοποιήσετε το λογισμικό τεκμηρίωσης javadoc** έτσι ώστε να μπορούμε να μετατρέψουμε τα σχόλια του προγράμματός σας σε HTML αρχεία και να τα δούμε με ένα browser.
- Σωστή εφαρμογή των Java Naming Conventions.
- Να ανεβάσετε στο Moodle όλα τα αρχεία java, ένα προς ένα, **όχι σε zip!**

Αξιολόγηση του προγράμματος

Το πρόγραμμα σας θα ελεγχθεί με δικό μας πρόγραμμα δοκιμής. Για αυτό είναι σημαντικό να ακολουθήσετε πιστά τις οδηγίες της εκφώνησης. Για να βεβαιωθείτε ότι το πρόγραμμα σας τρέχει κανονικά δημιουργήστε Junit test για την κάθε κλάση.

IV. Κριτήρια αξιολόγησης

Κλάση <code>Airport</code>	12 points
Κλάση <code>Room</code>	12 points
Κλάση <code>Hotel</code>	12 points
Κλάση <code>Reservation</code>	5 points
Κλάσεις <code>FlightReservation</code> , <code>HotelReservation</code> , <code>BnBReservation</code>	25 points
Κλάση <code>Basket</code>	18 points
Κλάση <code>Customer</code>	16 points
ΣΥΝΟΛΟ	100 points

Καλή επιτυχία