1. **ΔΙΑΔΡΑΣΗ**
2. Τι είναι η **διάδραση**;

Διάδραση είναι η επικοινωνία ανθρώπου-υπολογιστή

1. Ποιοι οι **όροι** της **διάδρασης** ;
   1. Ένα πεδίο εφαρμογής ορίζει μία περιοχή γνώσης και εμπειρίας σε κάποια δραστηριότητα του πραγματικού κόσμου. π.χ. Σχεδίαση γραφικών.
   2. Ένας στόχος είναι η επιθυμητή έξοδος από την εκτέλεση μιας εργασίας, π.χ. δημιουργία ενός σχήματος
   3. Οι εργασίες είναι λειτουργικές για το χειρισμό των βασικών εννοιών ενός πεδίου εφαρμογής, π.χ. κατασκευή ενός σχήματος
2. Τι είναι η **ανάδραση** ;
   1. Ανάδραση είναι η ενημέρωση του χρήστη από τον υπολογιστή.
3. Ποια είναι τα **βασικά μοντέλα διάδρασης** ; Πείτε λίγα λόγια για αυτά.
   1. Ο κύκλος εκτέλεσης – αξιολόγησης του **Norman**
      1. Ορισμός στόχου (δημιουργία περισσότερου φωτός)
      2. Σχηματισμός πρόθεσης (άναμμα φωτιστικού)
      3. Προσδιορισμός της αλληλουχίας ενεργειών (να μεταβείτε στο φωτιστικό και να το ανάψετε)
      4. Εκτέλεση των ενεργειών
      5. Αντίληψη της κατάστασης του συστήματος (άναψε το φως ;)
      6. Ερμηνεία της κατάστασης του συστήματος (αν δεν ανάψει, έχει καεί η λάμπα)
      7. Αξιολόγηση της κατάστασης του συστήματος βάσει των στόχων και των προσπαθειών (είναι αρκετό το φως ;)
   2. Το πλαίσιο αναφοράς της διάδρασης **Abowd & Beale**. Το πλαίσιο αναφοράς είναι προέκταση του μοντέλου διάδρασης Norman. Αποτελείται από 4 μέρη :
      1. Χρήστης
      2. Είσοδος
      3. Σύστημα
      4. Έξοδος

Κάθε συστατικό έχει τη δική του γλώσσα διάδραση → μετάφραση μεταξύ των συστατικών

Διεπιφάνεια = Είσοδος + έξοδος

Προβλήματα στη διάδραση = Προβλήματα στη μετάφραση

Το σύστημα και ο χρήστης περιγράφονται μέσω της γλώσσας.

Η **γλώσσα ουσίας** (συστήματος) περιγράφει τα υπολογιστικά χαρακτηριστικά του πεδίου που σχετίζονται με το σύστημα.

Η γλώσσα εργασίας (χρήστη) περιγράφει τα ψυχολογικά χαρακτηριστικά του πεδίου εφαρμογής που σχετίζονται με τον χρήστη.

1. **ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ**
2. Ποιες οι **3 κατηγορίες ευχρηστίας** του ISO 9241 ;
   1. Αποτελεσματικότητα
   2. Απόδοση
   3. Ικανοποίηση
3. **ΚΑΝΟΝΕΣ ΣΧΕΔΙΑΣΗΣ**
4. Ποιοι είναι οι χρυσοί κανόνες του **Shneiderman** ;
   1. Να προσπαθείτε να επιτύχετε την συνέπεια στις αλληλουχίες ενεργειών, τη διάταξη, την ορολογία, την χρήση εντολών, κ.ο.κ.
   2. Να δίνετε στους συχνούς χρήστες τη δυνατότητα να χρησιμοποιούν συντομεύσεις.
   3. Να παρέχετε πληροφοριακή ανάδραση για κάθε ενέργεια του χρήστη, σ’ ένα επίπεδο κατάλληλο για το εύρος της ενέργειας.
   4. Να σχεδιάζετε τους διαλόγους με τρόπο που να προκαλεί το φαινόμενο της περάτωσης.
   5. Να παρέχετε μηχανισμούς πρόληψης σφαλμάτων και απλού χειρισμού έτσι ώστε οι χρήστες να αποφεύγουν τα λάθη ή όταν κάνουν λάθη να παρέχονται σαφείς οδηγίες για την ανάκαμψη.
   6. Να επιτρέπετε την εύκολη αντιστροφή ενεργειών έτσι ώστε να μειώνεται το άγχος των χρηστών και να ενθαρρύνεται η εξερεύνηση.
   7. Να υποστηρίζεται τον «εσωτερικό» έλεγχο, έτσι ώστε ο χρήστης να έχει τον έλεγχο του συστήματος, το οποίο θα ανταποκρίνεται στις ενέργειες του.
   8. Να μειώνετε την επιβάρυνση της βραχυχρόνιας μνήμης κρατώντας απλές τις οθόνες, συγχωνεύοντας τις «πολυσέλιδες» οθόνες και παρέχοντας χρόνο για την εκμάθηση των αλληλουχιών ενεργειών.
5. **ΥΠΟΣΤΗΡΙΞΗ ΥΛΟΠΟΙΗΣΗΣ**
6. Τι γνωρίζετε για το “**Μοντέλο Παρουσίαση Έλεγχος MVC** ” ;
   1. Αρχικά βασίζεται πάνω στην Αντικειμενοστρέφεια.
      1. Μοντέλο → Εφαρμογή
      2. Παρουσίαση → Έξοδος
      3. Έλεγχος → Είσοδος
   2. Ένα μοντέλο μπορεί να συσχετιστεί με περισσότερες από μία τριάδες MVC, δίνοντας ευελιξία στην εφαρμογή (περισσότερους τρόπους υλοποίησης της ίδιας υπηρεσίας).
   3. Στο MVC βασίζεται η JAVA Swing
   4. Το MVC μοντέλο θα μπορούσε να υλοποιηθεί στη Java Swing με τον διαχωρισμό ενός προγράμματος σε τρεις τάξεις :
      1. Model, είναι ανεξάρτητο από τη διεπαφή, δεν γνωρίζει αν η είσοδος και η έξοδος γίνονται σε γραφική διεπαφή ή γραμμή κειμένου.
      2. View, θα περιέχει τον κώδικα υλοποίησης της εμφάνισης (π.χ μέθοδο initComponents() στο Netbeans) καθώς και τον κώδικα που τροποποιεί την εμφάνιση της διεπαφής (π.χ. μέθοδοι setText() κτλ)
      3. Controller, είναι ο συνδετικός κρίκος ανάμεσα στο Model και το View. Ουσιαστικά περιέχει τους ακροατές και τις μεθόδους υλοποίησης τους και καλεί μεθόδους από το Model και το View.
7. Αναλύστε τις 4 **Αρχιτεκτονικές** των **Windows**.
   1. X Windows Αρχιτεκτονική - 1. Κάθε εφαρμογή ή διεργασία που τρέχει σαν πελάτης στον Χ11 διακομιστή συσχετίζεται με ένα ιδεατό τερματικό ή κύριο παράθυρο. Ο Χ διακομιστής εκτελεί τις ακόλουθες εργασίες
      1. Επιτρέπει ή αρνείται την πρόσβαση στην οθόνη από πολλαπλές εφαρμογές πελάτη.
      2. Διερμηνεύει τις αιτήσεις που στέλνουν οι πελάτες για την εκτέλεση λειτουργιών στην οθόνη ή την παροχή άλλων πληροφοριών
      3. Αποπλέκει την ροή των φυσικών συμβάντων εισόδου που λαμβάνει από τον χρήστη και περνάει κάθε συμβάν στην κατάλληλη εφαρμογή πελάτη
      4. Ελαχιστοποιεί την κυκλοφορία στο δίκτυο ανακουφίζοντας τους πελάτες από την υποχρέωση να κρατούν στοιχεία για συγκεκριμένες πληροφορίες παρουσίασης (όπως οι γραμματοσειρές).
   2. X Windows Αρχιτεκτονική – 2. Ένα ξεχωριστό πρόγραμμα πελάτη, ο διαχειριστής παραθύρων, επιβάλλει τις αναγκαίες πολιτικές για την επίλυση αντικρουόμενων αιτήσεων εισόδου/εξόδου από/προς άλλες εφαρμογές πελάτη. Υπάρχουν αρκετοί διαχειριστές παραθύρων που μπορούν να χρησιμοποιηθούν στο περιβάλλον Χ, οι οποίοι υιοθετούν διαφορετικές πολιτικές:
      1. Μέθοδοι επιλογής από τον χρήστη του ενεργού προγράμματος πελάτη στο οποίο θα κατευθύνεται η είσοδος.
         1. ο χρήστης να ορίσει ρητά ένα παράθυρο ως ενεργό
         2. ο χρήστης να ορίσει έμμεσα ένα παράθυρο ως ενεργό με το ποντίκι
      2. Πλευρική παράθεση ή επικάλυψη των παραθύρων
   3. X Windows Αρχιτεκτονική – 3. Οι εφαρμογές πελάτη μπορούν να ορίσουν τη δίκη τους ιεραρχία υποπαραθύρων, τα οποία περιορίζονται στο χώρο του πατρικού παραθύρου. Τα X Windows ολοκληρώνονται με μία σειρά εγχειριδίων όπου καλύπτονται θέματα σχετικά με:
      1. κανόνες για τη μεταφορά δεδομένων μεταξύ εφαρμογών πελάτη
      2. μέθοδοι επιλογής του ενεργού προγράμματος πελάτη στο οποίο θα κατευθύνεται η είσοδος
      3. διατάξεις επικαλυπτόμενων παραθύρων και παραθύρων σε παράθεση.

* **ΤΕΧΝΙΚΕΣ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Ποιοι είναι οι 10 ευρετικοί κανόνες του **Nielsen** ;
   1. Ορατότητα της κατάστασης του συστήματος
   2. Αντιστοιχία μεταξύ του συστήματος και του πραγματικού κόσμου
   3. Έλεγχος και ελευθερία του χρήστη
   4. Συνέπεια και συμμόρφωση του χρήστη
   5. Πρόληψη σφαλμάτων
   6. Αναγνώριση αντί της ενθύμησης
   7. Ευελιξία και αποτελεσματικότητα
   8. Αισθητική και μινιμαλιστική σχεδίαση
   9. Βοήθεια για την αναγνώριση, την διάγνωση και την ανάκαμψη από σφάλματα
   10. Βοήθεια και τεκμηρίωση
2. Ποια η διαφορά «**Μεταξύ υποκειμένων**» και «**Εντός Υποκειμένων**» ;
   1. Μεταξύ υποκειμένων :
      1. Κάθε συμμετέχοντας αντιστοιχίζεται σε μία διαφορετική συνθήκη.
      2. Πλεονεκτήματα :
         1. Το φαινόμενο της μεταφοράς γνώσης (όταν ένας χρήστης αρχικά δουλεύει με μια συνθήκη και μετά με κάποια άλλη) είναι ελεγχόμενο. Κάθε χρήστης δουλεύει μόνο με μία συνθήκη.
      3. Μειονεκτήματα :
         1. Απαιτεί μεγάλο αριθμό συμμετεχόντων
         2. Αν υπάρχουν μεγάλες διαφορές στην εμπειρία των συμμετεχόντων ανά ομάδα μπορεί να ακυρωθούν τα αποτελέσματα.
   2. Εντός Υποκειμένων :
      1. Κάθε συμμετέχοντας εκτελεί τις εργασίες του κάτω από κάθε μία διαφορετική συνθήκη.
      2. Πλεονεκτήματα :
         1. Απαιτεί λιγότερους χρήστες από την μέθοδο μεταξύ-υποκειμένων
         2. Είναι αποτελεσματική ακόμα και όταν χρειάζεται εκμάθηση
         3. Υπάρχει μικρότερη πιθανότητα να επηρεαστούν οι μετρήσεις του πειράματος από τις διαφορές των συμμετεχόντων.
      3. Μειονεκτήματα :
         1. Δεν μελετάται σωστά η μεταφορά γνώσης. Λύνεται το πρόβλημα αυτό αν οι συμμετέχοντες περνάνε από όλες τις συνθήκες με διαφορετική σειρά.
   3. Τα κριτήρια επιλογής της σωστής μεθόδου αποφασίζονται με βάση :
      1. τους διαθέσιμους πόρους
      2. τον έλεγχο της μεταφοράς γνώσης
      3. Πόσο αντιπροσωπευτική είναι η ομάδα των συμμετεχόντων.

* **ΚΑΘΟΛΙΚΗ ΣΧΕΔΙΑΣΗ**

1. Τι είναι η **Καθολική Σχεδίαση** ;
   1. Σαν Καθολική σχεδίαση χαρακτηρίζεται η διαδικασία σχεδίασης προϊόντων με τέτοιο τρόπο, ώστε να μπορούν να χρησιμοποιούνται απ’ όσο το δυνατόν περισσότερους ανθρώπους σε όσο το δυνατόν περισσότερες περιπτώσεις.
   2. Δηλαδή διαδραστικά συστήματα τα οποία μπορούν να χρησιμοποιηθούν:
      1. Από οποιονδήποτε
      2. Με οποιαδήποτε γκάμα ικανοτήτων
      3. Σε οποιαδήποτε τεχνολογική πλατφόρμα.
   3. Αυτό μπορεί να επιτευχθεί εφοδιάζοντας το σύστημα με τις απαραίτητες εφεδρείες ή βοηθητικές τεχνολογίες
2. Ποιες οι **Αρχές** της **Καθολικής Σχεδίασης** ;
   1. Η αρχή της δίκαιης ή ισόνομης χρήσης
   2. Η αρχή της ευελιξίας της χρήσης
   3. Η αρχή του απλού και διαισθητικού συστήματος στη χρήση
   4. Η αρχή της αντιληπτής πληροφορίας
   5. Η αρχή της ανοχής στα σφάλματα
   6. Η αρχή της φυσικής προσπάθειας
   7. Η αρχή της απαίτησης μεγέθους και χώρου για προσέγγιση και χρήση.
3. Πως λειτουργεί η **Πολυτροπική Διάδραση** ;
   1. Αντιγράφοντας τη διάδραση του ανθρώπου με το περιβάλλον του, προσπαθούμε να δημιουργήσουμε πλουσιότερα διαδραστικά συστήματα τα οποία υποστηρίζουν περισσότερες από μία αισθήσεις.
   2. Αυτά τα συστήματα ονομάζονται πολυτροπικά και υποστηρίζουν την αρχή του πλεονασμού, επιτρέποντας στους χρήστες να επιλέγουν τον καταλληλότερο τρόπο διάδρασης ανάλογα με τις ικανότητες τους.

* **ΥΠΟΣΤΗΡΙΞΗ ΤΩΝ ΧΡΗΣΤΩΝ**

1. Ποια τα χαρακτηριστικά του **Ιδανικού Συστήματος Βοήθειας** ;
   1. Το ιδανικό σύστημα βοήθειας θα πρέπει να περιλαμβάνει κάποια από τα παρακάτω χαρακτηριστικά:
      1. Διαθεσιμότητα
         1. Ο χρήστης θα πρέπει να μπορεί να προσπελάζει τη βοήθεια ανά πάσα στιγμή.
         2. Είναι απαράδεκτο να πρέπει να κλείσει μία εφαρμογή για να ανοίξει τη βοήθεια.
         3. Το ιδανικό είναι η βοήθεια και η εφαρμογή να δουλεύουν παράλληλα.
         4. Η βοήθεια είναι πάντα διαθέσιμη σε παραθυρικά συστήματα ενώ στα μη-παραθυρικά συστήματα δεν είναι πάντα διαθέσιμη.
      2. Ακρίβεια και πληρότητα
         1. Η παρεχόμενη βοήθεια θα πρέπει να είναι σύγχρονη κάθε φορά με την τελευταία έκδοση του συστήματος και η βοήθεια που θα παρέχει ακριβής.
         2. Η παρεχόμενη βοήθεια θα πρέπει να καλύπτει κάθε παράμετρο που προγράμματος.
      3. Συνέπεια
         1. Η βοήθεια που παρέχεται από κάθε συστατικό της πρέπει να είναι συνεπής τόσο ως προς το ίδιο το συστατικό, όσο και ως προς τα υπόλοιπα συστατικά βοήθειας.
         2. Η online βοήθεια θα πρέπει να είναι συνεπής με το έντυπο υλικό τεκμηρίωσης.
         3. Θα πρέπει να υπάρχει συνέπεια της βοήθειας ως προς :
            1. Το περιεχόμενο της
            2. Την ορολογία της
            3. Το στυλ παρουσίασης της
         4. Θα πρέπει να υπάρχει συνέπεια της βοήθειας ανάμεσα στις εφαρμογές.
      4. Ευρωστία
         1. Είναι σημαντικό το σύστημα βοήθειας να επιδεικνύει αυξημένη ευρωστία γιατί συνήθως χρησιμοποιείται όταν αντιμετωπίζεται κάποια δυσκολία.
         2. Το σύστημα βοήθειας για να έχει ευρωστία θα πρέπει :
            1. Να διαχειρίζεται σωστά τα σφάλματα του χρήστη όταν χρησιμοποιεί την εφαρμογή,
            2. Να έχει προβλέψιμη συμπεριφορά,
            3. Να μπορεί να βασιστεί ο χρήστης στο σύστημα βοήθειας κάθε φορά που το έχει ανάγκη.
      5. Ευελιξία
         1. Ένα ευέλικτο σύστημα βοήθειας θα πρέπει να δίνει σε κάθε χρήστη την δυνατότητα να αλληλεπιδρά μαζί του με έναν τρόπο κατάλληλο για τις ανάγκες του.
         2. Ευέλικτα συστήματα βοήθειας είναι τα προσαρμόσιμα συστήματα βοήθειας, τα οποία συμπεραίνουν μόνα τους το επίπεδο της εμπειρίας του χρήστη και την εργασία που εκτελεί.
         3. Η χρήση του υπερκειμένου οδηγεί σε ευέλικτα συστήματα βοήθειας γιατί δίνει τη δυνατότητα στον χρήστη να μετακινείται μέσα στο περιεχόμενο της βοήθειας ανάλογα με τις ανάγκες του.
      6. Διακριτικότητα
         1. Το σύστημα βοήθειας δεν θα πρέπει να εμποδίζει τον χρήστη να συνεχίσει την εργασία του.
            1. Στις μη παραθυρικές διεπιφάνειες
         2. Το σύστημα βοήθειας δεν θα πρέπει να παρεμβάλλεται στην εργασία του χρήστη.
            1. Δεν είναι διακριτικό όταν ένα προσαρμόσιμο σύστημα βοήθειας το οποίο παρέχει βοήθεια με δική του πρωτοβουλία.