



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

1. Πακέτο Εκπαιδευτικού Υλικού του Code4SP

WP3:

Εκπαιδευτικό Υλικό του
έργου Code4SP

Εκπονήθηκε από το:
CDDR



Πληροφορίες του Έργου

Ακρώνυμο του Έργου: Code4SP

Τίτλος Έργου: Κωδικοποίηση για Κοινωνική Ένταξη/ Ενσωμάτωση

Αριθμός Έργου: 621417-EPP-1-2020-1-PT-EPPKA3-IP1-SOC-IN

Ιστοσελίδα του έργου: www.code4sp.eu

Εταίρος εξουσιοδότησης: CodeDoor

Εκδοχή/Εκδοση Εγγράφου: 1.0

Ημερομηνία Προετοιμασίας: 25/02/2022

Ιστορικό Εγγράφου			
Ημερομηνία	Έκδοση/Εκδοχή	Συντάκτης	Περιγραφή
25/02/2022	1	CodeDoor	Προσχέδιο

Πίνακας περιεχομένων

Πληροφορίες σχετικά με το Έργο	4
Θέμα:	4
1. Προγραμματισμός Η/Υ και βασικές έννοιες	4
Προϋποθέσεις:	4
Φόρτος εργασίας:	4
Περιγραφή:	4
Μαθησιακά αποτελέσματα:	4
Απαιτούμενο υλικό:	5
Σχέδιο μαθήματος:	5
Υποθέματα:	5
Επιπρόσθετοι πόροι:	5
1.1. Εισαγωγή στους υπολογιστές και στον προγραμματισμό	6
Ποιοι είναι μερικοί τρόποι χρήσης των υπολογιστών;	6
Ποιες άλλες συσκευές θεωρούνται ως υπολογιστικά συστήματα;	7
Ποια λογισμικά έχετε χρησιμοποιήσει;	7
1.2. Οι έννοιες του Υλισμικού και του Λογισμικού	8
Ορισμοί Υλισμικού και Λογισμικού	8
1.3 Πώς αποθηκεύουν τα δεδομένα οι Η/Υ	9
Η έννοια του δυαδικού συστήματος	9
Διαδικαστικός και Αντικειμενοστραφής Προγραμματισμός	9
Αποθήκευση αριθμών, χαρακτήρων	9
ASCII	9
Unicode	9
UTF-8	9
Αριθμοί	10
Άλλοι τύποι δεδομένων	10
1.4. Πώς λειτουργεί ένα πρόγραμμα	11
Ο κύκλος της προσκόμισης - αποκωδικοποίησης - εκτέλεσης	11
Από την γλώσσα μηχανής στην συμβολική γλώσσα	12
Λέξεις Κλειδιά, Εκτελεστές και Σύνταξη: μια επισκόπηση	13
Μεταγλωττιστές και Διερμηνείς	14
1.5. Προγράμματα και Γλώσσες Προγραμματισμού	15

Τύποι γλωσσών προγραμματισμού	15
Από ένα πρόγραμμα υψηλού επιπέδου σε εκτελέσιμο αρχείο	16
IDEs (Ολοκληρωμένο Περιβάλλον Ανάπτυξης)	17
Τα κοινά στοιχεία των γλωσσών προγραμματισμού	17

Θέμα:

1. Προγραμματισμός Η/Υ και βασικές έννοιες

Προϋποθέσεις:

Βασικές γνώσεις ηλεκτρονικών υπολογιστών, εγκατάσταση βασικού λογισμικού και βασικές γνώσεις επεξεργασίας αρχείων.

Φόρτος εργασίας:

5 ώρες

Περιγραφή:

Στο υποκεφάλαιο αυτό, θα δώσουμε μια σύντομη εισαγωγή στον προγραμματισμό ηλεκτρονικών υπολογιστών και τις βασικές έννοιες που σχετίζονται με αυτόν. Θα αναλύσουμε έννοιες υλισμικού και λογισμικού ηλεκτρονικών υπολογιστών, καθώς και τον τρόπο με τον οποίο οι υπολογιστές επεξεργάζονται τα δεδομένα. Στο τέλος, θα ρίξουμε μια ματιά για το πώς λειτουργεί ο προγραμματισμός και τι είδους γλώσσες προγραμματισμού είναι διαθέσιμες.

Μαθησιακά αποτελέσματα:

- Οι εκπαιδευόμενοι θα είναι σε θέση να αναγνωρίζουν την έννοια της Γλώσσας Σήμανσης Υπερκειμένου/ HyperText Markup Language (HTML) από την οικογένεια των γλωσσών περιγραφής και επεξεργασίας εγγράφων.
- Θα μπορούν να διακρίνουν τη δομή, το περιεχόμενο και το στυλ Διάκριση της δομής, του περιεχομένου και τα στυλ μιας σελίδας.
- Θα μπορούν να χρησιμοποιούν το HTML για τον σχεδιασμό ιστοσελίδων στο διαδίκτυο.

Απαιτούμενα υλικά:

- Υπολογιστής ή φορητός υπολογιστής
- Σύνδεση στο Διαδίκτυο

Σενάριο μαθήματος:

Ο συνολικός χρόνος για αυτό το θέμα είναι 10 ώρες και θα εναπόκειται στον εκπαιδευτή να αποφασίσει πόσο χρόνο θα αφιερώσει στη διδασκαλία κάθε υποθέματος. Προκειμένου να αξιοποιήσουμε στο έπακρο όλο τον διαθέσιμο χρόνο, προτείνουμε τη χρήση του εκπαιδευτικού υλικού που συντάχθηκε στο πλαίσιο του έργου (παρουσιάσεις PPT), το οποίο σχεδιάστηκε με γνώμονα την αποτελεσματική χρήση του χρόνου. Αυτές οι παρουσιάσεις αποτελούνται από τα ακόλουθα στοιχεία:

- Ανάπτυξη του υποθέματος και των βασικών ιδεών που πρέπει να διατηρηθούν.
- Προτεινόμενες δραστηριότητες.

Επομένως, εάν ο εκπαιδευτής ακολουθήσει τη λογική σειρά των PPTs, σίγουρα θα μπορέσει να ολοκληρώσει τη συνεδρία εντός του καθορισμένου χρονικού ορίου. Αυτές οι παρουσιάσεις μπορούν επίσης να διατεθούν στους μαθητές για ατομική μελέτη.

Υποθέματα:

- 1.1. Εισαγωγή στους υπολογιστές και στον προγραμματισμό
- 1.2. Έννοια υλισμικού και λογισμικού
- 1.3. Πώς αποθηκεύουν τα δεδομένα οι υπολογιστές
- 1.4. Πώς λειτουργεί ένα πρόγραμμα
- 1.5. Προγράμματα και γλώσσες προγράμματος

Επιπλέον πόροι:

- [Khan Academy](#): χρήσιμες πηγές σε θέματα ηλεκτρονικών υπολογιστών σε διάφορες γλώσσες
- Marc Andreessen: <https://future.a16z.com/so-software-is-eating-the-world/>

1.1. Εισαγωγή στους υπολογιστές και στον προγραμματισμό

Οι υπολογιστές είναι μηχανές που μπορούν να προγραμματιστούν για την εκτέλεση μιας ακολουθίας εντολών. Ο προγραμματισμός είναι η διαδικασία σχεδιασμού αυτών των οδηγιών. Τα προγράμματα είναι γραμμένα σε μια συγκεκριμένη γλώσσα που παρέχει μια δομή για τον προγραμματιστή και χρησιμοποιεί συγκεκριμένες οδηγίες για τον έλεγχο της ακολουθίας των λειτουργιών που εκτελεί ο υπολογιστής.

«Software is eating the world» (Marc Andreessen, <https://future.a16z.com/software-is-eating-the-world/>), οπότε ενημερώστε τους μαθητές σας για το γεγονός ότι στις μέρες μας σχεδόν τα πάντα είναι μηχανογραφημένα και προγραμματισμένα (βλ. επίσης: [Internet of Things](#)).

Με ποιους τρόπους μπορεί κανείς να χρησιμοποιήσει τους ηλεκτρονικούς υπολογιστές;

Αφήστε τους μαθητές σας να κάνουν ένα καταιγισμό ιδεών ή να σχεδιάσουν ένα νοητικό χάρτη για το πώς χρησιμοποιούν στην πραγματικότητα τους υπολογιστές. Υπάρχουν πολλοί τρόποι με τους οποίους μπορείτε να χρησιμοποιήσετε τους ηλεκτρονικούς υπολογιστές στην καθημερινή σας ζωή. Ακολουθούν μερικοί από τους πιο συνηθισμένους τρόπους:

- Για να περιηγηθείτε στο διαδίκτυο
- Για να ελέγξετε το ηλεκτρονικό σας ταχυδρομείο
- Για να κάνετε έρευνα για ένα έργο
- Για να παίξετε παιχνίδια
- Για να παρακολουθήσετε ταινίες ή τηλεοπτικές εκπομπές
- Για να ακούσετε μουσική
- Για να επεξεργαστείτε ένα έγγραφο ή ένα λογιστικό φύλλο
- Για να διαβάσετε τις ειδήσεις

Ποιες άλλες συσκευές είναι υπολογιστικά συστήματα;

Οι υπολογιστές δεν είναι μόνο οι συσκευές που βρίσκονται στα γραφεία μας ή στις τσέπες μας. Είναι επίσης οι συσκευές που λειτουργούν τα αυτοκίνητα, τα αεροπλάνα και τα πλοία. Είναι στις τηλεοράσεις μας, στα ψυγεία μας, ακόμα και στα ρολόγια μας. Εν ολίγοις, οι υπολογιστές είναι παντού.

Τι είδους λογισμικό έχετε χρησιμοποιήσει;

Αφήστε τους μαθητές σας να κάνουν ένα καταιγισμό ιδεών ή να σχεδιάσουν ένα νοητικό χάρτη για το λογισμικό που έχουν ήδη χρησιμοποιήσει. Προσπαθήστε να τους κάνετε να συνειδητοποιήσουν ότι κάθε λογισμικό αναπτύχθηκε από κάποιον προγραμματιστή. Ακόμη και το λογισμικό που είναι εγκατεστημένο στα smartphone κινητά, στα μηχανήματα αυτόματης ανάληψης μετρητών ή ακόμα και στις τηλεοράσεις που χρησιμοποιούμε καθημερινά.

1.2. Οι έννοιες του υλισμικού και του λογισμικού

Ορισμός του υλισμικού και του λογισμικού

Το υλισμικό αναφέρεται στα φυσικά στοιχεία ενός συστήματος υπολογιστή, ενώ το λογισμικό αναφέρεται στις εντολές και τα δεδομένα που κάνουν τον υπολογιστή να λειτουργεί.

Τα κύρια μέρη ενός υπολογιστή και οι λειτουργίες τους

Ένας υπολογιστής αποτελείται από 4 βασικά μέρη: την κεντρική μονάδα επεξεργασίας (CPU), τη μνήμη, τις συσκευές εισόδου και τις συσκευές εξόδου. Η κεντρική μονάδα επεξεργασίας/μητρική πλακέτα (CPU) είναι το μέρος του υπολογιστή που εκτελεί τους υπολογισμούς και ελέγχει τα άλλα μέρη. Η κύρια μνήμη είναι το μέρος όπου ο υπολογιστής αποθηκεύει τα δεδομένα που επεξεργάζεται. Οι συσκευές εισόδου είναι οι συσκευές εκείνες που χρησιμοποιεί ο χρήστης για την εισαγωγή δεδομένων στον υπολογιστή, όπως το πληκτρολόγιο και το ποντίκι. Οι συσκευές εξόδου είναι οι συσκευές εκείνες που χρησιμοποιεί ο υπολογιστής για να εμφανίζει τα αποτελέσματα των υπολογισμών του, όπως το μόνιτορ και ο εκτυπωτής.

1.3. Πώς αποθηκεύουν τα δεδομένα οι υπολογιστές

Έννοια: το δυαδικό σύστημα

Ένα δυαδικό σύστημα είναι ένας τρόπος αναπαράστασης πληροφοριών χρησιμοποιώντας δύο σύμβολα: το 0 και το 1. Το δυαδικό σύστημα είναι η απλούστερη μορφή αναπαράστασης πληροφοριών και χρησιμοποιείται στα υπολογιστικά συστήματα για την αποθήκευση και την επεξεργασία πληροφοριών. Το δυαδικό σύστημα είναι ένας εύχρηστος τρόπος αναπαράστασης πληροφοριών επειδή είναι πολύ απλός και μπορεί να επεξεργαστεί πολύ εύκολα από τα υπολογιστικά συστήματα.

Αποθήκευση αριθμών, χαρακτήρων

ASCII

Τα υπολογιστικά συστήματα αποθηκεύουν κείμενο και αριθμούς με διάφορους τρόπους, ο καθένας από τους οποίους παρουσιάζει τα δικά του πλεονεκτήματα ή μειονεκτήματα. Ο πιο συνηθισμένος τρόπος αποθήκευσης κειμένου είναι υπό την μορφή χαρακτήρων ASCII (Αμερικανικός Πρότυπος Κώδικας για Ανταλλαγή Πληροφοριών) Στο ASCII, κάθε χαρακτήρας αντιπροσωπεύεται από έναν αριθμό, από το 0 μέχρι το 127. Αυτός ο αριθμός ονομάζεται κώδικας ASCII. Όταν ένας υπολογιστής αποθηκεύει ένα κείμενο υπό την μορφή χαρακτήρων ASCII, απλά αποθηκεύει τους κώδικες ASCII για κάθε χαρακτήρα στο κείμενο.

Κωδικοποίηση Unicode

Ένας άλλος τρόπος για να αποθηκεύσετε ένα κείμενο είναι υπό την μορφή χαρακτήρων Unicode. Η κωδικοποίηση Unicode είναι ένα διεθνές πρότυπο που ορίζει έναν μοναδικό αριθμό για κάθε χαρακτήρα σε κάθε γλώσσα. Όταν ένας υπολογιστής αποθηκεύει ένα κείμενο υπό τη μορφή χαρακτήρων Unicode, αποθηκεύει τον κώδικα Unicode για κάθε χαρακτήρα στο κείμενο.

UTF-8

Το UTF-8 είναι μια κωδικοποίηση χαρακτήρων που μπορεί να αποθηκεύσει ένα κείμενο και αριθμούς σε έναν μοναδικό χαρακτήρα Unicode. Αυτή η κωδικοποίηση χρησιμοποιείται ευρέως στο διαδίκτυο επειδή μπορεί να κωδικοποιήσει όλους τους χαρακτήρες σε ένα ευρύ φάσμα γλωσσών. Το UTF-8 είναι μια κωδικοποίηση μεταβλητού μήκους, που σημαίνει ότι μπορεί να κωδικοποιήσει χαρακτήρες διαφορετικών μεγεθών. Η μικρότερη κωδικοποίηση είναι 1 ψηφιολέξη, και η μεγαλύτερη κωδικοποίηση είναι 4 ψηφιολέξεις. Αυτή η κωδικοποίηση είναι αντιστρόφως συμβατή με την κωδικοποίηση ASCII, που σημαίνει ότι το κείμενο ASCII θα κωδικοποιηθεί σε UTF-8 χρησιμοποιώντας 1 ψηφιολέξη.

Αριθμοί

Ο πιο συνηθισμένος τρόπος αποθήκευσης αριθμών είναι σε ακέραια δυαδική μορφή. Στο δυαδικό σύστημα, κάθε αριθμός αντιπροσωπεύεται από μια συμβολοσειρά 0s και 1s. Για παράδειγμα, ο αριθμός 12 μπορεί να αναπαρασταθεί ως 01001000. Ο αριθμός 12 μπορεί επίσης να αναπαρασταθεί σε δεκαεξαδικό σύστημα, το οποίο είναι ένα σύστημα αρίθμησης με βάση το 16. Στο δεκαεξαδικό σύστημα, κάθε αριθμός αντιπροσωπεύεται από μια σειρά δεκαεξαδικών ψηφίων. Για παράδειγμα, ο αριθμός 12 μπορεί να αναπαρασταθεί ως C. Όταν ένας υπολογιστής αποθηκεύει έναν αριθμό στο δυαδικό ή δεκαεξαδικό σύστημα, αποθηκεύει την ακέραια τιμή του αριθμού.

Άλλοι τύποι δεδομένων

Οι υπολογιστές αναφέρονται συχνά ως ψηφιακές συσκευές. Ο όρος ψηφιακός μπορεί να χρησιμοποιηθεί για να περιγράψει οτιδήποτε χρησιμοποιεί δυαδικούς αριθμούς. Τα δυαδικά δεδομένα είναι δεδομένα τα οποία αποθηκεύονται σύμφωνα με το δυαδικό σύστημα και μια ψηφιακή συσκευή είναι οποιαδήποτε συσκευή που λειτουργεί με δυαδικά δεδομένα. Στο κεφάλαιο αυτό, έχουμε συζητήσει για το πώς οι αριθμοί και οι χαρακτήρες αποθηκεύονται σε δυαδική μορφή, αλλά οι υπολογιστές λειτουργούν επίσης με πολλούς άλλους τύπους ψηφιακών δεδομένων. Για παράδειγμα, σκεφτείτε τις φωτογραφίες που βγάζετε με την ψηφιακή σας κάμερα. Αυτές οι εικόνες αποτελούνται από μικροσκοπικές κουκκίδες χρώματος γνωστές και ως εικονοστοιχεία. (Ο όρος pixel σημαίνει εικονοστοιχείο.) Κάθε εικονοστοιχείο σε μια εικόνα μετατρέπεται σε έναν αριθμητικό κώδικα που αντιπροσωπεύει το χρώμα του εικονοστοιχείου. Ο αριθμητικός κώδικας αποθηκεύεται στη μνήμη ως δυαδικός αριθμός.

Η μουσική που παίζετε στο CD player σας, iPod ή MP3 player είναι επίσης ψηφιακή. Ένα ψηφιακό τραγούδι αποτελείται από μικρά κομμάτια μουσικής που ονομάζονται δείγματα ή ντέμο. Κάθε δείγμα/ντέμο μετατρέπεται σε δυαδικό αριθμό, ο οποίος μπορεί να αποθηκευτεί στη μνήμη ενός υπολογιστή. Όσο περισσότερα δείγματα/ντέμο έχει ένα τραγούδι, τόσο περισσότερα θα μοιάζει με την πρωτότυπη μουσική που γράφτηκε όταν

αναπαράγεται. Ένα CD τραγούδι υψηλής ποιότητας έχει περισσότερα από 44.000 δείγματα/ντέμο ανά δευτερόλεπτο!

1.4. Πώς λειτουργεί ένα πρόγραμμα

Υπάρχουν πολλοί διαφορετικοί τύποι προγραμμάτων ηλεκτρονικών υπολογιστών, αλλά όλα παρουσιάζουν τα ίδια βασικά στοιχεία: μια διεπαφή χρήστη, έναν επεξεργαστή και η μνήμη. Η διεπαφή χρήστη επιτρέπει στο χρήστη να εισάγει πληροφορίες και οδηγίες στο πρόγραμμα, ο επεξεργαστής εκτελεί τις οδηγίες και η μνήμη αποθηκεύει το πρόγραμμα και τα δεδομένα που επεξεργάζεται. Τα περισσότερα προγράμματα ηλεκτρονικών υπολογιστών είναι γραμμένα σε μια γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία είναι μια γλώσσα που έχει σχεδιαστεί για να μπορούν να την διαβάζουν και να την γράφουν εύκολα οι άνθρωποι. Ωστόσο, ο επεξεργαστής μπορεί να κατανοήσει μόνο τον κώδικα της μηχανής, ο οποίος είναι μια σειρά από μονάδες και μηδενικά. Έτσι, πριν ένα πρόγραμμα μπορεί να εκκινήσει τη λειτουργία του, πρέπει να μετατραπεί σε κώδικα μηχανής. Αυτό γίνεται από ένα πρόγραμμα που ονομάζεται μεταγλωττιστής ή μεταφραστής. Ο μεταγλωττιστής διαβάζει το πρόγραμμα και το μετατρέπει σε κώδικα μηχανής. Στη συνέχεια αποθηκεύει τον κώδικα του μηχανήματος σε ένα αρχείο που ονομάζεται εκτελέσιμο. Όταν ο χρήστης εκτελεί/τρέχει το πρόγραμμα, το εκτελέσιμο φορτώνεται στη μνήμη και ο επεξεργαστής εκτελεί τις εντολές.

Ο κύκλος προσκόμισης – αποκωδικοποίησης – εκτέλεσης

Ο κύκλος προσκόμισης - αποκωδικοποίησης - εκτέλεσης είναι μια βασική διαδικασία που χρησιμοποιεί ένας ηλεκτρονικός υπολογιστής για την εκτέλεση εντολών. Ο κύκλος ξεκινά όταν ο υπολογιστής λαμβάνει μια εντολή από τη μνήμη. Στη συνέχεια, αποκωδικοποιεί την οδηγία για να προσδιορίσει αυτό που πρέπει να εκτελέσει. Τέλος, εκτελεί τις οδηγίες. Ο κύκλος στη συνέχεια επαναλαμβάνεται, λαμβάνοντας την επόμενη οδηγία από τη μνήμη.

Από τη γλώσσα της μηχανής στη συμβολική γλώσσα

Καθώς ο προγραμματισμός στη γλώσσα μηχανής, που αποτελείται μόνο από δυαδικό κώδικα, είναι πολύ περίπλοκος για τον άνθρωπο, υπάρχει επίσης και η συμβολική γλώσσα ή γλώσσα assembly. Η συμβολική γλώσσα είναι μια γλώσσα προγραμματισμού χαμηλού επιπέδου για έναν υπολογιστή, μικροεπεξεργαστή ή άλλη προγραμματιζόμενη συσκευή, στην οποία ο προγραμματιστής χρησιμοποιεί εντολές στη συμβολική γλώσσα για τον έλεγχο της λειτουργίας της συσκευής. Η συμβολική γλώσσα πρέπει να είναι συγκεκριμένη για να ανταποκρίνεται σε ένα συγκεκριμένο μικροεπεξεργαστή ή οικογένεια μικροεπεξεργαστών. Αποτελείται από μια σειρά μνημονικών κωδικών, συμβολικών ονομάτων για τις λειτουργίες που μπορεί να εκτελέσει ο μικροεπεξεργαστής, και τους χειριστές (δεδομένα) πάνω στους οποίους πρόκειται να εκτελεστούν αυτές οι λειτουργίες. Η συμβολική γλώσσα μετατρέπεται σε κώδικα μηχανής, μια μορφή δυαδικού κώδικα που είναι συγκεκριμένη για έναν συγκεκριμένο τύπο υπολογιστή και μπορεί να γίνει κατανοητή από τον επεξεργαστή του υπολογιστή. Ο κώδικας μηχανής είναι η μόνη μορφή κώδικα που μπορεί να εκτελέσει άμεσα ο επεξεργαστής.

Ο προγραμματισμός της συμβολικής γλώσσας είναι ευκολότερος από τον προγραμματισμό της γλώσσας μηχανής γιατί ο τελευταίος δεν είναι αρκετά ταχύς και αποτελεσματικός ως προς την ανάγνωση ή την παραγωγή ενός πηγαίου κωδικού. Γι' αυτό λοιπόν το λόγο, δημιουργήθηκαν υψηλού επιπέδου γλώσσες προγραμματισμού (όπως C# ή python).

Σήμερα υπάρχουν πολλές γλώσσες προγραμματισμού υψηλού επιπέδου. Μερικά από τα πιο κοινά είναι Java, Python, και Ruby. Οι γλώσσες προγραμματισμού υψηλού επιπέδου είναι πιο εύχρηστες από τις γλώσσες προγραμματισμού χαμηλού επιπέδου. Αυτές οι γλώσσες επιτρέπουν στο χρήστη να επικεντρωθεί μόνο στην βασική αποστολή του και όχι στις λεπτομέρειες του υπολογιστή. Αυτό τις καθιστά ιδανικές για τη δημιουργία εφαρμογών και προγραμμάτων. Οι γλώσσες προγραμματισμού υψηλού

επιπέδου τείνουν επίσης να είναι πιο «επιεικείς» σε σύγκριση με τις γλώσσες προγραμματισμού χαμηλού επιπέδου. Εάν κάνετε ένα λάθος κατά την εγγραφή ενός κώδικα σε μια γλώσσα υψηλού επιπέδου, ο μεταγλωττιστής θα είναι συνήθως σε θέση να το διορθώσει για σας. Αυτό θα σας βοηθήσει να εξοικονομήσετε σημαντικό χρόνο ή ταλαιπωρία κατά την κωδικοποίηση.

Λέξεις-κλειδιά, Χειριστές/Εκτελεστές, και Σύνταξη: μια επισκόπηση

Σήμερα υπάρχουν πολλές γλώσσες προγραμματισμού υψηλού επιπέδου. Η καθεμιά διαθέτει το δικό της μοναδικό σύνολο λέξεων-κλειδιών, εκτελεστών και σύνταξης. Για να είστε αποτελεσματικοί με μια γλώσσα προγραμματισμού υψηλού επιπέδου, είναι σημαντικό να είστε εξοικειωμένοι με τις συγκεκριμένες λέξεις-κλειδιά, τους χειριστές/εκτελεστές και τη σύνταξη που χρησιμοποιείται από την εκάστοτε γλώσσα ξεχωριστά. Μερικές από τις πιο κοινές λέξεις-κλειδιά που χρησιμοποιούνται στις γλώσσες προγραμματισμού υψηλού επιπέδου περιλαμβάνουν: *αν, στη συνέχεια, διαφορετικά, ενώ, για, κάνε, σπάσε, συνέχισε*. Αυτές οι λέξεις-κλειδιά χρησιμοποιούνται για τον έλεγχο της ροής της εκτέλεσης του προγράμματος. Οι χειριστές/εκτελεστές είναι σύμβολα που αντιπροσωπεύουν λειτουργίες που μπορούν να εκτελεστούν σε τιμές. Οι συνηθέστεροι χειριστές/εκτελεστές περιλαμβάνουν: *+* (προσθήκη), *-* (αφαίρεση), *** (πολλαπλασιασμός), */* (διαίρεση) και *%* (ποσοστό). Αυτοί οι εκτελεστές μπορούν να χρησιμοποιηθούν για τον υπολογισμό των αποτελεσμάτων των εκφράσεων. Η σύνταξη μιας γλώσσας προγραμματισμού είναι το σύνολο των κανόνων που διέπουν τον τρόπο με τον οποίο πρέπει να γραφτεί ο κώδικας για να ερμηνευτεί από τον μεταγλωττιστή ή τον διερμηνέα. Η σύνταξη μιας γλώσσας προγραμματισμού υψηλού επιπέδου είναι συνήθως πιο επιεικής από τη σύνταξη μιας γλώσσας χαμηλότερου επιπέδου. Αυτό μπορεί να διευκολύνει τους αρχάριους να μάθουν να προγραμματίζουν.

Μεταγλωττιστές και διερμηνείς

Οι μεταγλωττιστές και οι διερμηνείς υπολογιστών είναι σημαντικά εργαλεία για τους προγραμματιστές λογισμικού. Ένας μεταγλωττιστής λαμβάνει έναν κώδικα γραμμένο σε μια γλώσσα και τον μετατρέπει σε κώδικα που μπορεί να εκτελεστεί σε μια διαφορετική

μηχανή. Ένας διερμηνέας λαμβάνει έναν κώδικα γραμμένο σε μία γλώσσα και τον εκτελεί όπως είναι, χωρίς να τον συντάξει πρώτα. Οι μεταγλωττιστές χρησιμοποιούνται συνήθως για γλώσσες που έχουν μεγάλη δομή, όπως το C ή το Java. Οι διερμηνείς χρησιμοποιούνται συνήθως για γλώσσες που είναι πιο ευέλικτες, όπως το Python ή το Ruby. Οι μεταγλωττιστές συνήθως παράγουν ταχύτερο κώδικα από τους διερμηνείς. Ωστόσο, οι διερμηνείς είναι συνήθως πιο φορητοί, πράγμα που σημαίνει ότι μπορούν να τρέξουν σε περισσότερους τύπους μηχανών. Η επιλογή μεταξύ των δύο εξαρτάται πάντα από την περίπτωση/ περίπτωση. Εάν θεωρείται ότι είναι σημαντικότερη για τις ανάγκες σας, τότε ένας μεταγλωττιστής είναι καταλληλότερος. Εάν η φορητότητα σας είναι πιο σημαντική, τότε ο διερμηνέας είναι καλύτερη επιλογή.

1.5. Προγράμματα και γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού χρησιμεύουν στη δημιουργία προγραμμάτων, τα οποία χρησιμοποιούνται για τον έλεγχο της συμπεριφοράς μιας μηχανής, συνήθως ενός υπολογιστή. Μια γλώσσα προγραμματισμού παρέχει τη δομή στον προγραμματιστή για να δώσει τις εντολές στη μηχανή αλλά και για επικοινωνήσει αυτές τις εντολές σε άλλους προγραμματιστές. Υπάρχουν πολλές γλώσσες προγραμματισμού που χρησιμοποιούνται σήμερα. Οι πιο δημοφιλείς είναι οι C, Java, Python και JavaScript.

Τύποι γλωσσών προγραμματισμού

Υπάρχουν δεκάδες γλώσσες προγραμματισμού που χρησιμοποιούνται σήμερα, αλλά μπορούν να ταξινομηθούν ευρέως σε πέντε βασικές κατηγορίες:

- **Γλώσσες προγραμματισμού χαμηλού επιπέδου:** Αυτές οι γλώσσες προγραμματισμού είναι πολύ κοντά στην γλώσσα μηχανής και χρησιμοποιούνται για τον προγραμματισμό μικροεπεξεργαστών και άλλων συσκευών χαμηλού επιπέδου. Δεν είναι εύληπτες και γι' αυτό δεν είναι τόσο δημοφιλείς όσον αφορά τον προγραμματισμό για γενικούς σκοπούς. Παραδείγματα: Συμβολική γλώσσα, γλώσσα προγραμματισμού C και συμβολική γλώσσα χαμηλού επιπέδου.
- **Γλώσσες προγραμματισμού υψηλού επιπέδου:** Αυτές οι γλώσσες προγραμματισμού έχουν σχεδιαστεί για να είναι εύληπτες και εύχρηστες. Είναι δημοφιλείς για τον προγραμματισμό για γενικούς σκοπούς. Παραδείγματα: Java, C++ και Python.
- **Γλώσσες σεναρίου:** Οι γλώσσες σεναρίου έχουν σχεδιαστεί για να είναι εύκολες στη χρήση και είναι δημοφιλείς για σκοπούς σεναρίου. Παραδείγματα: Python, Ruby και JavaScript.
- **Γλώσσες ανά τομέα:** Οι γλώσσες ανά τομέα έχουν σχεδιαστεί για μια συγκεκριμένη εργασία ή κλάδο. Δεν είναι εύληπτες και γι' αυτό δεν είναι τόσο δημοφιλείς όσον αφορά τον προγραμματισμό για γενικούς σκοπούς. Παραδείγματα: MATLAB, SQL και FORTRAN.

- **Αντικειμενοστραφείς γλώσσες προγραμματισμού:** Αυτές οι γλώσσες προγραμματισμού βασίζονται σε ένα πρότυπο αντικειμενοστραφούς προγραμματισμού. Παραδείγματα: Java, C++ και Python.

Από ένα πρόγραμμα υψηλού επιπέδου σε ένα εκτελέσιμο αρχείο

Όταν ένα πρόγραμμα υπολογιστή είναι γραμμένο σε μια γλώσσα υψηλού επιπέδου, μεταφράζεται πρώτα σε μια γλώσσα χαμηλότερου επιπέδου, η οποία μπορεί να γίνει πιο εύκολα κατανοητή από τις μηχανές. Η γλώσσα χαμηλότερου επιπέδου μεταγλωττίζεται στη συνέχεια σε ένα εκτελέσιμο αρχείο, το οποίο μπορεί να εκτελεστεί σε έναν υπολογιστή.

IDEs (Ολοκληρωμένο Περιβάλλον Ανάπτυξης)

Ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης («Integrated Development Environment», IDE) είναι μία εφαρμογή λογισμικού η οποία παρέχει εκτεταμένες διευκολύνσεις στους προγραμματιστές ηλεκτρονικών υπολογιστών για την ανάπτυξη λογισμικών. Ένα IDE αποτελείται συνήθως από έναν επεξεργαστή πηγαίου κώδικα, εργαλεία αυτοματισμού κατασκευής και έναν αποσφαλματωτή. Ο επεξεργαστής πηγαίου κώδικα επιτρέπει στον προγραμματιστή να γράψει κώδικα, ενώ τα εργαλεία αυτοματοποίησης κατασκευής αυτοματοποιούν τη διαδικασία σύνταξης αυτού του κώδικα σε μια μορφή που μπορεί να τρέξει ο υπολογιστής. Ο αποσφαλματωτής επιτρέπει στον προγραμματιστή να περάσει μέσα από τον κώδικα, εξετάζοντας την κατάσταση του προγράμματος σε κάθε σημείο της εκτέλεσής του. Τα IDE χρησιμοποιούνται συχνά σε συνδυασμό με ένα σύστημα ελέγχου έκδοσης, το οποίο επιτρέπει σε διαφορετικούς προγραμματιστές που εργάζονται πάνω στην ίδια εργασία να μοιράζονται και να συγχωνεύουν τις αλλαγές τους απρόσκοπτα.

Τα κοινά στοιχεία στις γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού υπολογιστών παρουσιάζουν ορισμένα κοινά στοιχεία, παρά τις διαφορές τους. Όλες οι γλώσσες προγραμματισμού έχουν έναν τρόπο να

αντιπροσωπεύουν τις εντολές στον υπολογιστή σε μια μορφή που ο υπολογιστής μπορεί να καταλάβει. Αυτό συνήθως ονομάζεται κώδικας, ή πηγαίος κώδικας. Οι προγραμματιστές χρησιμοποιούν κώδικες για να δημιουργήσουν προγράμματα και εφαρμογές λογισμικού. Όλες οι γλώσσες προγραμματισμού έχουν επίσης έναν τρόπο οργάνωσης των εντολών έτσι ώστε να μπορούν να επαναχρησιμοποιηθούν, να τροποποιηθούν ή να μοιραστούν με άλλους προγραμματιστές. Αυτό συνήθως ονομάζεται βιβλιοθήκη ή άρθρωμα. Οι βιβλιοθήκες και τα αρθρώματα επιτρέπουν στους προγραμματιστές να δημιουργούν σύνθετα προγράμματα με βάση το έργο άλλων προγραμματιστών. Τέλος, όλες οι γλώσσες προγραμματισμού έχουν έναν τρόπο να μεταφέρουν πληροφορίες στον χρήστη σχετικά με το τι κάνει το πρόγραμμα και πώς λειτουργεί. Αυτό συνήθως ονομάζεται έξοδος ή αποσφαλμάτωση πληροφοριών. Οι πληροφορίες εξόδου και αποσφαλμάτωσης βοηθούν τους προγραμματιστές να κατανοήσουν και να διορθώσουν τα προβλήματα που παρουσιάζουν τα προγράμματά τους.

Διαδικαστικός και αντικειμενοστραφής προγραμματισμός

Υπάρχουν δύο κύριοι τύποι προγραμματισμού: ο διαδικαστικός και ο αντικειμενοστραφής. Ο διαδικαστικός προγραμματισμός περιλαμβάνει μια διαδικασία βήμα προς βήμα, ενώ ο αντικειμενοστραφής προγραμματισμός περιλαμβάνει τη δημιουργία αντικειμένων που αλληλεπιδρούν μεταξύ τους. Ο διαδικαστικός προγραμματισμός θεωρείται συχνά απλούστερος από τον αντικειμενοστραφή προγραμματισμό. Είναι εύκολο να μάθετε τα βήματα που απαιτούνται για να ολοκληρώσετε μια εργασία και είναι εύκολο να αλλάξετε τη σειρά αυτών των βημάτων χωρίς να επηρεάσετε το αποτέλεσμα. Ωστόσο, ο διαδικαστικός προγραμματισμός μπορεί να είναι λιγότερο αποτελεσματικός επειδή μπορεί να είναι δύσκολη η επαναχρησιμοποίηση κώδικα που έχει συνταχθεί για μια συγκεκριμένη εργασία. Ο αντικειμενοστραφής προγραμματισμός είναι πιο περίπλοκος από τον διαδικαστικό προγραμματισμό, αλλά επιτρέπει μεγαλύτερη ευελιξία και επαναχρησιμοποίηση του κώδικα. Τα αντικείμενα μπορούν να δημιουργηθούν για συγκεκριμένες εργασίες και στη συνέχεια να επαναχρησιμοποιηθούν όπως απαιτείται. Επιπλέον, ο αντικειμενοστραφής

κώδικας είναι συχνά ευκολότερο να διαβαστεί και να κατανοηθεί από ό, τι ο διαδικαστικός κώδικας. Ωστόσο, ο αντικειμενοστραφής προγραμματισμός μπορεί να είναι πιο δύσκολο να διδαχθεί και μπορεί να είναι λιγότερο αποτελεσματικός από τον διαδικαστικό προγραμματισμό.