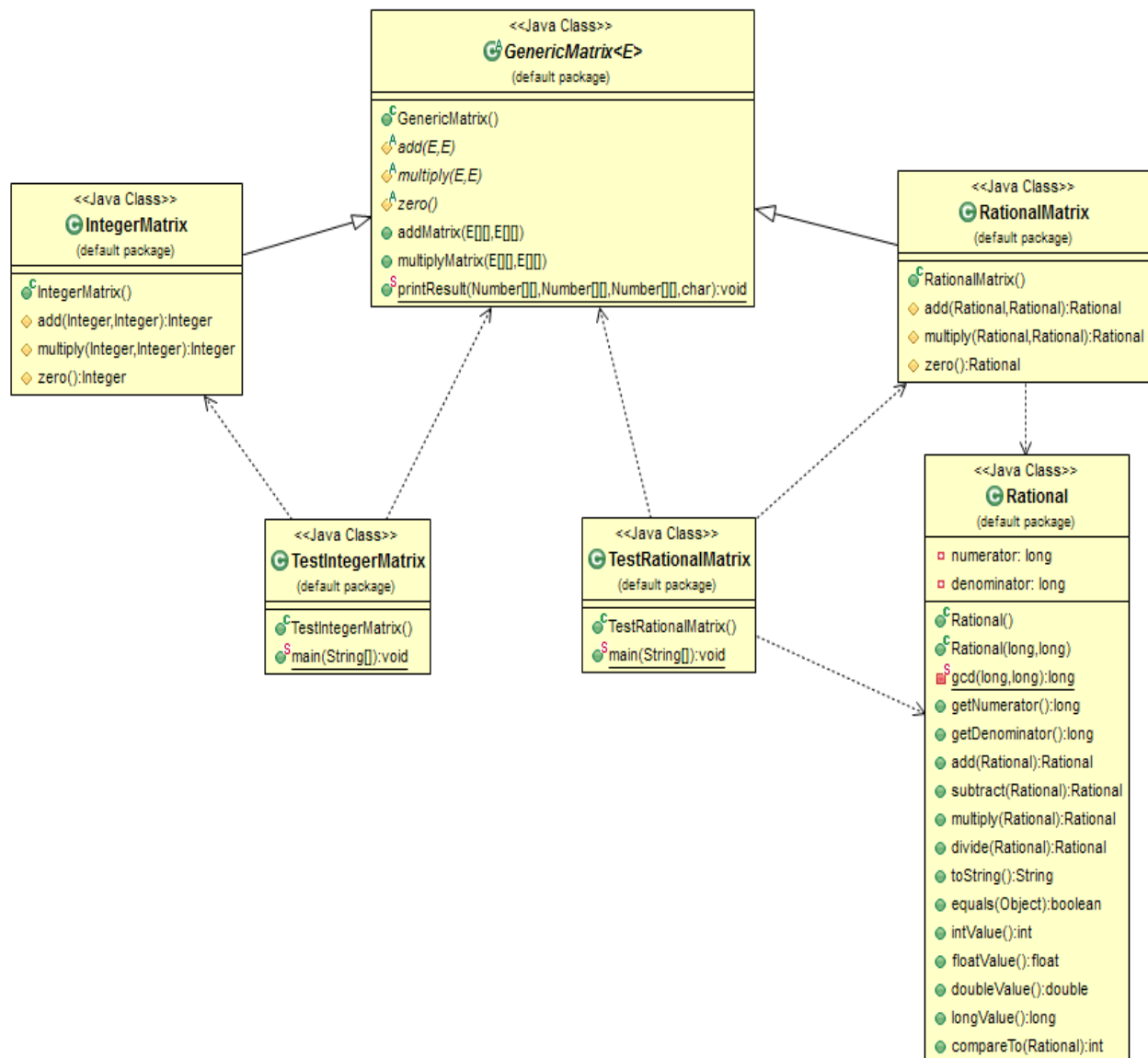


**ΕΡΓΑΣΤΗΡΙΟ 20****Generics**

1. Σχεδιάστε μια κλάση `GenericStack` που υλοποιεί μια στοίβα χρησιμοποιώντας ένα πίνακα. Πρέπει να ελέγχετε πάντα το μέγεθος της στοίβας πριν εισάγετε ένα νέο στοιχείο. Εάν η στοίβα είναι γεμάτη, τότε δημιουργήστε ένα νέο πίνακα με διπλάσιο μέγεθος και αντιγράψτε τα στοιχεία στο νέο πίνακα.
2. Στην προηγούμενη άσκηση, η κλάση `GenericStack` υλοποιήθηκε χρησιμοποιώντας σύνθεση (composition). Σχεδιάστε μια νέα κλάση `GenericStack` χρησιμοποιώντας κληρονομικότητα από την κλάση `ArrayList`.
3. Σε αυτήν την άσκηση πρέπει να σχεδιάσετε κλάσεις για πράξεις σχετικά με πίνακες χρησιμοποιώντας generics. Οι πράξεις πρόσθεση και πολλαπλασιασμός για όλους τους πίνακες είναι ίδιες με τη διαφορά ότι οι τύποι των στοιχείων τους διαφέρουν. Ως εκ τούτου, μπορείτε να σχεδιάσετε μια υπερκλάση (supperclass) που περιγράφει τις κοινές λειτουργίες που μοιράζονται οι πίνακες όλων των τύπων, ανεξάρτητα από τους τύπους στοιχείων τους, και μπορείτε να ορίσετε υποκλάσεις (subclass) προσαρμοσμένες σε συγκεκριμένους τύπους πινάκων. Δοκιμάστε της κλάσεις που σχεδιάσατε με διάφορους τύπους πινάκων.
4. Χρησιμοποιήστε την κλάση `Complex` και σχεδιάστε μια κλάση `ComplexMatrix` η οποία επιτρέπει πράξεις σε πίνακες με μιγαδικούς αριθμούς. Η κλάση `ComplexMatrix` πρέπει να κληρονομεί από τη κλάση `GenericMatrix` και να υλοποιεί τις μεθόδους `add`, `multiple`, και `zero`. Θα πρέπει να αλλάξετε την κλάση `GenericMatrix` και να αντικαταστήσετε κάθε παρουσίαση της κλάσης `Number` με την κλάση `Object`, επειδή η κλάση `Complex` δεν είναι subtype της κλάσης `Number`. Γράψτε ένα πρόγραμμα δοκιμών που δημιουργεί δύο πίνακες και εμφανίζει το αποτέλεσμα της πρόσθεσης και του πολλαπλασιασμού των πινάκων μέσω της κλήσης της μεθόδου `printResult`.



Διάγραμμα κλάσεων για την άσκηση 3:





Διάγραμμα κλάσεων για την άσκηση 4:

