Permbajtja

1. Prezantimi
2. Hyrje në rrjetat kompijuterike
3. Qëllimi
   1. Vështirësitë që studentët kanë mbi rrjetat kompijuterike
   2. Zgjidhjet e mundshme
4. Zgjidhja me një aplikacion mobile
   1. Zgjidhja nëpërmjet një aplikacioni android
   2. Zgjedhja e platformës android
5. Zhvillimi aplikacionit
   1. Përdorimi i gjuhes “Processing”
   2. Të dhëna mbi programin
   3. Praktikat e mira të programimit
6. Zhvillime të mëtejshme të aplikacionit

RRJETAT KOMPJUTERIKE

Hyrje ne rrjetat kompjuterike

Në të kaluarën deri 20 apo 30 vjet më parë përdorej metoda me një kompijuter të vetëm qe i sherbente te gjithe nevojave kompjuterike te organizates.Sot ky eshte zevendesuar me nje model tjetër ne te cilin nje numer i madh por te nderlidhur kompjutrash qe e bejne kete pune.

Keto sisteme quhen rrjete kompjuterike, që në vetëvete janë koleksion i kompjutrave autonom te nderlidhur me nje teknologji te vetme.Dy kompjutra thuhet se jane te nderlidhur kur ata jane jane ne gjendje te shkembejne informacion. As interneti dhe as www. nuk jane rrjet kompjutrash. Interneti eshte rrjet rrjetash ndersa www eshte nje sistem shperndares, i cili eshte nje koleksion i kompjutrave indipendent qe i shfaqen perdoruesit si nje sistem i vetem koherent, ka nje model te vetin i cili implementohen nga nje shtrese softweri ne maje te sistemit operativ e quajtur middleware. Ngjashmeria qendron ne ate qe si rrjeti kompjuterik ashtu edhe sistemi shperndares sherbejne per te levizur files ndryshimi qendron ne ate qe kush e provokon sistemi apo perdoruesi.

LLOJET E RRJETAVE: (SI KLASIFIKOHEN RRJETAT?)  
PAN-(Personal Area Network) Perdoret ne distancat deri ne 1m largesi  
LAN- (Local Area Network) Kur distanca varion nga 10m deri ne 1km  
MAN-(Metropolitan Area Network) --------!!!!!  
WAN-(Wide Area Network) Distanca 1000km deri 10000km  
INTERNET-(Rrjet Global) Distanca mbi 10000km

RRJETAT PEER TO PEER   
Ne keto lloj rrjetash kompjuterat komunikojne drejtperdrejt me njeri tjetrin nepermjet nje kablli te vetem. Kompjuteret ndajne te dhenat dhe pajisjet ndermjet tyre. Kompjterat kane autoritet te barabarte dhe perdoren kompjutere per perdorim te thjeshte personal.  
AVANTAZHET E PEER TO PEER (CILET JANE AVANTAZHET E RRJETAVE PEER TO PEER?)  
-Rrjetat peer to peer jane te thjeshte per tu konfiguruar.  
-Perdoren ne mjedise ku nuk kerkohet shume profesionalizem.  
-Kerkojne pak shpenzime per tu mirembajtur.  
-Perdoren ne mjedise kritike ne lidhje me koston.  
DISAVANTAZHET E RRJETAVE PEER TO PEER  
-Nuk jane shume fleksibel  
-Nuk jane shume te shkallezueshem  
-Nuk jane shume praktik pasi ne lidhjen e shume kompjuterave nuk I centralizojne dot burimet.

RRJETAT KLIENT SERVER  
Keto rrjeta perdoren kompjutere special SERVER te cilet procesojne te dhenat dhe lehtesojne komunikimin. Ndersa KLIENTET jane kompjutera me perdorim te pergjithshem qe quhen ndryshe WORKSTATION.  
Detyra baze e SERVERIT eshte qe te realizoje ndarjen e burimeve ndermjet klienteve.  
Ne modelin KLIENT SERVER zhvillohen dy procese: Procesi KLIENT dhe Procesi SERVER.  
(SI REALIZOHET PROCESI KLIENT-SERVER?)  
Procesi KLIENT I dergon mesazh SERVERIT nepermjet rrjetit dhe vete kalon ne gjendje pritje.  
Kur procesi server merr mesazhin kryen punen e kerkuar ose kerkon te dhena per klientin, ben mesazhin dhe I kthen pergjigje klientit. Klientet mund ti egzekutojne aplikimet dhe ruajne te dhenat ne HDD HEADERIN e tyre. Serveri nga ana tjeter ofron opsionin e perdorimit te aplikimeve dhe te te dhenave te perbashketa. Klientet nuk komunikojne drejtperdrejt permes ketyre por perdorin serverin si ndermjetes.  
SERVERAT jane kompjutera shume te fuqishem te pajisur me HARDWARE special te projektuar per te siguruar funksionin e menaxhimit te rrjetit gje e cila nuk realizohet nga KLIENTET.  
AVANTAZHET E KLIENT SERVER MBI PEER TO PEER  
Avantazhet:  
-Perdorimi I paswordeve ne rrjetat klient server  
-Ka aksesim te burimeve te perbashketa ne nje system te centralizuar  
-Serverat jane te optimizuar te perballojne ngarkesa te medha te procesimit te te dhenave dhe trajtojne kerkesa te dedikuara nga klienti.  
-Serverat mund te lidhin shume kompjutera ne rrjet sepse ato kan aftesi te medha procesimi te te dhenave.   
ELEMENTET BAZE TE RRJETAVE  
(SHPJEGONI FUNKSIONET E ELEMENTEVE BAZET TE RRJETAVE?)  
KLIENTI- Eshte kompjuteri qe kerkon burime apo sherbime nga nje kompjuter tjeter ne rrjet.  
SERVERI- Eshte kompjuteri qe menaxhon burime te perbashketa.  
WORKSTATION- Eshte kompjuteri I lidhur ose jane rrjet.  
NOS- (Network Operating System) Sistemi operativ per rrjetin. Eshte software qe ekzekutohet ne nje server dhe e ndihmon serverin te menaxhoje te dhenat, te menaxhoje perdoruesin, grupet, sigurine, aplikimet, dhe funksione te tjera qe ofron rrjeti.  
HOSTI- Eshte Server qe menaxhon burimet e perbashketa.  
NYJA- Mund te jete Klient ose Server ose pajisje tjeter ne rrjet qe komunikon dhe identifikohet nga nje numer unik qe quhet adrese rrjeti.  
TOPOLOGJIA- Ka te beje me shperndarjen fizike te kompjuterave ne rrjet.  
PROTOKOLLI- Jane rregulla qe I vendos rrjeti per transferimin e te dhenave.  
PAKETAT E TE DHENAVE- Jane njesi te vecanta te dhenash qe transferohen nga nje kompjuter te tjeteri.  
ADRESIMI- Skema qe I vendoset nje Workstationi apo nje pajisjeje tjeter ne rrjet dhe quhet numer unik identifikues.  
PAJISJET E TRANSMETIMIT- Jane mjete neper te cilat transmetohen dhe merren te dhenat.

STANDARTET E RRJETAVE DHE MODELI OSI  
OSI-(Open System Interconnection)  
STANDARTET jane marreveshje te dokumentuara qe mbajne specifikime teknike dhe kritere te tjera specifike qe percaktojne se si nje product ose nje sherbim I vecante duhet te realizohet ose te projektohet.   
Standartet vendosin marredheniet ndermjet arkitekturave dhe platformave te ndryshme.  
Disa nga organizatat e standarteve jane:  
-ANSI- American Nation Standards Institute  
-IEEE- Institute of Electrical Enleterning Engineers  
-ISO- International Organization for Standardisation  
-ITU- International Telecamunication Union

MODELI OSI  
Eshte nje prezantim teorik I asaj qe ndodh ndermjet dy nyjeve ne nje rrjet.  
OSI e ndan arkitekturen e rrjetit ne shtate shtresa:  
-Shtresa e aplikimit (application)  
-Shtresa e prezantimit (presentation)  
-Shtresa e seksionit (section)  
-Shtresa e transportit (transport)  
-Shtresa e rrjetit   
-Shtresa e lidhjes se te dhenave (data link)  
-Shtresa fizike

Ky model nuk percakton llojin Hardware opo Software qe duhet te suportoje cdo shtrese.  
Nuk percakton se si programet aplikative nderveprojne me njeri tjetrin apo si keto programe nderveprojne me njerezit.  
FUNKSIONI I SHTRESES  
Shtresa fizike eshte shtresa me e ulet e modelit OSI kjo shtrese mban pajisjet fizike si KABLLIMI lidhesit dhe perseritesit.  
Protokollet ne shtresen fizike gjenerojne tension per te transmetuar dhe per te marre te dhena apo sinjale qe mbartin te dhena. Kjo shtrese percakton normen e transmetimit te te dhenave dhe monitoron normat e gabimeve te te dhenave. Problemet fizike ne rrjet ndikojne drejtperdrejt ne shtresen fizike, pajisjet qe I gjejme ne shtresen fizike perfshijne PERSERITESIT dhe HUBET.  
Karta NIC e nderfaqes me rrjetin veprojne si ne shtresen fizike ashtu edhe ne shtresen e lidhjes se te dhenave.   
- Shtresa e lidhjes se te dhenave   
Eshte shtresa e dyte e modelit OSI e cila kontrollon komunikimet ndermjet shtreses se rrjetit dhe asaj fizike. Funksioni primar I kesaj shtrese eshte qe ti ndaje te dhenat qe merr nga shtresa e rrjetit ne korniza me te vogla te dhenash te tilla qe te mund te transmetohen nga shtresa fizike.  
KORNIZA E TE DHENAVE  
Eshte nje paket e strukturuar e cila sherben per te zhvendosur te dhenat dhe qe nuk permban vetem te dhenat por edhe adresat e rrjetit te derguesit dhe adresat e rrjetit te marresit.  
Adresat I tregojne rrjetit ku ta dergoje kornizen nderkohe kontrolli I gabimeve dhe informacioni I kontrollit qe ndodhen mbi shtresen e lidhjes se te dhenave sigurojne qe informacioni te arrije ne destinacion pa problem.  
SHTRESA DATALINK-(Shtresa e Lidhjeve) pret pranimin e te dhenave nga marresi qe te dhenat u marrin [strukturen mberiten]-----  
Nese derguesi nuk e merr kete mesazh pranimi shtresa e tij jep udhezime per ritransmetim te te dhenave. Kjo shtrese kontrollon rrjedhjen e informacionit duke lejuar qe karta NIX ti perpunoje te dhenat pa gabime, kjo shtres funksionon ne menyre te pavarur.  
SHTRESA E RRJETIT  
Eshte shtresa e trete e OSI e cila perkthen adresat e rrjetit ne adresa fizike. Kjo shtrese vendos per itinerarin drejtimin me te mire te mundshem te transmetimit te te dhenave nga derguesi tek marresi. Duke marre parasysh: 1-prioritetet e dergimit 2-Perplasjet ne rrjet 3-Cilesine e sherbimit 4-Koston e rrugeve qe duhen zgjedhur.  
Protokollet e shtreses se rrjetit realizojne gjithashtu segmentimin e te dhenave e cila nenkuptohet si nje operim te dhenash nga paketa te medha ne paketa me te vogla, por gjithashtu kryen edhe procesin e ribashkimit qe eshte procesi I kundert I segmentimit, gje e cil;a ka te beje me procesin e rindertitmit e paketave te medha. Ne destinacion formohet paketa fillestare si pasoj e procesit te ribashkimit.  
SHTRESA E TRANSPORTIT  
Siguron qe te dhenat te transmetohen ne menyre te sigurte ne te njejten radhe dhe pa gabime.  
Protokollet ne shtresen e transportit realizojne kontroll te rrjedhshem te transportit, pra percakton normen e shpejtesise ne te cilen marresi mund te pranoje te dhenat.  
Ne shtresen e rrjetit te dhenat copetohen ne ate mase qe te mund te suportohen nga shtresa e transportit. Kur keto paketa te dhenash kalojne ne shtresen e transportit ato pajisen me nje numer sekuencash ne menyre te tille qe kur te mberrijne ne destinacion te bashkohen ne radhen e duhur. Ne qofte se te dhenat arrijne te sakta ne destinacion shtresa e transportit e nyjes marrese dergon nje mesazh pranimi duke I treguar derguesit qe te dhenat u morren pa gabime, nese te dhenat permbajne gabime shtresa e transportit kerkon derguesit qe ti ridergoje te dhenat. Edhe kur te dhenat nuk pranohen per nje kohe te gjate shtresa e transportit e derguesit I konsideron ato te humbura dhe I ridergon.

Shtresa e seksionit  
Realizon mirembajtjen dhe komunikon ndermjet dy nyjeve ne nje rrjet. Fjala seksion I referohet nje lidhjeje te shkembimit te te dhenave ndermjet dy aneve. Funksionet baze te shtreses se seksionit jane:  
Vendosja dhe mirembajtja e nje lidhje komunikimi;   
Sinkronizimi I dialogut ne nje lidhje;  
Kur lidhja ndermjet dy nyjeve shkeputet duhet te percaktohet se kur te rifilloje transmetimi I te dhenave, monitorimi dhe identifikimi ---------------- te seksionit duke bere qe vetem nyjet e autorizuara te mund ti aksesojne seksionin.  
Shtresa e prezantimit  
Sherben si perkthyes ndermjet aplikimit dhe rrjetit. Ne kete shtrese te dhenat formatohen ne ate menyre qe rrjeti ta kuptoje. Ne kete shtrese te dhenat enkriptohen dhe dekriptohen duke iu pajisur dhe me fjalekalime. Protokollet e kesaj shtrese kontrollojne dhe dekodojne informacionet mbi grafiket dhe format e fileve.  
Shtresa e aplikimit  
Siguron nderfaqe me software qe I mundesojne programeve perdorimin e sherbimeve te rrjetit. Nje nderfaqe e programit aplikativ lejon nje program te nderveproje me sistemin operativ. Keto nderfaqe I perkasin pikerisht kesaj shtrese te modelit OSI. Programuesit I perdorin keto nderfaqe per te krijuar nje marredhenie ndermjet kodit te tyre dhe sistemit operativ.

ALOHA

Aloha është një sistem për koordinimin dhe arbitrimin në një kanal të përbashkët në rrjetet e komunikimit. Ajo është zhvilluar në 1970 nga Norman Abramson dhe kolegët e tij në Universitetin e Hawaii. Sistemi origjinal përdorej për transmetime radio, por sistemi është zbatuar në sistemet e komunikimit satelitor.

Një sistem i përbashkët i komunikimit si Aloha kërkon një metodë të trajtimit të përplasjeve që ndodhin kur dy ose më shumë sisteme kërkojnë për të transmetuar në kanal në të njëjtën kohë. Në sistemin aloha, një nyje transmeton kur ka të dhëna në dispozicion për të dërguar.Nëse një nyje tjetër transmeton në të njëjtën kohë, një përplasje ndodh, dhe paketat që janë të transmetueshme humbasin. Megjithatë, një nyje mund të dëgjojë transmetimet, madje edhe të vetat, dhe të përcaktojë nëse paketat janë transmetuar.

**Aloha do të thotë "Hello".** Aloha është ‘’multiple access [protocol](http://ecomputernotes.com/computernetworkingnotes/computer-network/protocol)’’ në shtresën DATALINK dhe propozon si terminale të shumta të hapur të aksesojnë pa ndërhyrje apo përplasje. Në vitin 1972 Robert zhvilloi një protokoll që do të rrisë kapacitetin e aloha dy herë. Protokolli Slotted Aloha përfshin ndarjen e intervali kohor në pjesë diskrete dhe çdo interval “slot” korrespondon me periudhën kohore për një paketë apo grup paketash. Kjo metodë kërkon sinkronizimin midis nyjeve dërguar për të parandaluar përplasjet.

Ka dy versione ndryshme të aloha:

(i) Aloha  të pastër   
(ii) Slotted aloha

**(I) ALOHA e pastër**

**• Në** Aloha pastër, stacionet transmetojnë korniza sa herë që ata kanë të dhëna për të dërguar.

• kur dy ose më shumë stacione transmetojë njëherësh, ka goditje dhe “framet” shkatërrohen.

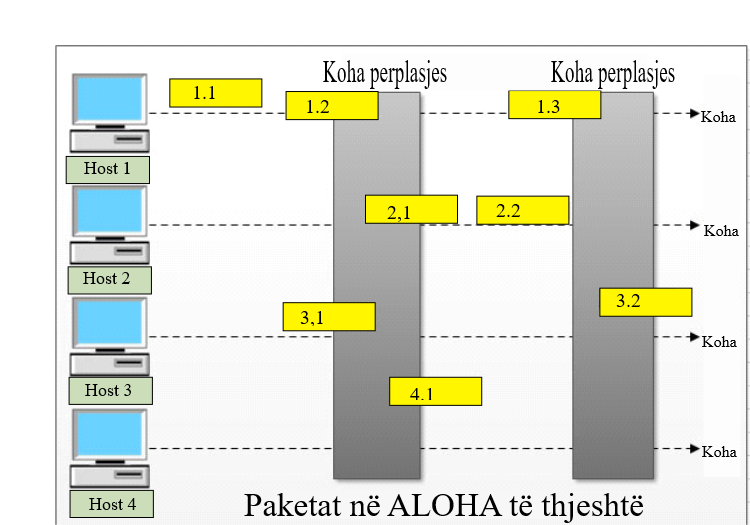
• Në Aloha pastër, sa herë që ndonjë stacion transmeton një “frame”, ajo pret njohjen nga pranuesit.

• Në qoftë se pranimi nuk është marrë brenda afatit të caktuar, stacioni supozon se “fram-a” (ose pranimi) është shkatërruar.

• Nëse “frame” është shkatërruar për shkak të përplasjes stacioni pret për një sasi të rastit të kohës dhe ia dërgon atë përsëri. Këtë herë afati pritjes duhet të jetë i rastësishëm, ndryshe “frame” e njëjta do të përplaset përsëri dhe përsëri.

• Prandaj Aloha e pastër dikton se kur afati kalon, çdo stacion duhet të presë për një çast të rastit para se ridërgojë “frame” e saj. Kjo kohë e rastësishme do të ndihmojë për të shmangur goditjet me shume.

• Figura tregon një shembull të përplasjeve kornizë në Aloha pastër.



• Në fig ka katër stacione që kanë ndarë me njëri-tjetrin për përdorim në kanal të përbashkët. Të gjitha këto stacione janë duke transmetuar “frame”. Disa prej këtyre “frame” përplasen për shkak se “frame” të shumta janë në garë për kanalin përbashkët. Vetëm dy korniza, “frame” 1.1 dhe 2.2 kornizë arrijnë të mbijetojnë. Të gjitha “frame”e tjera janë shkatërruar.

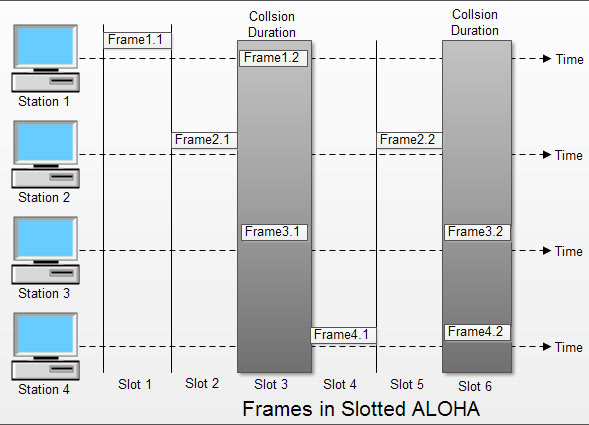
• Sa herë që dy “frame” do të përpiqen për të zënë kanalin në të njëjtën kohë, do të ketë një përplasje dhe të dy do të jetë të dëmtuara. Nëse bit i parë i një “frame” të re përputhet me vetëm pak nga fundit i një “frame” tjetër pothuajse të përfunduar, të dy “frame” do të shkatërrohen krejtësisht dhe të dy do të duhet të ritransmetohen.

**(Ii) Slotted aloha**

• Slotted Aloha u shpik për të përmirësuar efikasitetin e Alohas së pastër sepse shanset e përplasjes në Aloha pastër janë shumë të larta.

• Në slotted aloha, koha e kanalit të përbashkët është i ndarë në intervale diskrete të quajtura slots.

• Stacionet mund të dërgojë një “frame” vetëm në fillim të slotit dhe vetëm një “frame” dërgohet në çdo slot.



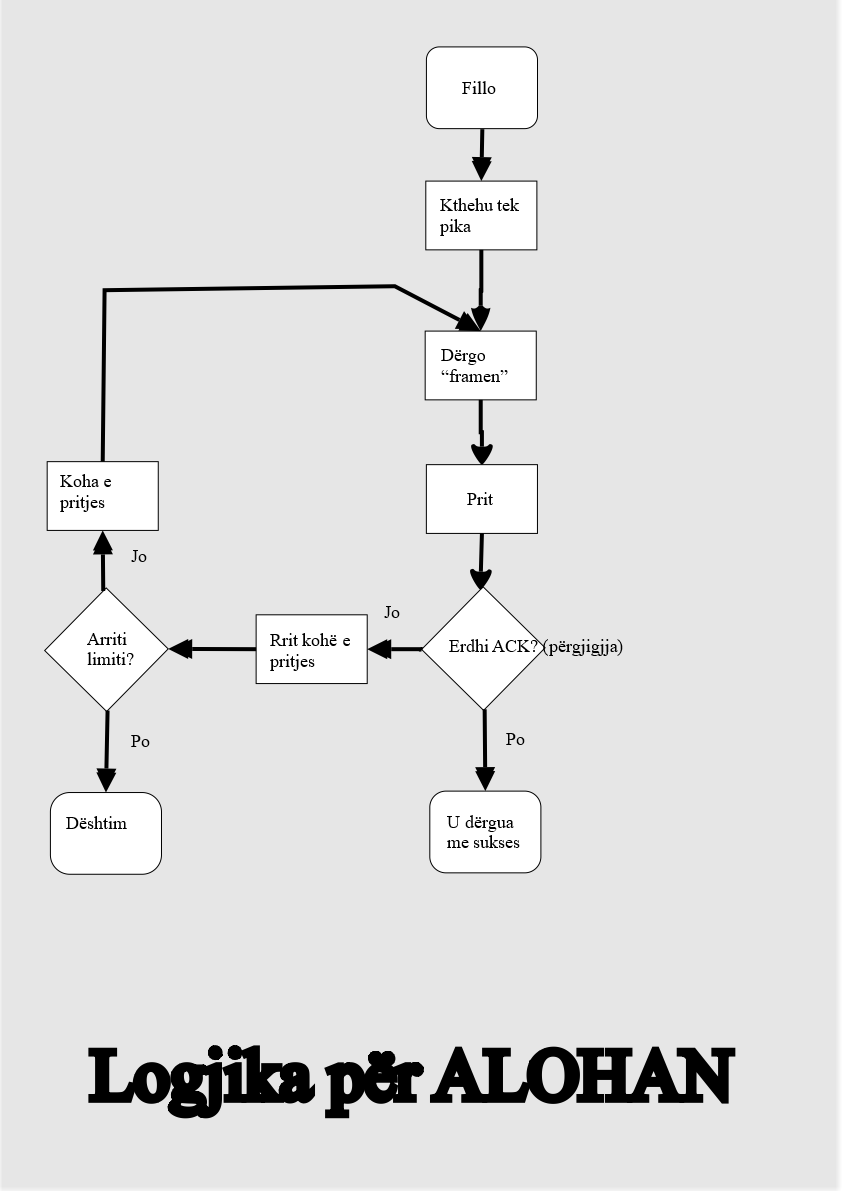
Në slotted aloha, nëse stacioni nuk është në gjendje për të vendosur “frame” mbi kanalin në fillim të slotit  *Psh*  nëse humbet kohën e slotit të vet atëherë stacioni duhet të presë deri në fillim të slotit tjetër.

• Në slotted aloha, ka ende një mundësi e përplasjes në qoftë se dy stacione përpiqen për të dërguar në fillim të të njëjtit segment kohor siç është paraqitur në figurë.

• Slotted Aloha ende ka një avantazh mbi Aloha pastër pasi shanset e përplasjes janë reduktuar në një gjysmë edhe pse transmetimi eshtë më i kufizuar.

## **Protokolli Flow Chart për aloha:**

Figura tregon grafikun e rrjedhës protokoll për Aloha.



**Shpjegim:**

• Një stacion i cili ka një “frame” të gatshme do të dërgojë atë.

• Pastaj ai pret për disa kohë.

• Nëse ai merr njofimin mbapsht, atëherë transmetimi është i suksesshëm.

• Përndryshe stacioni përdor një strategji pritjeje, dhe e dërgon paketën përsëri.

• Pas shumë herëve nëse nuk ka njoftim, atëherë stacioni anullon idenë e transmetimit.

Burim reference: http://ecomputernotes.com/computernetworkingnotes/communication-networks/what-is-aloha

Distance Vector Routing (DVR)

Protokollet eDistance Vector Routing bazojnë vendimet e tyre në rrugën më të mirë për një destinacion të caktuar bazuar në distancë. Distanca matet zakonisht në HOPS, edhe pse distanca metrikë mund të jetë vonesa, pako humbur, apo diçka të ngjashme. Nëse distanca është metrikë hop, atëherë çdo herë që një pako shkon përmes një router, një hop konsiderohet të ketë përshkuar. Rruga me numrin më të vogël të HOPS në një rrjet të caktuar është konstatuar të jetë rruga më e mirë drejt këtij rrjeti.

Vektori tregon drejtimin në atë rrjet të veçantë. Protokollet Distance vektoriale dërgojnë gjithë tabelën e rutimit për fqinjët e lidhur direkt. Shembuj të protokolleve vektor distancë përfshijnë **RIP - Routing Information Protocol** dhe **IGRP - Brendshme Gateway Routing Protocol**.

Nëse jeni të interesuar në gjetjen e më shumë informacion mbi RIP, shikoni shkrimet e mia se si të konfigurosh Routing Information Protocol[RIPv1](https://www.pluralsight.com/blog/cisco-how-to-configure-rip-1) dhe [RIPv2](https://www.pluralsight.com/blog/cisco-how-to-configure-rip-2) .

# Address Resolution Protocol (ARP)

Përkufizimi - Çfarë Address Resolution Protocol (ARP) do të thotë.

Address Resolution Protocol (ARP) është një protokoll rrjeti i nivelit të ulët për përkthimin e adresave nga shtresa e rrjetit në adresat shtresën e link-ut.   
  
ARP shtrihet në mes të shtresave 2 dhe 3 të modelit OSI, edhe pse ARP nuk është përfshirë në kuadrin OSI dhe lejon kompjutera për të futur njëri -tjetrin nëpër një rrjeti përpara komunikimit.   
  
Për shkak se protokollet janë njësi bazë të komunikimit të rrjetit, adresa është e varur në protokollet si ARP, e cila është e vetmja metodë e besueshme e trajtimit detyrat e kërkuara.

Kur konfigurohet një kompjuter të ri rrjeti, çdo sistem është caktuar mbi një adresë Internet Protocol (IP) për identifikimin dhe komunikim primar. Një kompjuter gjithashtu ka një identitet unik me anë të adreses së media access control (MAC).Prodhuesit ja kanë ngjitur adresën MAC në kartën lokale Area Network (LAN). Adresa MAC është e njohur edhe si adresën fizike e kompjuterit.   
  
Para se dy kompjutera të komunikojnë, secili duhet të di IP e tjetrit afërm apo MAC adresat. Nëse kompjuteri A ka vetëm MAC adresën e kompjuterit B, kompjuteri A mund të zbulojë IP adresën e saj duke dërguar një kërkesë ARP për kompjuter B. Kompjuteri B pastaj mund të përgjigjti duke bashkangjitur adresën e tij IP me ARP për kompjuterin A. Kjo adresë përkthimi dhe ky proces shkëmbimi i thjeshtë është roli kryesor i ARP.   
  
Tabelat ARP mund të ruhen për të rritur normat e transmetimit nga mbajtja e adresave të njohur në rrjetin dhe transmetimin e ndonjë MAC ose IP që ndrzshon nga ARP.   
  
Nuk ka asnjë autentifikim nevojshme në këtë nivel, kështu që levizja e IP dhe MAC Adresat është e mundur. Ndonjë software shtesë mund të jetë i nevojshm të drejtoje tabelat e ARP dhe parandalimin e sulmeve dashakeqe të disa përdoruesve.

Qëllimi i këtij projekti

Zgjidhja nëpërmjet një aplikacioni android

Zgjedhja e platformës android

Zhvillimi aplikacionit

Përdorimi i gjuhes “Processing”

Të dhëna mbi programin

Praktikat e mira të programimit

Praktikat e mira te kodimit

Hyrje

Teknikat e mira të kodimit dhe praktikat e programimit janë shenjat dalluese të një programues profesionist. Pjesa më e madhe e programit konsiston në marrjen e një numri të madh të zgjedhjesh të vogla, kështu përpiqesh për të zgjidhur një sërë më të madhe të problemeve. Sa mençurisht këto zgjedhje janë bërë varet në masë të madhe me aftësitë dhe ekspertizën e programuesit.

Ky kapitull trajton disa teknika themelore kodimit dhe ofron një koleksion të praktikave të kodimit nga e cila kemi disa gjëra të cilat janë përdorur edhe në aplikacionin tim. Teknikat e kodimit janë kryesisht ato që përmirësojnë leximin dhe mirëmbajtjen e kodit, ndërsa praktikat e programimit janë kryesisht rritje të performancës.

Lexueshmëri e kodit burim ka një ndikim të drejtpërdrejtë në se sa mirë një zhvillues e kupton një sistem kompjuterik. Mirëmbajtja e kodit i referohet sa lehtë sistemi software mund të ndryshohet për të shtuar veçori të reja, të ndryshuar tiparet ekzistuese, për të rregulluar problemet, ose të përmirësuar punën. Edhe pse lexueshmëri dhe mirëmbajtje janë rezultat i shumë faktorëve, një aspekt të veçantë të zhvillimit të softuerit në të cilin mbi të gjithë zhvilluesve ka ndikim është teknika e kodimit. Metoda më e lehtë për të siguruar që një ekip i zhvilluesve do të japin kodin e cilësisë së duhur është të krijohet një standard kodimi, i cili është i detyrueshëm pastaj në komentet rutinë të kodit.

Standardet e kodimit dhe shqyrtimet e kodit

Një standard i plotë kodimi përfshin të gjitha aspektet e ndërtimit të kodit dhe, ndërsa nga zhvilluesit duhet të ushtrojnë kujdes në zbatimin e tij, ai duhet të ndiqet nga afër. Kodi burimor i plotësuar duhet të pasqyrojnë një stil të harmonizuar, edhe në qoftë se një zhvillues i vetëm ka shkruar kodin në një seancë. Në fillim të një projekti software, të krijojë një standard kodim për të siguruar që të gjithë zhvilluesit për këtë projekt janë duke punuar në bashkëpunim. Kur projekti software do të përfshijnë kod burimi ekzistues, ose kur kryejnë mirëmbajtje mbi një sistem ekzistues software, standardi kodimit duhet të deklarojë si të merren bazat ekzistuese të kodit.

Megjithëse qëllimi kryesor për kryerjen e rishikimeve në kod përgjatë ciklit jetësor të zhvillimit është të identifikojë defektet mbi kod, komentet mund të përdoret gjithashtu për të zbatuar standardet e kodimit në mënyrë uniforme. Aderimi në një standard kodimi mund të jetë i mundshëm vetëm kur ndiqen gjatë gjithë projektit software nga fillimi në përfundimin. Kjo nuk është praktike, as nuk është e kujdesshme, për të vendosur një standard kodues pas faktit të kryer.

Teknikat e kodimit

Teknikat e kodimit përfshijnë shumë aspekte të zhvillimit të softuerit dhe, edhe pse ata zakonisht nuk kanë asnjë ndikim në funksionimin e aplikimit, ato kontribuojnë në një të kuptuar të mirë të kodit burim. Për qëllimet e këtij dokumenti, të gjitha format e kodit burimor janë konsideruar, duke përfshirë programimin, scripting, “Markup languages”, dhe gjuhë “query” ose anketash nëse do t’u shprehnim në shqip.

Teknikat e kodimit të përcaktuara këtu nuk janë propozuar për të formuar një grup të papërkulur të standardeve të kodimit. Përkundrazi, ata kanë për qëllim të shërbejë si një udhërrëfyes për zhvillimin e një standard kodim për një projekt të veçantë software.

Teknikat coding mund t’i ndajmë në tre seksione:

* [Emrat](https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx#cfr_names)
* [Koment](https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx#cfr_comments)et
* [Format](https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx#cfr_format)i

Emrat

Ndoshta një nga ndihmat më me ndikim për të kuptuar rrjedhën logjike të një kërkese është se si quhen elementet e ndryshme të aplikimit. Një emër duhet të tregojë se "çfarë" dhe jo "si." Duke shmangur emrat që ekspozojnë zbatimin themelor, i cili mund të ndryshojë, të ruajnë një shtresë e abstraksionit që thjeshton kompleksitetin. Për shembull, ju mund të përdorni GetNextRouter()(*marrë nga aplikacioni “Bazat e network-ut”*) në vend të GetNextArrayElement().

Një parim i emërtimit është se ka vështirësi në zgjedhjen e një emri të duhur mund të tregojnë se ju keni nevojë për të analizuar më tej ose të përcaktojë qëllimin e një artikulli. Bëni emrat kohë të mjaftueshme për të qenë kuptimplotë, por të shkurtër të mjaftueshme për të shmangur duke u gojor. Programuar, një emër unik shërben vetëm për të dallojnë një objekt nga një tjetër. emrat ekspresive të funksionojë si një ndihmë për lexuesit e njeriut; për këtë arsye, kjo ka kuptim për të siguruar një emër që lexuesi i njeriut mund ta kuptojë. Megjithatë, të jenë të sigurt se emrat e zgjedhura janë në përputhje me rregullat dhe standardet e gjuhës të zbatueshme s.

Në vijim janë rekomanduar emërtimin teknika:

Rutinat

Shmangni emrat e pakapshëm që janë të hapur për interpretim subjektiv, të tilla si Analyze(), ose xxK8 për një ndryshore. Emra të tillë të kontribuojë në dykuptimësi më shumë se abstraksionit.

Në gjuhë objekt-oriented, ai është i tepërt për të përfshirë emra të klasës në emër të pronave të klasës, të tilla si tab.tabFindRow. Në vend të kësaj, përdorni tab.findRow ()(stepNum *marrë nga aplikacioni “Bazat e network-ut”*)..

Përdorni metodën folje-emër për emërtimin rutinat që kryejnë disa operacion në një objekt të caktuar, të tilla si CalculateInvoiceTotal().

Në gjuhë që lejojnë funksion mbingarkesat, të gjitha mbingarkesa duhet të kryejë një funksion të ngjashëm. Për ato gjuhë që nuk lejojnë funksion mbingarkesat, të krijojë një standard emërtimin që lidhet funksione të ngjashme.

Variablat

Është mirë të vendoset një mbaresë dalluese (si mesatare, Sum, Min, Max, Index) në fund të një emri variable kur duket e përshtatshme.

Është mirë të përdoren palët e kundërta në emrat e variablave, si min / max, fillim / fund, dhe hap / mbyll.

Duhe qënë se shumica e emrave janë ndërtuar nga bashkimi i disa fjalëve së bashku, përdorni format të përzier të shkronjave të mëdha dhe të vogla për të lehtësuar leximin e tyre.Përveç kësaj, për të ndihmuar dallimin midis variablave, përdorni modelin Pascal ( ActivateScene() *marrë nga aplikacioni “Bazat e network-ut”*) për emrat e zakonshëm ku shkronjën e parë të çdo fjalë është kapitalizuar. Emrat e variablave, përdorni deve zorrë (stepNum *marrë nga aplikacioni “Bazat e network-ut”*) ku shkronja e parë të çdo fjalë përveç të parës është e kapitalizuar.

Variablat buleane duhet të përmbajë “Is” që nënkupton “PO/JO” ose “True/False”. Psh vlerat të tilla si “IsActive()” (*marrë nga aplikacioni “Bazat e network-ut”*)

Duhet shmangur përdorimi i termave të tilla si Flag/Shenjë/statusi kur kemi emërtimin e variablave të gjendjes, të cilat ndryshojnë nga variablat Boolean se ata mund të kenë më shumë se dy vlera të mundshme. Në vend të documentStatus, përdorin një emër më përshkrues të tilla si documentFormatType.

Edhe për një ndryshore jetëshkurtër që mund të shfaqen në vetëm disa rreshta të kodit, sërisht është mirë të përdoret një emër kuptimplotë. Përdoren emra me një shkronjë të variablave “i” apo ”j”, vetëm në raste të tilla ciklet e shkurtra for.

Nëse përdoret Konvencioni i emërtimit hungarez i Charles Simonyi, ose ndonjë derivat të tij, të zhvillohet një listë e prefikseve standarde për projektin për të ndihmuar zhvilluesit vazhdimisht me emërtimin e variablave.

Për emrat e variablave, ndonjëherë është e dobishme për të përfshirë ndonjë simbol që tregon qëllimin e variablit, të tilla si prefixing një g\_për variablat globale.

Konstantet duhet të jetë mbi të gjitha me shkronja të mëdha me vijë poshtë mes fjalëve, të tilla si NUM\_DAYS\_IN\_WEEK. Gjithashtu, fillojnë grupet e enumeracioneve me një prefiks të përbashkët, si FONT\_ARIAL dhe FONT\_ROMAN.

Të tjera

Duhet të kemi minimizuar përdorimin e shkurtesave. Në qoftë se shkurtimet janë përdorur, të jenë në përputhje me përdorimin e tyre. Një shkurtim duhet të ketë vetëm një kuptim dhe po ashtu, çdo fjalë shkurtuar duhet të ketë vetëm një shkurtim. Për shembull, në qoftë se duke përdorur *min* për të shkurtuar *minimum* , ta bëjmë këtë kudo dhe më vonë nuk e përdorim atë për të shkurtuar “*minutë”* .

Kur emërtimin e funksioneve, të përfshijë një përshkrim të vlerës duke u kthyer, të tilla si getValueLabel()(stepNum *marrë nga aplikacioni “Bazat e network-ut”*).

Emrat e file-ve dhe skedarëve/dosjeve, si emrat e procedurës, duhet të përshkruajnë me saktësi çfarë qëllimi ata shërbejnë.

Shmangni emrat ripërdorimin për elemente të ndryshme, të tilla si një emër rutinë i quajtur ProcessSales()dhe një ndryshore të quajtur iProcessSales.

Shmangni homonyms kur emërtimin elementet për të parandaluar konfuzionin gjatë shqyrtimeve të kodit, të tilla si *“write”* dhe *“right”* .

Kur emërtimin elemente, për të shmangur duke përdorur fjalë që zakonisht keqshqiptohen ose shqiptohen me vështirësi. Gjithashtu, të jenë të vetëdijshëm për dallimet që ekzistojnë mes anglishtes amerikane dhe britanike, të tilla si “color/colour” dhe “check/cheque”. Por dhe fjalë të tjera që shpesh shkruhen gabim. Kjo sepse do të ketë vështirësi në kërkim nëpër kod dhe problemi shtohet edhe më shumë kur ka më shumë se një zhvillues.

Shmangni përdorimin shenjat tipografike për të identifikuar llojet e të dhënave, të tilla si $ për stringat ose% për numrat.

Komentet

Dokumentacioni Software ekziston në dy forma, të jashtëm dhe të brendshëm. Dokumentacioni i jashtëm është ruajtur jashtë kodit burim, të tilla si specifikimet, fotografi ndihmë, si dhe dokumentet e projektimit. Dokumentacioni i brendshëm është i përbërë nga komentet që zhvilluesit shkruani në kodin burim në kohën e zhvillimit.

Një nga sfidat e dokumentacionit software është siguruar se komentet janë të mirëmbajtura dhe të përditësuar paralelisht me kodin burimor. Edhe pse duhet komentuar kodin burim i shërben asnjë qëllimi në kohë të drejtuar, ai është i paçmuar për një zhvillues i cili duhet të mbajë një pjesë veçanërisht të ndërlikuar ose të rëndë të softuerit.

Në vijim janë rekomanduar teknika komentimi:

Kur modifikoni kodin, gjithmonë mbani koment të përditësuar.

Në fillim të çdo rutine pune, ajo është e dobishme për të siguruar standarde, komente të njëllojtë, duke treguar qëllimin e punës, supozimet dhe kufizimet. Një koment unik duhet të jetë një hyrje të shkurtër për të kuptuar pse ekziston kjo punë apo ndërhyrje dhe çfarë mund të bëjë.

Shmangni duke shtuar komente në fund të një linj kodi; Komentet e end-line bëjnë kodin më të vështirë për t’u lexuar. Megjithatë, komentet fund të linjës janë të përshtatshme kur shfaqim deklarata ndryshueshme. Në këtë rast, lidh të gjitha komentet e end-line në një pozicion të përbashkët.

Shmangni përdorimin komenteve rrëmujë, të tilla si një linjë të tërë plot me ylla. Në vend të kësaj, përdorni hapësirën e bardhë për të ndarë komentet nga kodi.

Shmangni rrethimin e një blloku komenti me një kornizë tipografike. Mund të duket tërheqëse, por është e vështirë për të ruajtur.

Para vendosjes, duhet të kemi hequr të gjitha komentet e përkohshme ose pa lidhje për të shmangur konfuzionin përgjatë punës së ardhshme të mirëmbajtjes.

Nëse keni nevojë për komente për të shpjeguar një pjesë komplekse të kodit, shqyrtojeni kodin për të përcaktuar nëse ju duhet për rishkruar atë.Nëse është e mundshme, mos dokumentoni kod të keq, rishkruajeni atë. Edhe pse performanca nuk duhet të sakrifikohet zakonisht për të bërë kodin lehtë për syrin njerëzor, një ekuilibër duhet të mbahet në mes të performancës dhe mirëmbajtjes.

Përdorni fjali të plota kur shkruani komente. Komentet duhet të qartësojë kodin, nuk shtoni dykuptimësi.

Komentoni gjatë kodimit, për shkak se më shumë gjasa nuk do të ketë kohë për të bërë atë më vonë. Gjithashtu, duhet që ju të merrni një shans për të rihapur kodin që ju keni shkruar, atë që është e qartë sot ndoshta nuk do të jetë e qartë gjashtë javë nga tani.

Shmangni përdorimin e komenteve të tepërta ose të papërshtatshme, si vërejtjet humoristike.

Përdorni komente për të shpjeguar qëllimin e kodit. Ata nuk duhet të shërbejë si përkthimet fjalë për fjalë të kodit.

Koment asgjë që nuk është lehtësisht e dukshme në kodin.

Për të parandaluar përsëritjen e problemeve, gjithmonë përdorni komente në problemet e rregulluara dhe punimet rreth kodit, veçanërisht në një mjedis ekipi.

Përdorni komente mbi kodin që përbëhet nga pozicione dhe degë logjike. Këto janë fushat kryesore që do të ndihmojnë lexuesin kur lexoni kodin burim.

Ndani komentet me shenjat e hapjes (apo mbylljes) se komenteve me hapësirë ​​të bardhë. Duke bërë kështu do të bëhet që komentet të dalin të lehtë për të gjetur kur shihet pa patur ngjyra.

Gjatë aplikacionit, të ndërtohen komente duke përdorur një stil uniformë, me shenjat e pikësimit të qëndrueshme dhe strukturën.

Formati

Formatimi bën organizimin logjik të kodit në mënyrë të jashtezakonshëme. Duke marrë kohë për të siguruar që kodi burim është i formatuar në mënyrë konsistente, rrjedha logjike është e dobishme për veten tuaj dhe për zhvilluesit e tjerë të cilët duhet të deshifroj kodin burim.

Në vijim janë rekomanduar teknika formatimit:

Krijimi i një madhësi standarde për një nënpikë, si katër hapësira, dhe e përdorin atë në mënyrë të vazhdueshme.

Përdorni një font “Monospace” (font që i ka shkronjat me gjerësi të njëjtë) kur publikoni versionet “hard” të kodit burim.

Me përjashtim të konstanteve, të cilat janë të shprehura në mënyrën më të mirë në të gjitha karakteret me shkronjë të madhe me vijë poshtë, përdorin rastin e përzier në vend të “underscore” apo vijës poshtë për të bërë emrat më të lehtë për të lexuar.

Vendose kllapat të hapura dhe të mbyllura vertikalisht në një drejtim, të tilla si:

for (i = 0; i <15; i ++)

{

...

}

Ju mund të përdorni një stil pjerrët, ku kllapat hapëse hapur paraqiten në fund të rreshtit dhe kllapat mbyllëse të shfaqet në fillim të rreshtit, të tilla si:

|  |
| --- |
| if(tab.findRow(R.name,"Destinacioni")!=null){ |
| msg = "Routerat jane lidhur me pare\n"; |
| }  (*marrë nga aplikacioni “Bazat e network-ut”*)  Gjithashtu edhe stili kur kemi shume kllapa qe hapen dhe mbyllen njëra pas tjetrës   |  | | --- | | if (dist(mouseX, mouseY, position.x, position.y)<20) {  return true; | | } else { | | return false; | | } | | } | |

(*marrë nga aplikacioni “Bazat e network-ut”*)

Cilado style është zgjedhur, përdorni atë stil të gjithë kodin burim.

Krijimi i një gjatësi maksimale për komentet dhe kodet për të shmangur dritaren që të lëvizi kodin burim dhe për të lejuar për prezantim të pastër të kopjes “hard”.

Përdorin hapësira para dhe pas shumicën e operatorëve kur kjo nuk ndryshon qëllimin e kodit.

Vendos një hapësirë ​​pas çdo presje në listat presje-kufizuar, të tilla si vlerat e vektorëve dhe argumente, kur kjo nuk ndryshon qëllimin e kodit.

Përdorni hapësirë ​​të bardhë për të siguruar të dhëna organizative në kodin burimor. Duke bërë kështu krijon "paragrafët" të kodit, të cilat duhen për të ndihmuar lexuesit në të kuptuarit e segmentimi logjik të softuerit.

Kur një vijë është e thyer në disa linja, e bëjmë të qartë se linja është e paplotë pa vijën e mëposhtme.

Kur është e përshtatshme, për të shmangur vendosjen e më shumë se një deklaratë për rresht. Një përjashtim është një cikël psh: for (i = 0; i < 6; i++).

Praktikat programim

Zhvilluesit me përvojë ndjekin praktika të shumta të programimit ose “rregulla të gishtit”, e cila zakonisht rrjedh nga mësimet e mësuara me mund. Praktikat e listuara më poshtë nuk janë të gjithë-përfshirëse, dhe nuk duhet të përdoren pa u marrë në konsideratë siç duhet. Programuesit veteran devijojnë nga këto praktika me rastin, por jo pa shqyrtim të kujdesshëm të pasojave të mundshme. Duke përdorur praktikën më të mirë programimit në kontekstin e gabuar mund të shkaktojë më shumë dëm sesa dobi.

Për të ruajtur burimet, të jemi selektiv në zgjedhjen e llojit të të dhënave për të siguruar madhësinë e një variable nuk është tepër i madh.

Mbani jetëgjatësinë e variablave sa më të shkurtër të jetë e mundur, kur variablat përfaqësojnë një burim i caktuar, për të cilat mund të ketë “grindje”, të tilla si një lidhje të bazës së të dhënave.

Mbani vendodhjen e variablave sa më të vogla të jetë e mundur për të shmangur konfuzionin dhe për të siguruar mirëmbajtjen. Gjithashtu, ruajmë kodin burimor, potenciali për të thyer pa dashje pjesë të tjera të kodit mund të minimizohet, nëse shtrirja e variablit është e kufizuar.

Përdorni variablat dhe rutinat për një dhe vetëm një qëllim. Përveç kësaj, shmangni krijimin e rutinve me shumë qëllime që kryejnë një sërë funksionesh që nuk kanë lidhje.

Kur shkruani klasa, shmangni përdorimin e variablave publike. Në vend të kësaj, të përdorni procedurat/funksionet për të siguruar një shtresë e enkapsulimi dhe gjithashtu për të lejuar një mundësi për të validuar ndryshimet e vlerës.

Komponentët pa status janë të preferuar, kur shkallëzimi apo performanca janë të rëndësishme. Hartoni komponentët për të pranuar të gjitha vlerat e nevojshme si parametrave të dhëna në vend të duke u mbështetur mbi pronat e objekteve kur thërrisni metoda. Duke bërë kështu eliminon nevojën për të ruajtur objekt gjendjen mes thirrjeve të metodave. Kur është e nevojshme për të ruajtur gjendje, e konsiderojnë duke përdorur metodat alternative, të tilla si ruajtjen e gjendjes në një bazë të dhënash.

Duhet të mos mbahet hapur lidhjet me dhënat kur përdoren kredencialet e përdoruesit specifik. Lidhjet që janë hapur duke përdorur kredenciale të tilla nuk mund të grumbullohen dhe të ripërdoren, duke humbur kështu të mirat e lidhjeve.

Shmangni përdorimin e detyruar të dhënat e konvertimit, të cilat mund të japin rezultate të papritura. Kjo ndodh kur dy ose më shumë variabla të llojeve të ndryshme të të dhënave janë të përfshirë në të njëjtën shprehje. Kur është e nevojshme për të kryer një kastim për një arsye të parëndësishme, duhet të jepet në një koment shoqërues.

Shmangni përdorimin e variablave në fushën e zbatimit.

Përdorni RETURN në procedurat e ruajtura për të ndihmuar programin nëse e di se procedura ka punuar si duhet.

Përdorni teknika detyruese që në fillim kur është e mundur.

Përdorni  Switch në vend të kontrollit të përsëritura të një variable të përbashkët.

Përfundime

Përdorimi i rregullt i teknikave të kodimit dhe praktikave të mira të programimit i cili krijon kod më aftësi apo cilësi të mira luan një rol të rëndësishëm në performancën dhe kualitetin e software-it. Gjithashtu kjo ndihmon në rishikimet e kodit, e cila thjeshtëson punën në rast të problemeve apo zhvillimeve të mëtejshme. Kjo do të sillte kursim të madh në kohë pasi nuk do të duhet të shkruajmë gjithçka nga e para.

Zhvillime të mëtejshme të aplikacionit

Zgjidhja me një aplikacion mobile