

1. Τι είναι ο αντικειμενοστραφής προγραμματισμός;

Ο αντικειμενοστραφής προγραμματισμός είναι μια μεθοδολογία προγραμματισμού που βασίζεται στην ιδέα της οργάνωσης του κώδικα σε αντικείμενα, τα οποία αναπαριστούν πραγματικά ή αφηρημένα αντικείμενα με σχετικές ιδιότητες και συμπεριφορά.

2. Ποιες είναι οι βασικές αρχές του αντικειμενοστραφούς προγραμματισμού;

Οι βασικές αρχές του αντικειμενοστραφούς προγραμματισμού είναι:

- Ενθυλάκωση (Encapsulation): Η ιδέα ότι τα δεδομένα και οι συναρτήσεις που τα επεξεργάζονται πρέπει να είναι συσκευασμένα μαζί σε ένα αντικείμενο.

- Κληρονομικότητα (Inheritance): Η δυνατότητα να δημιουργούμε νέες κλάσεις βασισμένες σε υπάρχουσες κλάσεις, κληρονομώντας τις ιδιότητες και τις συμπεριφορές τους.

- Πολυμορφισμός (Polymorphism): Η ιδέα ότι μπορούμε να χρησιμοποιούμε ένα αντικείμενο της ίδιας κλάσης για να αναπαραστήσουμε διάφορες μορφές ή να χρησιμοποιούμε μια γενική διεπαφή για να αλληλεπιδράσουμε με αντικείμενα διάφορων κλάσεων.

3. Ποιο είναι το πλεονέκτημα του αντικειμενοστραφούς προγραμματισμού σε σχέση με τον διαδικαστικό προγραμματισμό;

Το πλεονέκτημα του αντικειμενοστραφούς προγραμματισμού είναι ότι παρέχει μια καλύτερη οργάνωση του κώδικα και τη δυνατότητα επαναχρησιμοποίησης του. Ο κώδικας γίνεται πιο κατανοητός και ευέλικτος, καθώς μπορούμε να αναπαραστήσουμε πραγματικά προβλήματα με αντικείμενα και να αλληλεπιδράσουμε με αυτά. Επιπλέον, η κληρονομικότητα και ο πολυμορφισμός επιτρέπουν την επέκταση και την ανακατασκευή του κώδικα χωρίς την ανάγκη για επαναγραφή του από την αρχή.

4. Τι είναι ένα αντικείμενο στον αντικειμενοστραφή προγραμματισμό;

Στον αντικειμενοστραφή προγραμματισμό, ένα αντικείμενο είναι ένα αναπαράσταση ενός συγκεκριμένου αντικειμένου ή έννοιας που έχει ιδιότητες (μεταβλητές) και συμπεριφορά (μεθόδους) που το προσδιορίζουν.

5. Τι είναι ένας τύπος δεδομένων;

Ένας τύπος δεδομένων είναι ένας ορισμός για τον τύπο των δεδομένων που μπορούν να αποθηκευτούν σε μια μεταβλητή. Στον αντικειμενοστραφή προγραμματισμό, οι τύποι δεδομένων συνήθως σχετίζονται με τις κλάσεις και τα αντικείμενα.

6. Ποια είναι η σημασία των κλάσεων στον αντικειμενοστραφή προγραμματισμό;

Οι κλάσεις είναι ο ορισμός για τα αντικείμενα που μπορούν να δημιουργηθούν. Οι κλάσεις καθορίζουν τις ιδιότητες και τις μεθόδους που έχουν τα αντικείμενα που ανήκουν σε αυτές. Είναι ο προτύπος από το οποίο δημιουργούνται τα αντικείμενα.

7. Τι είναι η κληρονομικότητα και ποια είναι η χρησιμότητά της;

Η κληρονομικότητα είναι η δυνατότητα μιας κλάσης να κληρονομήσει ιδιότητες και μεθόδους από μια άλλη κλάση. Αυτό επιτρέπει την ανακατασκευή και την επαναχρησιμοποίηση του κώδικα, καθώς οι κλάσεις που κληρονομούν μπορούν να προσθέσουν ή να αλλάξουν τη συμπεριφορά που κληρονομήθηκε.

8. Ποια είναι η διαφορά μεταξύ αντικειμένων και κλάσεων;

Ένα αντικείμενο είναι ένα συγκεκριμένο παράδειγμα μιας κλάσης, ενώ μια κλάση είναι ο γενικός ορισμός για τα αντικείμενα που μπορούν να δημιουργηθούν. Μια κλάση ορίζει τις ιδιότητες και τις μεθόδους που θα έχουν τα αντικείμενα που δημιουργούνται από αυτήν.

9. Τι είναι η πολυμορφισμός και πώς χρησιμοποιείται στον αντικειμενοστραφή προγραμματισμό;

Ο πολυμορφισμός είναι η ιδιότητα μιας μεθόδου να εκτελεί διαφορετικές ενέργειες ανάλογα με τον τύπο του αντικειμένου που την καλεί. Αυτό σημαίνει ότι η ίδια μέθοδος μπορεί να έχει διαφορετική συμπεριφορά ανάλογα με την κλάση του αντικειμένου που την καλεί. Αυτό επιτρέπει την ευελιξία και την ανακατασκευή του κώδικα, καθώς μπορούμε να χρησιμοποιήσουμε μια γενική διεπαφή για να αλληλεπιδράσουμε με διάφορες μορφές αντικειμένων.

10. Τι είναι το JVM;

Το JVM (Java Virtual Machine) είναι ένα περιβάλλον εκτέλεσης που εκτελεί τον κώδικα Java. Αναλαμβάνει τη μεταγλώττιση του κώδικα Java σε ενδιάμεσο μορφή, γνωστό ως bytecode, και εκτελεί αυτόν τον bytecode σε διαφορετικά λειτουργικά συστήματα.

11. Τι είναι το JRE;

Το JRE (Java Runtime Environment) είναι ένα πακέτο που περιλαμβάνει τον απαραίτητο εκτελέσιμο κώδικα και βιβλιοθήκες για να εκτελεστεί ένα πρόγραμμα Java. Περιλαμβάνει το JVM, τις βασικές κλάσεις της Java API (Application Programming Interface) και άλλες απαραίτητες εξαρτήσεις για την εκτέλεση εφαρμογών Java.

12. Ποια είναι η σχέση μεταξύ του JVM και του JRE;

Το JRE περιλαμβάνει το JVM ως μέρος του πακέτου του. Απλώς, το JRE παρέχει το πλήρες περιβάλλον εκτέλεσης για να τρέξει μια Java εφαρμογή, ενώ το JVM είναι υπεύθυνο για την πραγματική εκτέλεση του bytecode που παράγεται από τον κώδικα Java.

13. Ποια είναι η χρησιμότητα του JVM και του JRE;

Ο JVM επιτρέπει την πολυπλατφορμική εκτέλεση του κώδικα Java, καθώς μεταφράζει τον κώδικα σε bytecode, ανεξάρτητο από το λειτουργικό σύστημα. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να αναπτύξουν μια φορά και να εκτελέσουν τον κώδικα τους σε διάφορες πλατφόρμες. Το JRE παρέχει το περιβάλλον εκτέλεσης που απαιτείται για να τρέξει η Java εφαρμογή, παρέχοντας τις απαραίτητες κλάσεις και βιβλιοθήκες.

14. Ποια είναι η διαφορά μεταξύ του JRE και του JDK;

Το JDK (Java Development Kit) είναι ένα πακέτο που περιλαμβάνει το JRE καθώς και εργαλεία ανάπτυξης που απαιτούνται για να δημιουργήσετε και να αναπτύξετε εφαρμογές Java. Το JDK περιλαμβάνει τον μεταγλωττιστή της Java (javac), τον αρχείο εκτέλεσης Java (java) και άλλα εργαλεία ανάπτυξης, ενώ το JRE παρέχει απλώς το περιβάλλον εκτέλεσης για να τρέξει μια εφαρμογή Java.

15. Πώς μπορώ να εκτελέσω ένα πρόγραμμα Java που έχω γράψει;

Για να εκτελέσετε ένα πρόγραμμα Java, πρέπει να διασφαλίσετε ότι έχετε εγκαταστήσει το JRE (Java Runtime Environment) στον υπολογιστή σας. Στη συνέχεια, μπορείτε να ανοίξετε ένα τερματικό ή μια κονσόλα και να πληκτρολογήσετε την εντολή "java <όνομα_κλάσης>" για να εκτελέσετε το πρόγραμμα. Το "<όνομα_κλάσης>" πρέπει να είναι το όνομα της κύριας κλάσης στο πρόγραμμά σας.

16. Τι συμβαίνει κατά την εκτέλεση ενός προγράμματος Java;

Κατά την εκτέλεση ενός προγράμματος Java, το JVM (Java Virtual Machine) μεταγλωττίζει τον κώδικα Java σε bytecode, που είναι μια μορφή ενδιάμεσου κώδικα που μπορεί να εκτελεστεί από το JVM. Στη συνέχεια, το JVM εκτελεί τον bytecode και εκτελεί τις εντολές του προγράμματος γραμμένες στον κώδικα Java.

17. Ποιες είναι οι βασικές εντολές για την εκτέλεση ενός προγράμματος Java;

Για να εκτελέσετε ένα πρόγραμμα Java, οι βασικές εντολές που μπορείτε να χρησιμοποιήσετε είναι:

"javac" για να μεταγλωττίσετε τον κώδικα Java σε bytecode.

"java" για να εκτελέσετε το πρόγραμμα Java από το bytecode.

Αυτές οι εντολές πρέπει να εκτελεστούν από το τερματικό ή την κονσόλα του υπολογιστή σας.

18. Ποια είναι η κύρια κλάση ενός προγράμματος Java;

Η κύρια κλάση ενός προγράμματος Java είναι η κλάση που περιέχει τη μέθοδο "main". Η μέθοδος "main" είναι η εκκίνηση του προγράμματος και περιέχει τον κώδικα που θα εκτελεστεί πρώτος κατά την εκτέλεση του προγράμματος.

19. Τι είναι ένα String στην Java;

Απάντηση: Ένα String είναι μια ακολουθία χαρακτήρων στην Java. Αναπαριστάται από την κλάση "String" και μπορεί να περιέχει οποιονδήποτε συνδυασμό χαρακτήρων, όπως γράμματα, αριθμούς, σύμβολα και κενά διαστήματα.

20. Πώς μπορώ να δημιουργήσω ένα String στην Java;

Απάντηση: Μπορείτε να δημιουργήσετε ένα String στην Java χρησιμοποιώντας τον τελεστή ανάθεσης (=) και τα εισαγωγικά ("""). Για παράδειγμα, μπορείτε να γράψετε: String myString = "Hello, World!";

21. Πώς μπορώ να επιτύχω συνένωση (concatenation) δύο Strings;

Μπορείτε να συνενώσετε δύο Strings χρησιμοποιώντας τον τελεστή "+" ή τη μέθοδο concat().

21. Πώς μπορώ να αποκτήσω το μήκος ενός String;

Μπορείτε να αποκτήσετε το μήκος ενός String χρησιμοποιώντας τη μέθοδο length()

22. Τι είναι η κλάση Math στην Java;

Η κλάση Math στην Java παρέχει μια συλλογή από στατικές μεθόδους και στατικές μεταβλητές για μαθηματικούς υπολογισμούς. Μπορείτε να χρησιμοποιήσετε αυτές τις μεθόδους για να εκτελέσετε διάφορες πράξεις, όπως ακέραιες και πραγματικές πράξεις, εύρεση ελάχιστων και μέγιστων, υπολογισμός τριγωνομετρικών συναρτήσεων, κλπ.

23. Πώς μπορώ να χρησιμοποιήσω την κλάση Math στην Java;

Η κλάση Math είναι μια ενσωματωμένη κλάση, οπότε δεν απαιτείται η δημιουργία αντικειμένων. Μπορείτε να χρησιμοποιήσετε τις στατικές μεθόδους και μεταβλητές της κατευθείαν μέσω του ονόματος της κλάσης, π.χ. Math.methodName(). Παράδειγμα: double result = Math.sqrt(16);.

24. Ποιες είναι μερικές από τις συνήθεις μεθόδους της κλάσης Math;

Ορισμένες από τις συνήθεις μεθόδους της κλάσης Math περιλαμβάνουν:

abs(): Επιστρέφει την απόλυτη τιμή ενός αριθμού.

sqrt(): Επιστρέφει την τετραγωνική ρίζα ενός αριθμού.

pow(): Υπολογίζει τη δύναμη ενός αριθμού.

max(): Επιστρέφει τον μεγαλύτερο από δύο αριθμούς.

24. Τι είναι η δομή ελέγχου "if-else";

Η δομή ελέγχου "if-else" χρησιμοποιείται για να εκτελέσει μια συγκεκριμένη ενέργεια αν ένας όρος είναι αληθής (true), και μια διαφορετική ενέργεια αν ο όρος είναι ψευδής (false). Η δομή μπορεί να έχει μια εναλλακτική παράγραφο "else" για να εκτελέσει κώδικα όταν ο όρος είναι ψευδής.

25. Τι είναι η δομή ελέγχου "switch-case";

Η δομή ελέγχου "switch-case" χρησιμοποιείται για να εκτελέσει διαφορετικές ενέργειες βάσει της τιμής μιας μεταβλητής. Η δομή ελέγχου "switch" ελέγχει την τιμή της μεταβλητής και εκτελεί τον αντίστοιχο κώδικα στην περίπτωση που ταιριάζει με την τιμή.

26. Τι είναι ο ternary operator;

Απάντηση: Ο ternary operator είναι ένας τελεστής που χρησιμοποιείται για να εκτελέσει διαφορετική ενέργεια βάσει μιας συνθήκης. Έχει τη μορφή: condition ? expression1 : expression2. Αν η συνθήκη είναι αληθής, επιστρέφεται η τιμή της expression1, διαφορετικά επιστρέφεται η τιμή της expression2