

# **RoCyberCon #3 – 6 martie 2011**

## **- prima conferinta online din Romania a ITistilor -**

*“Linux & Windows si OOP in PHP + lansarea WFC”*

<http://rocybercon.worldit.info:9090>

### ***Cuprins :***

- 1. WFC – un bit poate ascunde multe...**
- 2. Medii grafice linux, interconectarea windows – linux si alternative windows la programe linux**
- 3. OOP in PHP**

## WFC – un bit poate ascunde multe...

[19:05] <@Andrei> Buna seara si bine ati (re)venit la RoCyberCon, acum ajuns la a treia editie :D  
[19:05] == Catalin has changed nick to malwarewiki  
[19:06] <@Andrei> va reamintesc ca pentru a avea acces la prezentarile ce le folosim in timpul conferintei trebuie sa fiti autentificati pe clientul nostru web  
[19:06] <@Andrei> seara, va avea un program mai interesant ca pana acum  
[19:06] <@Andrei> voi incepe eu prin prezentarea unui subiect surpriza  
[19:06] <@Andrei> apoi johane va continua pe partea de Linux : medii grafice, interconectare si alternative la aplicatii windows pentru linux  
[19:07] <@Andrei> iar Nytro va incheia prin o discutie extrem de interesanta legata de PHP & OOP :D  
[19:07] <@Andrei> fara prea multe alte comentarii, propun sa-i dam drumul si voi scoate mute-ul, daca e necesar il voi reactiva :D  
[19:07] <@Andrei> .mode -m  
[19:07] == mode/#rocybercon [-m] by Guardian  
[19:07] <@Andrei> \$slideshare\_open\_wfc-unbitpoateascundemulte-110306094804-phpapp01  
[19:07] <EMz> salut baieti  
[19:08] <Spock> :)  
[19:08] <@Andrei> tema din aceasta seara - WFC sau un nou eveniment marca worldit  
[19:08] <@Andrei> \$slideshare\_page\_1  
[19:08] <@Andrei> \$slideshare\_page\_2  
[19:08] <Bercy> Zack Boss lovitura de baros  
[19:08] <@TheJuderger> .ban Bercy  
[19:08] == mode/#rocybercon [+b \*!\*web\*a\*@\*?156?93A\*?IP] by Guardian  
[19:08] <@Andrei> dupa cum vedeti  
[19:09] <@Andrei> avem o idee noua, multumita lui George ,unul din redactorii activi ai blogului  
[19:09] <@Andrei> apoi norocul a facut sa dea peste worldit (norocul nostru :D)  
[19:09] <@Andrei> cativa amici de la bitdefender & creativity ne sunt alaturi :)  
[19:09] <@Andrei> un eveniment - rocybercon #3 cea mai buna ocazia de a-l anunta  
[19:09] == mode/#rocybercon [-b \*!\*web\*a\*@\*?156?93A\*?IP] by TheJuderger  
[19:09] <@Andrei> si o data - 7 martie, sau maine  
[19:10] <@Andrei> \$slideshare\_page\_3  
[19:10] <@Andrei> procesam toate astea, am dezbatut, analizat si regandit  
[19:10] <@Andrei> avem in joc premii importante, se va desfasura pe blog :D  
[19:10] <@Andrei> fara amestecul magic nu ar iesi ceva frumos, nu? :D  
[19:10] <@Andrei> rezultatul?  
[19:10] <@Andrei> \$slideshare\_page\_4  
[19:10] <@Andrei> \$slideshare\_page\_5  
[19:10] <@Andrei> nu-i sfarsitul lumii :D  
[19:11] <@Andrei> doar o mica eroare tehnica :D  
[19:11] <@Andrei> \$slideshare\_page\_6  
[19:11] \* pastoru slaps TheJuderger around a bit with a large fishbot  
[19:11] <TinKode> cati sunteti ba in plm :))  
[19:11] <@TheJuderger> .ban TinKode  
[19:11] == mode/#rocybercon [+b \*!\*eb?ha\*@\*.96EEA\*AC.??] by Guardian  
[19:11] <@Andrei> WFC - worldit forensic contest :D  
[19:11] <@Andrei> noul eveniment marca worldit, eveniment care speram noi sa va tina in suspans si sa va puna la treaba intr-un mod productiv, interesant de pe urma caruia veti avea de castigat

[19:12] <@Andrei> avem niste scopuri clare :

[19:12] <@Andrei> Scopul acestor probe este de a evidentia problemele de securitate ce sunt prezente in sistemele informatice care sunt folosite in activitatea zilnica, de a informa utilizatorii asupra unor metode de contracare a lor si de a aduce la cunostinta oamenilor modalitatile in care se desfasoara procesele de auditare a unui sistem informatic.

[19:12] <@Andrei> \$slideshare\_open\_wfc-unbitpoateascundemulte-110306094804-phpapp01

[19:12] <@Andrei> \$slideshare\_page\_6

[19:12] <@Andrei> va fi un eveniment lunar

[19:13] <@Andrei> impartit pe 3 perioada

[19:13] <@Andrei> perioade\*

[19:13] <@Andrei> perioada de teasing (ii spunem noi) - o saptamana in care oricine doreste sa participe la o editie noua se inscrie. avantajul lor va fi ca vor primi proba in maxim 24 de ore de la inscriere

[19:14] <@Andrei> perioada publica - 2 saptamani in care vom publica si oferi timp participantilor sa rezolve (problemele nu vor fi atat de dificile, dar e bine sa aveti timp) :D

[19:14] == mode/#rocybercon [-b \*!\*?eb?ha?@\*.96EEA\*AC.??] by TheJudger

[19:14] <@Andrei> si ultima perioada in care vom afisa castigatorii, rezultatele si solutiile la problemele dezbatute :)

[19:14] <@Andrei> \$slideshare\_open\_wfc-unbitpoateascundemulte-110306094804-phpapp01

[19:14] == s3v3n [webchat@WIT-D4D05CF1.b.ipv4ilink.net] has left #rocybercon []

[19:15] == deneo [webchat@WIT-820DC26F.ambra.ro] has left #rocybercon []

[19:15] <@Andrei> premiile spun eu ca vor fi extrem de interesante, avand in vedere ca sponsorii s-au gandit la tricouri, licente si chiar produse si aparatura hardware :)

[19:15] <@Andrei> \$slideshare\_page\_7

[19:15] <cyder> neata

[19:15] <cyder> am pierdut ceva?

[19:15] <malwarewiki> tot

[19:15] <@Johane> Andrei: Ce tip de probleme vor fii?

[19:15] <BGS> nu ai pierdut nimic ia un loc si taci :)

[19:16] <@Andrei> vor fi probleme de forensic, asa cum ii spune si titlul

[19:16] \* raven slaps Nytro around a bit with a large fishbot

[19:16] <@Andrei> de la analiza unor snippeturi

[19:16] <Spock> Challenges vor tine numai de web-testing? sau si alte domenii? Criptografie, social engineering, securitatea pe servere Linux etc.?

[19:16] <@Andrei> pana la analiza imaginilor, hard disk-urilor corupte samd

[19:16] <Spock> Ok.

[19:16] <@Andrei> steganografie samd

[19:16] <@Andrei> :)

[19:16] <reV`> CSI: Romania xD

[19:16] <malwarewiki> De cine au fost facute

[19:16] <blackboy1337> Andrei cand ii dai drumu ?

[19:17] <@Andrei> maine va fi dat drumul

[19:17] <s3v3n> e ceva activitate aici? sau doar eu nu primesc nimic :|

[19:17] <@Andrei> malwarewiki Problemele sunt propuse de mine, Ilie George, si el un veteran al evenimentelor de acest gen din afara :)

[19:17] <YcreB> sall all

[19:18] == YcreB [webchat@4161592F.FBAEF41A.B4903D63.IP] has left #rocybercon []

[19:18] <bcman> salut tuturor

[19:18] <SENEQ\_o> cum te inscrii?

[19:18] <@Andrei> inscrierea se va face printr-un formular  
[19:18] <@Andrei> formular disponibil de maine :)  
[19:18] <ghostwhite> conteaza modalitatea de rezolvare sau sky is the limit  
[19:18] <pr00f> Exista un minimum de varsta ?  
[19:18] <SENEQ\_o> ms  
[19:19] <@TheJudger> .ban s3v3n  
[19:19] == mode/#rocybercon [+b \*!\*w\*?chat@\*?i??4\*1??\*.???] by Guardian  
[19:19] <@Andrei> nu  
[19:19] <@Andrei> oricine poate citi, scrie si raspunde poate participa :)  
[19:19] <@Andrei> singura restrictie e sa fie din romania (sa traiasca aici) :D  
[19:19] <ghostwhite> juriul?  
[19:20] <@Andrei> nu va fi un juriu, cel care va propune challenge-ul va avea un barem pe care jurizeaza clar - 1p pentru ca ai ajuns pana aici, 2 ca ai ajuns pana aici samd :)  
[19:20] <@Andrei> la final se trage linie, se aduna si clasamentul se publica asa cum e el :)  
[19:20] <@Andrei> in cazul in care vor fi doi sau mai multi la egalitate  
[19:20] <Spock> vreau si io un fullscreen <http://static.slidesharecdn.com/swf/ssplayer2.swf>  
[19:21] <s3v3n> i'm here  
[19:21] <@Andrei> vom face o tragere la sorti (asta doar daca numarul celor la egalitate depasesc premiile alocate)  
[19:21] <ghostwhite> baraj la puncte egale sau cel mai bun timp  
[19:21] <@Andrei> ghost dureaza prea mult :)  
[19:21] <ghostwhite> ok  
[19:21] <Fi8svrs> cand incepe?  
[19:21] <@Andrei> maine :)  
[19:21] <@Andrei> \$slideshare\_open\_wfc-unbitpoateascundemulte-110306094804-phpapp01  
[19:21] <@Andrei> alte intrebari? :D  
[19:21] <blackboy1337> la ce ora andrei?  
[19:22] <@Andrei> cand reusesc sa termin regulamentul si anuntul :D  
[19:23] <blackboy1337> Andrei ce multi sunt :D  
[19:23] <@Andrei> daca nu il vom lasa pe johane sa ne duca prin lumea linuxului :D  
[19:23] <cyder> yeey  
[19:23] <@Andrei> daca veti mai avea intrebari, nu ezitati sa luati legatura cu mine, regulamentul va fi afisat maine :)  
[19:24] <@TheJudger> ca sa vorbiti pe privat cu Andrei, click pe nickname-ul lui si apoi Query :)  
[19:24] <@Andrei> da :)  
[19:24] <Spock> <http://spock.hi2.ro/slide1.html>

## **Medii grafice linux, interconectarea windows – linux si alternative windows la programe linux**

[19:24] <@Johane> Acum ca nu aveti întrebări o sa începem incursiunea în lumea linux  
[19:24] <BGS> can i haz za slide?:P  
[19:24] <@Andrei> sure, private :D  
[19:25] <@Johane> O sa încep cu o mica întrebare.  
[19:25] <Spock> pt aia care au pierdut  
[19:25] <@Johane> Căai din voi folositi linux în prezent?

[19:25] <@TheJudger> me  
[19:25] <reV``> o/  
[19:25] <dimitriepirghie> Eu asta astept partea cu linuxu, dar eu folosesc  
[19:25] <@Andrei> eu :)  
[19:25] <dimitriepirghie> johane  
[19:25] <BGS> can i haz slideshow ?:D  
[19:26] <@Johane> Bun atunci  
[19:26] <Zack> Dc nu merge slidshare-ul ??  
[19:26] <@Johane> Deja v?d c? sunt utilizatori  
[19:26] <@Andrei> ups, something happened :)  
[19:26] <@TheJudger> Andrei, cred ca ai dat nume gresit  
[19:26] <@Johane> O s? prin a v? spune c? linux are foarte multe medii grafice  
[19:26] <@Andrei> nu, la mine s-a deschis :)  
[19:26] <Spock> mai multe ca BSD, johane  
[19:26] <reV``> and it died  
[19:26] <Spock> ca tot vorbeam aseara  
[19:27] == pr00f [webchat@785EF2D9.20F682C6.1B258699.IP] has left #rocybercon []  
[19:27] == s3v3n [webchat@WIT-D4D05CF1.b.ipv4ilink.net] has left #rocybercon []  
[19:27] <Xenon> test  
[19:27] <BGS> presimt un brainstorming intens pe tema asta johane :))  
[19:27] <s3v3n1> .join  
[19:27] <reV``> Andrew, banui ca a crapat webchatu'  
[19:27] <@TheJudger> da  
[19:27] <@Johane> test  
[19:27] <SENEQ\_o> :D am fost la baie si credeam ca so terminat  
[19:27] <pr00f> crapase chat-ul ?  
[19:28] <bluexaren> da  
[19:28] <BGS> da  
[19:28] <epsilonomega13> pr00f, da  
[19:28] <robert> da  
[19:28] <pr00f> dammit  
[19:28] <@Andrei> da, a picat din cauze ce nu le stiu inca, asteptam 2 minute sa-si revina :)  
[19:28] <@Andrei> sa reintrati ma refer :)  
[19:28] <@Johane> Pauz? 2 minute din cauza unor mici probleme tehnice  
[19:28] <reV``web> merge, ce-i drept  
[19:28] == reV`` has changed nick to reV  
[19:28] <@Johane> Continuate  
[19:29] <robert> [18:27] == 2 operator(s) online [18:27] == 5 channels formed  
[19:29] <@Johane> Cum spuneam mai devreme linux are foart emulte medii grafice  
[19:29] <@Johane> Unele din ele sunt foarte elemntare, altele sunt foarte avansate, cu mult mia avansate decat windows chiar  
[19:29] <bcman> si mie mi s-a blocat cateva secunde  
[19:29] <BGS> liniste va rog lasatil sa spuna!  
[19:29] <bcman> si am facut refresh la pagina  
[19:30] <Spock> da-ti mute la useri!!!  
[19:30] <BGS> ok a crapat putin serverul acum lasatil sa spuna prezentarea  
[19:30] <@Andrei> .mode +m  
[19:30] == mode/#rocybercon [+m] by Guardian  
[19:31] == pirdion [dimitrie@92.86.30.210] has left #rocybercon []

[19:33] \* Andrei Datorita problemelor tehnice neasteptate vom renunta pentru putin timp la sincronizare si vom merge clasic. Prima prezentare : <http://www.slideshare.net/RoCyberCon/medii-grafice-linux>

[19:33] <@Johane> Se pare ca din cauza unor probleme tehnice nu osa vedem slideurile

[19:33] <@Johane> Lxde este un mediu grafic adresat calculatoarele slabe si foarte slabe

[19:34] <@Johane> Nu consuma memorie aproape deloc si nici putere de procesare

[19:34] <@Andrei> .mode -m

[19:34] == mode/#rocybercon [-m] by Guardian

[19:34] <Spock> johane, se vede slideul

[19:35] <@Johane> Foloseste ca manager de fisiere PCManFm

[19:35] <cyder1> eu nu il vad

[19:35] <pr00f> Se vede slide-ul, si putem folosi metoda manuala de click in partea dreapta, scuzati-mi interventia

[19:35] <@Johane> Este bazat pe OpenBox

[19:35] <@Andrei> Daca mentineti offtopic, voi fi nevoit sa dau din nou mute. Prezentarea e aici - <http://www.slideshare.net/RoCyberCon/medii-grafice-linux>

[19:36] <@Johane> Personalizarea lui lxde in cel mai bun caz este modificarea pozitiei barelor si a culorilor acestuia

[19:36] <@Johane> Nu este printre cele mai customizabile medii grafice in consecinta

[19:36] == mode/#rocybercon [+o Nytro] by Guardian

[19:36] <@Johane> Vom trece acum la xfce

[19:37] <@Johane> Un mediu grafic mai avansat decat lxde, bazat pe Gtk+

[19:37] <@Johane> gtk+ este toolkitul scris pentru gimp

[19:37] <@Johane> lxde are un consum putin mai mare de resurse decat lxde dar nu cu mult mai mare

[19:38] <@Johane> Personal il rulez pe un single core la 1,7 Ghz si 512 Mb ram si merge foarte bine

[19:38] <Spock> xfce\*

[19:39] <Spock> xfce are un consum mai mare de resurse

[19:39] <Spock> nu lxde

[19:39] <@Johane> Este mult mai customizabil decat lxde si are printre alte avntaje si transparenta ferestrelor

[19:39] <bcman> suporta efecte gen compiz sau altele?

[19:39] <BGS> buna intrebare dar haideti sa le pastram pentru final :)

[19:39] <reV`web> intrebarile la sfarsit. lasa omu' sa termine

[19:40] <@Johane> Vom trece acum la E17-Enlightment

[19:40] <@Johane> Din punctul meu de vedere este unu din cele mai frumoase medii grafice linux

[19:41] <@Johane> Efectele sale sunt foarte interesante, si curios este faptul ca nu necesita accelerare hardware, dar este recomandata

[19:42] <@Johane> Este bazat pe ELF( Enlightenment Library Foundation )

[19:42] <@Johane> Elf este folosit si pe telefoane mobile

[19:43] <@Johane> Gnome

[19:43] <@Johane> Este printre cele mai utilizate medii grafice din linux

[19:44] <@Andrei> sa inteleg ca ELF sta la baza secretului care face din E17 un distro fara necesitatea accelerarii hardware?

[19:44] <@Johane> Andrei: da

[19:45] <@Johane> Este bazat si el la randul sau pe GTK+

[19:45] <Maverick> what??

[19:46] <@Johane> Optiunile sale de configurare sunt aproape nelimitate

[19:46] <@Johane> Singurul lucru care ne limiteaza la gnome este imaginatia si cunostintele

[19:46] <@Andrei> @Maverick Vorbim de E17-Enlightment. :)

[19:46] <BGS> vrei sa zici GNOME  
[19:46] <@Johane> Andrei: gnome  
[19:47] <@Andrei> da:))  
[19:47] == rimageo [webchat@E80EE808.A4D78910.41AC7E87.IP] has left #rocybercon []  
[19:47] <@Johane> Pentru efecte grafice complete este necesara activarea accelerari hardware  
[19:48] <@Johane> Daca placa video este suportata de linux instalarea driverelor se face foarte usor  
[19:48] <@Johane> Toate aceste efecte si briz-brizuri vin cu un cost semnificativ de memorie si putere de procesare  
[19:48] <@Johane> \*putere  
[19:49] <@Johane> KDE-Qt  
[19:49] <@Johane> Are cineva de aici telefoane nokia cu symbian?  
[19:49] <s3v3n1> here  
[19:49] <ghitzZza> da  
[19:49] <pr00f> prezent  
[19:49] <@TheJudge> Johane cum ne dam seama ce sistem de operare avem pe telefon ? :)  
[19:49] <@Johane> Poate nu stiti dar simbyam foloseste acelasi mediu grafic ca si KDE  
[19:49] <@Johane> Adica QT  
[19:49] <Maverick> @Johane Nokia N8 ai? cu factura..  
[19:49] <Maverick> ?  
[19:50] <s3v3n1> despre care symbian mai exact vorbesti ?  
[19:50] <@Johane> TheJudge: Daca il vad imi dau seama si eu  
[19:50] <@Johane> Nu am telefon cu symbian deci nu stiu  
[19:50] <s3v3n1> @Johane: Qt nu este mediu grafic  
[19:50] <@Johane> s3v3n: toate symbian, maemo  
[19:50] <s3v3n1> KDE este mediu grafic  
[19:50] <Spock> thejudge, fara mistouri  
[19:50] <ghitzZza> poi e bazat pe Qt  
[19:50] <Spock> Qt e toolkit.  
[19:51] <@Johane> Sa revenim acum la KDe  
[19:51] <@Johane> Ultima versiune de KDE este bazata pe Qt4.4  
[19:51] <kNigHt> salut :)  
[19:52] <@Andrei> salut & shht  
[19:52] <@Johane> De la versiunea 4.0 s-au adus multe imbunatatiri la qt  
[19:52] <Paul4games> ce de lume:)  
[19:52] <bluexaren> taci ba  
[19:52] <Spock> da, nu mai e la fel ca KDE3. de-aia si-a pierdut si multi useri... :(  
[19:52] <@Johane> Spock: da asa e  
[19:52] <reV> se poate un +m? mersi.  
[19:53] <@Johane> Acum foloseste ceva mai multe resurse decat qt3 dar se imbunatesc permanent atat consumul cat si performanta  
[19:53] <@Johane> Este mediul cel mai recomandat incepatorilor deoarece se aeamana cu windows foarte mult  
[19:54] <@Johane> Setarile sale le depaseste pe cele ale Gnome  
[19:54] <@Johane> Se poate seta in mod si chip  
[19:54] <BGS> scuze dar gnome era cel mai apropiat de windows :)  
[19:54] <@Johane> BGS: fals  
[19:54] <@Johane> KDE este  
[19:54] <@Nytro> Nu, KDE  
[19:55] <BGS> ok vom dezbate mai tarziu pe treaba asta

[19:55] <reV> KDE intr-adevar  
[19:55] == DrOptix [webchat@4D9CDF8B.BE61D9E0.673BE40F.IP] has left #rocybercon []  
[19:55] <BGS> rev invata sa scriii intradevar dupa dati cu parerea.  
[19:55] <@TheJudger> .ban bgs  
[19:55] == mode/#rocybercon [+b \*!\*w?bc?at@\*?7\*8?5??2.\*P] by Guardian  
[19:55] <@TheJudger> calmeaza-te  
[19:56] <bluexaren> lasalatil ma pe rev in pace  
[19:56] <@Andrei> .unban bgs  
[19:56] == mode/#rocybercon [-b \*!\*w?bc?at@\*?7\*8?5??2.\*P] by Guardian  
[19:56] <@Andrei> .mode +m  
[19:56] == mode/#rocybercon [+m] by Guardian  
[19:56] <@Johane> Kde este cel mai "mancacios mediu" grafic linux prin memoria consumata si puterea de procesare  
[19:57] <@Johane> Intrebari pana acum?  
[19:57] <@Andrei> .mode -m  
[19:57] == mode/#rocybercon [-m] by Guardian  
[19:57] <s3v3n1> ai zis ceva de gimp si gtk+  
[19:57] <Spock> doar corectari. Qt e toolkit :)  
[19:57] <pr00f> aprob Spock  
[19:57] <epsilonomega13> s3v3n, spune  
[19:57] <Spock> da, gtk+ a fost initial folosit ca sa se construiasca GIMP  
[19:58] <s3v3n1> "initial" face diferenta  
[19:58] <s3v3n1> apoi  
[19:58] <pr00f> gtk+ = gimp toolkit plus ?!  
[19:58] <s3v3n1> aveam o interbare referitor la symbian si qt  
[19:58] <epsilonomega13> BTW:Eu si johane suntem aceeasi persoana  
[19:58] <SENEQ\_o> cam cat de bun tb sa fie calculatorul sa mearga bine KDE ( niste specificatii medium)  
[19:58] <s3v3n1> si as spune ca symbian nu e facut pe qt  
[19:58] <s3v3n1> ci in qt pot fi facute aplicatii pt symbian  
[19:58] <s3v3n1> adica symbian suporta qt, dar nu e facut pe  
[19:58] <reV`> @Johane: Daca ar fi sa faci un clasament dpdv al "eye candy"-ului, cum ar sta treaba?  
[19:58] <s3v3n1> bazat pe \*  
[19:58] <epsilonomega13> s3v3n, symbiam foloseste qt sa afiseze tot pe crean  
[19:58] <Brabus> Salut  
[19:58] <Spock> asa se numeste. +-ul ala e doar de estetica acolo  
[19:58] <@Nytro> Qt nu inseamna numai elemente grafice  
[19:59] <Spock> Alte intrebari?  
[19:59] <epsilonomega13> rev ordinea ar fii in ordinea expusa de mine adica lxde->xfce->e17->gnome->kde  
[19:59] <Spock> Sincer, johane, ma asteptam sa faci o greseala si sa consideri Fluxbox Desktop enviroment :-D cand ii window manager...  
[19:59] <BGS> de ce kde este cel mai apropiat de windows?  
[20:00] <Spock> pai, compara designul  
[20:00] <epsilonomega13> SENEQ\_o, un dual core si un 1Gb de ram merge super  
[20:00] <Spock> si ai sa vezi  
[20:00] <s3v3n1> pentru ca vizual seamana cel mai mult  
[20:00] <BGS> da l-am comparat si nu prea are vreo legatura cu windows  
[20:00] <epsilonomega13> Spock, :D



[20:00] <BGS> gnome mi se pare cel mai apropiat  
[20:00] <Spock> Compara pisi designul  
[20:00] <BGS> si nu e doar fiindca spun eu la installerul mandriva  
[20:00] <Spock> ti se pare gnome mai apropiat de windows?  
[20:00] <s3v3n1> @BGS: incearca si gnome, si kde. ai sa intelegi diferenta dupa ce le incerci  
[20:00] <BGS> spune clar gnome cel mai apropiat de windows  
[20:00] <BGS> s3ven1 le-am incercat stai linistit  
[20:00] <epsilonomega13> BGS, kde e cel mai apropiat de windows  
[20:01] <BGS> revin cu argumente brb  
[20:01] <epsilonomega13> Ca design este aproape identic la prima vedere  
[20:01] <SENEQ\_o> ms ... mai am o intrebare ... care este cel mai fiabil mediu grafic ? din punct de vedere al bugurilor si al supportului( updatari )?  
[20:01] <reV'> uita-te mai bine BGS: <http://www.kde.org/screenshots/>  
[20:01] <+Cyborg> Title: KDE - Screenshots (at [www.kde.org](http://www.kde.org))  
[20:01] <pr00f> Ce distributie linux 'domneste' din punct de vedere al frumusetii grafice ?  
[20:01] <epsilonomega13> SENEQ\_o, toate se pot folosi fara probleme  
[20:01] <s3v3n1> bine, poate e mai apropiat kde de windows pentru ca odata cu kde vin si cateva aplicatii care amintesc windows mai mult decat ubuntu de exemplu  
[20:01] <SENEQ\_o> ms  
[20:01] <epsilonomega13> Doar ca gnome ai sanse sa il strici mai greu  
[20:02] <s3v3n1> ar fi si device managerul care by default in ubuntu lipseste  
[20:02] <pr00f> Ce distributie linux 'domneste' din punct de vedere al frumusetii grafice ?  
[20:02] <epsilonomega13> pr00f, sunt multe distributii, fiecare cu frumusetea ei  
[20:02] <@Nytro> Ce intrebari...  
[20:02] <Patrunjel> Buna seara natiune :)  
[20:02] <epsilonomega13> Nytro, intrebari de incepatori :)  
[20:03] <s3v3n1> @pr00f, majoritatea au si versiuni kde si gnome, deci nu poti sa afirmi ca ar "domni" cineva in domeniu  
[20:03] <m0rphic> foarte frumoase screenshot-urile KDE  
[20:03] <epsilonomega13> Cum nu mai sunt intrebari o sa continui cu urmatoarea parte  
[20:03] <SENEQ\_o> se pot pune simultan mai multe medii grafice pe un calculator ?  
[20:03] <BGS> da  
[20:03] <epsilonomega13> SENEQ\_o, da se pot  
[20:03] <Spock> lol seneq iesi de pe irc acuma :))  
[20:03] <SENEQ\_o> :))  
[20:03] <SENEQ\_o> era pt nytro intrebarea:D  
[20:03] <@Andrei> baieti, nu suntem aici la ora de talente :)  
[20:04] <Shady21> Buna seara baietii si eventuale fete !  
[20:04] <@Nytro> Simultan nu, dar se pot schimba rapid  
[20:04] <SENEQ\_o> =))  
[20:05] <s3v3n1> wiki Symbian:  
[20:05] <s3v3n1> Default user interface      Currently based on S60 platform, later to use Qt  
[20:05] == s3v3n1 has changed nick to s3v3n  
[20:06] <epsilonomega13> Alte intrebari?  
[20:06] <@Andrei> baieti, mai sunt nelamuriri? Daca nu johane e nerabdator sa treaca mai departe pentru ca Nytro nu mai rezista in banca lui. :D  
[20:06] <s3v3n> ce inseamna simultan? sa fie instalate toate, sau sa fie folosite toate?  
[20:06] <s3v3n> @SENEQ\_o  
[20:06] \* Nytro slaps Andrei around a bit with a large fishbot

[20:06] <SENEQ\_o> sa fie instalate  
[20:06] <epsilonomega13> s3v3n, sa fie instalate toate, dar nu folosite toate in aceeasi secunda  
[20:06] <s3v3n> da, oricate  
[20:07] <BGS> da se poate  
[20:07] <SENEQ\_o> da am inteles=))  
[20:07] <BGS> " \_ "  
[20:07] <epsilonomega13> Alte intrebari?  
[20:07] <@Andrei> good, trecem mai departe la urmatoare parte sustinuta de johane, care imi place mie cel putin :D Nu vom risca sa folosim sincronizarea slideshare asa ca ramenm pe clasic :  
<http://www.slideshare.net/RoCyberCon/interconectarea-linux-windows>  
[20:07] <@Andrei> .mode +m  
[20:07] == mode/#rocybercon [+m] by Guardian  
[20:09] <@Andrei> .op Johane  
[20:09] == mode/#rocybercon [+o Johane] by Guardian  
[20:09] <@Johane> Mersi  
[20:10] <@Johane> Cu toate ca linux poate sa semene foarte bine cu windows , ele sunt fundamental diferite  
[20:10] <@Johane> Interconectarea sau transferul de fisiere intre cele 2  
[20:10] <@Johane> O sa incep cu banalul mail  
[20:11] <@Andrei> .mode -m  
[20:11] <@Johane> Este o metoda foarte buna pentru transferul de fisiere mici ( undeva pe la 2 MB pe mail)  
[20:11] == mode/#rocybercon [-m] by Guardian  
[20:12] <@Johane> Mesajele ajung instant la destinatie  
[20:13] <@Johane> Se pot trimite in diverse retele  
[20:13] <@Johane> Ex:de la firma acasa  
[20:13] <@Johane> Gmail->Yahoo! etc.  
[20:13] <Birkoff> nu ajung chiar instant... depinde si de configurarea serverului si de incarcarea retelei  
[20:13] <epsilonomega13> Birkoff, chiar daca nu e instant nu fac mult timp  
[20:14] <@TheJudger> exista mai demult gmail drive care iti oferea nush cati gb de stocare pt fisiere, nu stiti mai exista ? ;)  
[20:14] <@Johane> Cu toate ca rapida si facila aceasta metoda are un mare dejavantaj, si anume dimensiunea fisierelor  
[20:14] <Birkoff> thejudger daca esti grabit vezi ultimul slider...  
[20:14] <Paul4games> parca inca mai merge  
[20:15] <epsilonomega13> TheJudger, exista o alte versiune acum  
[20:15] <epsilonomega13> E pentru google docs si tine fisiere de 2Gb marime  
[20:15] <@TheJudger> bun  
[20:15] <epsilonomega13> In limita spatiului disponibil  
[20:15] <reV`web> problema cu Gmail drive e limita de 25mb/fisier  
[20:15] <@Johane> Vom trece acum sa vorbim despre DropBox  
[20:16] <@Johane> Cine a folosit sau foloseste dropbox?  
[20:16] <s3v3n> here  
[20:16] <@TheJudger> me too  
[20:16] <reV`web> o/  
[20:16] <@Johane> Bun :0  
[20:16] <Paul4games> folosesc dropbox de aproape un an si nu am probleme cu el  
[20:16] <@Johane> asta inseamana ca profitati din plin de avantajele sale  
[20:17] <Paul4games> da este destul de bun cand fac format la tot hardul,pun in dropbox ce consider

eu important iar dupa ce instalez iara windowsul doar descarc dropboxul

[20:18] <@Johane> Interconectarea intre diverse sisteme de operare, printre care si o versiune inca alfa de dropbox pentru android

[20:18] <Birkoff> intrebare - cat de sigur este transferul si salvarea datelor sensibile prin astfel de metode? (ma refer la dropbox si metode similare) adica fiind pe net chiar daca sunt in cloud totusi de unde stii ca altcineva nu poate avea acces la ele?

[20:18] <epsilonomega13> Birkoff, daca iti sparge cineva contul nu ai nici o garantie

[20:19] <epsilonomega13> Poate pune mana pe toate datele tale

[20:19] <Birkoff> stiu daia nici nu folosesc astfel de metode... prefer hdd portabil... eventual datele criptate pe usb

[20:19] <Paul4games> doar daca ia cineva access la contul tua,alta cale nu vad decat sa iei access eventual al serverele dropbox

[20:20] <@Andrei> Birkoff a pus o intrebare desteapta - cat de sigur stau informatiile in acel cloud? Care e garantia ca integritatea si confidentialitatea lor nu e afectata? Au fost scandaluri sau chestii de genu? :)

[20:20] <Paul4games> stiu ca am vazut ceva similar cu dropboxul doar ca iti crypta datele cand le uploada iar cand le downloada le decrypta sau cv de genul

[20:20] <@Johane> Daca tin eu minte limita la dropbox pe fisier este 200 Mb per fisier

[20:20] <Birkoff> stiu daia am si pus intrebarea... cloud e chinul adminilor de sistem :D

[20:21] <epsilonomega13> Andrei, ele sunt sigure dar nu as recomanda sa fie tinute acolo date extrem de sensibile

[20:22] <@Johane> Vom trece acum la samba

[20:22] <s3v3n> oricum, datele extrem de sensibile/importante ar fi informatii textuale, ceea ce cred ca nu avem o problema in a le stoca undeva "la sigur", din moment ce nu ocupa prea mult spatiu

[20:23] <Birkoff> dupa cum vad eu treburile, orice transfer de date pe net este nesigur, de unde stii ca nu este bagat un program de monitorizare in reseaua interna sau pe serverul providerului (care a fost compromis) sau pe undeva intre mine si destinatar... oricine care stie si are acces poate intercepta aceste date... deci concluzia e = datele importante nu le transfera in retea sau pe net... doar pe local / hdd extern / banda magnetica

[20:23] <@Johane> Samba share este o modalitate de a conecta un linux cu windows

[20:23] <kNigHt> din cate stiu dropbox tine datele in aes256

[20:23] <Birkoff> nu e vb de datele ajunse deja la destinatie... e vb de pana ajung acolo

[20:23] <s3v3n> @Birkoff: true.

[20:23] <@Nytro> Nimic nu e sigur, nici local pe hard portabil, se poate pierde, se poate fura

[20:24] <s3v3n> vreti siguranta? criptati.

[20:24] <Birkoff> trec prin reseaua interna a firmei, trec prin serverele providerului, trec prin cine stie ce alte locatii pana ajung la destinatie... pe tot acest traseu pot fi interceptate

[20:24] <kNigHt> n-or folosi ssl?

[20:24] <@Johane> Daca aceasta conectare si stocare este facuta corect poate fi foarte sigura

[20:24] <@Johane> Baie?i va rog, discutam la pauza

[20:24] == mode/#rocybercon [+1 100] by TheJudger

[20:24] <s3v3n> ok

[20:25] <@Johane> Daca aceasta conectare si stocare este facuta corect poate fi foarte sigura

[20:25] == mode/#rocybercon [-b \*!\*e?c??t@\*??EE?9??\*?P] by TheJudger

[20:25] <ValerianVreme> experienta, daca imi scuzati interpunerea, arata ca marile probleme de securitate au tinut mai mult de lipsa unei minime decente in protectia datelor

[20:25] <ValerianVreme> siguranta ca atare nu exista

[20:25] <ValerianVreme> e doar un principiu

[20:25] <reV`> ochiu' ager

[20:26] <@Johane> Stocarea se face pe un alt pc din retea locala, pe un hdd crpitat  
[20:26] <@Andrei> Da, dar nu te opreste nimeni sa le criptezi inainte de a le urca undeva :)  
[20:26] <@Andrei> sau a le pune undeva  
[20:26] <Birkoff> da, principiul este "cum sa facem datele cat mai greu de accesat de cineva neautorizat" si asta implica atat transferul lor cat si stocarea si citirea lor in siguranta  
[20:26] <Birkoff> cu asta am incheiat las omul sa prezinte  
[20:26] <s3v3n> <https://www.dropbox.com/help/27>  
[20:26] <s3v3n> dropbox foloseste si ssl, si criptare la destinatie  
[20:27] <+Cyborg> Title: Dropbox - How secure is Dropbox? - (at www.dropbox.com)  
[20:27] <s3v3n> scz. next plz  
[20:27] <ValerianVreme> cheia hardware ramane de baza  
[20:27] <epsilonomega13> ValerianVreme, cheia hardware la care nu are acces nimeni  
[20:27] <epsilonomega13> Ca daca ai acces la ea sa dus totul de rapa  
[20:28] <epsilonomega13> Deci sa continuam  
[20:28] <@Andrei> sa nu ajungem sa intram in subiectul asta ca va fi o concluzie de genul - lipsa unor periferice, pc intr-un seif samd sunt o solutie ideala  
[20:28] <epsilonomega13> Andrei, si mai putem aduga la ide  
[20:28] <epsilonomega13> Idee\*  
[20:28] <@Johane> Stocarea samba se poate adresa atat incepatorilor cat si avansatilor  
[20:29] <@Johane> Putem seta serverul in mod si chip  
[20:29] <@Johane> Putem cripta ,putem pune limita de acces local etc.  
[20:30] <@Johane> Sa va vorbesc putin si de ftp ca il folositi cu totii  
[20:30] == BGS\_ has changed nick to BGS  
[20:31] <@Johane> Ftp este folosit in serverele web pentru a da acces la diverse fisiere si setari din server  
[20:32] <@Johane> Se poate folosi ca inlocuitor la ssh pentru cei care vor sa ocoleasca consola din diverse motive  
[20:32] <@Johane> Foloseste portul 21 ( se poate schimba) pentru comunicare  
[20:32] <Birkoff> ftp -> file transfer protocol - un protocol des folosit pe internet pe langa http pentru a transfera fisiererele intre 2 locatii... nu are legatura cu accesul la diverse fisiere si setari din server asa cum spui  
[20:32] <epsilonomega13> Birkoff, ba poti avea acces  
[20:33] <s3v3n> +1 Birkoff  
[20:33] <epsilonomega13> Depinde cum este setat serverul respectiv  
[20:33] <Birkoff> accesul este dat de setarile serverului pentru fiecare user care acceseaza datele prin acel protocol  
[20:33] <epsilonomega13> Si ce drepturi are utilizatorul  
[20:33] <s3v3n> @epsilonomega13 principiul nu are treaba cu setarile  
[20:33] <s3v3n> este vorba despre un serviciu de transfer de fisiere  
[20:33] <@TheJudge> Johane te rog vorbeste de pe un singur nickname, mersi.  
[20:33] <@Nytro> Doar daca prin setari va referiti la permisiunile fisierelor  
[20:33] <s3v3n> nu e o exprimare ok pt chmo  
[20:33] <s3v3n> d  
[20:34] <@Johane> Pauza de intrebari  
[20:34] <@Johane> Sariti cu intrebariel si comentarii  
[20:34] <Birkoff> chmod -> change mode -> se refera la o comanda linux pentru a schimba drepturile de acces acordate unui fisier sau director  
[20:34] <@Johane> \*intrebarile  
[20:35] <s3v3n> ma refeream la permisiunile fisierelo

[20:35] <s3v3n> r  
[20:35] <@Nytro> Nu am intrebari, dar as vrea sa fac cateva precizari  
[20:35] <s3v3n> rwx  
[20:35] <@Johane> Nytro: te rog  
[20:35] <@Nytro> In primul rand FTP nu e un protocol sigur  
[20:35] <@Nytro> Si nu ma refer numai la transferul fisierelor care nu se face cryptat  
[20:35] <@Johane> Nytro: Nici nu am spus asta  
[20:35] <@Nytro> Ci si la logare, un pas important  
[20:35] <@Nytro> Datele sunt trimise in plain-text  
[20:35] <@Nytro> Adica parola este clara  
[20:36] <Birkoff> FTP over SSH, the practice of tunneling an FTP session over SSH, sometimes called Secure FTP (rarely used, because FTP's normal 2-channel nature makes such tunneling difficult)  
[20:36] <Birkoff> citat de pe wikipedia  
[20:36] <@Nytro> SFTP, alta mancare de psete  
[20:36] <s3v3n> SFTP != FTPS  
[20:36] <@Nytro> Si as mai preciza ca FTP foloseste 2 canale de comunicatie  
[20:36] <@Nytro> Da  
[20:36] <kNigHt> vrei sa spui ca intr-un STFP datele de logare se trimit plain-text?  
[20:36] <s3v3n> nu, se referea la ftp  
[20:36] <@Johane> Nytro: se poate folosi si scp pentru asta  
[20:37] <s3v3n> scp nu il poti folosi pe windows (din cate stiu eu )  
[20:37] <@Johane> s3v3n: nativ nu dar existsta programe  
[20:37] <@Nytro> Da, ce voiam sa subliniez cu cele 2 canale de comunicatie: unul pentru control si unul pentru date, este faptul ca se poate transfera un singur fisier la un moment dat  
[20:38] <@Nytro> Adica se creeaza 2 conexiuni intre clientul si serverul FTP  
[20:38] <@Nytro> Pe una circula comenzile, pe cealalta se realizeaza transferul fisierelor  
[20:38] <@Johane> Alte intrebari?  
[20:38] <Birkoff> revenind la subiectul acestei ore, transferul de date intre 2 sisteme se poate face in mai multe moduri, prin diverse protocoale, cele mai cunoscute fiind ftp, http, tcp fiecare cu avantajele si dezavantajele de rigoare  
[20:39] <Birkoff> o conexiune sigura intre 2 sisteme implica de obicei ssh  
[20:39] <@TheJudger> cum pot sa transfer fisiere de pe windows pe o partitie de linux ?  
[20:39] <Birkoff> ftp cel mai rapid  
[20:39] <@Nytro> Dirver pentru acel sistem de fisiere  
[20:40] <@Johane> TheJudger: ftp, samba share  
[20:40] <s3v3n> total commander avea plugin pentru partitii ext  
[20:40] <@Nytro> Pe acelasi calculator nu?  
[20:40] <@TheJudger> da  
[20:40] <@Johane> s3v3n: ext nu ext4  
[20:40] <s3v3n> ext in general. ext 2, ext 3. ext4 nu stiu exact  
[20:40] <@Johane> s3v3n: ext4 este diferit de 3  
[20:40] <@Nytro> Gasisem si folosisem un driver pentru ext3 pe Windows, dar am avut probleme cu el  
[20:41] <@Johane> ext3 este doar un ext2 cu jurnalizare adugata  
[20:41] <@TheJudger> deci se poata "monta" o partitie de linux ext4 pe windows ?  
[20:41] <s3v3n> @Johane: captain obvious  
[20:41] <Birkoff> intr-o retea de firma de exemplu, se pot transfera datele de la client la server prin protocolul tcp (network computers)  
[20:41] <@Johane> Birkoff: da se poate

[20:41] <Birkoff> serverul are share la directorul la care au acces clientii  
[20:42] <Birkoff> in general in ro se foloseste windows server cu exchange  
[20:42] <@Johane> Birkoff: fiecare client poate folosi numele propriu sau un nume comun la toti  
[20:42] <@TheJudger> johane, se poata "monta" o partitie de linux ext4 pe windows ?  
[20:42] <@Johane> Depinde de setarile serverului  
[20:42] <@Johane> TheJudger: pe moment nu  
[20:42] <Birkoff> in win 7 am auzit ca s-ar putea  
[20:42] <@Johane> Ext3 se poate dar nu 4  
[20:42] <@TheJudger> am inteles  
[20:42] <@Johane> Birkoff: nu se poate  
[20:43] <Birkoff> nu am testat :)  
[20:43] <@Johane> Am teste eu :)  
[20:43] <@Johane> testat\*  
[20:43] <s3v3n> tell us more  
[20:43] <@Johane> alte intrebari mai sunt?  
[20:43] <kNigHt> linux recunoaste partitii windows, poti folosi un cd bootabil pt transfer  
[20:44] <@Johane> kNigHt: da linux cunoaste partitii undows folsind driverul ntfs-3g  
[20:44] <@Nytro> Hmmm <http://www.diskinternals.com/linux-reader/>  
[20:44] <Birkoff> e posibil cred cu aplicatii externe care sa monteze acea partitie in asa fel incat sa fie inteleasa de sistemul windows... dar practic nu mi-a trebuit si nu stiu detalii  
[20:44] <s3v3n> sau eventual un stick bootabil. pt rapiditate  
[20:44] <+Cyborg> Title: Access to Ext 2/3/4, HFS and ReiserFS from Windows (at [www.diskinternals.com](http://www.diskinternals.com))  
[20:45] <@TheJudger> deci se poate  
[20:45] <@TheJudger> :)  
[20:45] <@Nytro> Nu e nevoie neaparat de ntfs-3g daca kernelul e compilat cu suport read-write pentru NTFS :)  
[20:45] <@Nytro> Da, asa se pare  
[20:46] <@TheJudger> da, tocmai l-am testat  
[20:46] <alex67x> o  
[20:46] <@TheJudger> si pot citi fisiere din windows7, de pe partitia ext4 pe care e ubuntu  
[20:47] <@Johane> TheJudger: Scrie clar :CITI nu si scrie  
[20:47] <@Johane> Poate doar citi ext3/4 dar nu si scrie  
[20:48] <@Johane> Dac anu mai sunt intrebari o sa trec la ultima parte si anume alternative linux la programe windows  
[20:48] <@TheJudger> da, de scrie nu pot scrie :(  
[20:48] <@TheJudger> \*scris  
[20:48] \* Andrei ultima parte a lui johane : <http://www.slideshare.net/RoCyberCon/alternative-linux-la-programe-windows>  
[20:48] <@Johane> \$slideshare\_page 1  
[20:48] <@Johane> \$slideshare\_page1  
[20:49] <@Johane> Noii veniti din lumea windows cauta alternative la programele folosite in windows  
[20:50] <@Johane> In urmatoarele randuri o sa spune catev alaternative gratuite la cele windows  
[20:50] <Birkoff> da, asta e unul din motivele la care am renuntat sa lucrez in linux de mai multe ori... nu gaseam alternative la programele pe care le folosesc...  
[20:50] <@Johane> O sa in cep cu IM ( Instant messaging)  
[20:50] <@Johane> O sa vorbesc la inceput de Yhoo!  
[20:51] <@Johane> Yahoo!  
[20:51] <@Johane> Se poate folosi pidgin foarte bine, dar transferul de fisiere nu merge, camera web

si sunetul nu merg

[20:51] <@Johane> Vestea buna este ca merge ambele dar separat  
[20:52] <@Johane> Kopete este un program de linux scris pentru KDe  
[20:52] <@Johane> KDE\*  
[20:52] <@Nytro> Transferul de fisiere functioneaza pe Pidgin @ Yahoo!  
[20:52] <@TheJudger> Nytro, mie nu-mi merge  
[20:52] <@Johane> Nytro: 1 da una ba  
[20:52] <bcman> mie nu imi merge  
[20:52] <s3v3n> @Nytro: adevarat. doar uneori mai are faze  
[20:52] <@TheJudger> de pe yahoo pe pidgin  
[20:52] <Birkoff> nu cu toti... depinde si de versiunea de yahoo care o are celalalt...  
[20:52] <@Johane> De[inde de situatie  
[20:52] <Spock> Scuze, nu urmaresc discutia. Dar la IRC puteai trece si irssi. :-)  
[20:53] <@Nytro> 2.7.3 mergea bine, aparent un Unknown la time remaining, dar functiona :)  
[20:53] <s3v3n> per total is crapy :P  
[20:53] <@Nytro> YMSG nu e un protocol open-source, sunt probleme...  
[20:53] <@Johane> Ei bine pe kopete din ce am testat eu merge intotdeauna transferul de fisiere  
[20:54] <@Johane> Am trimis la toti cei din lista mea un fisier de test si la toti a ajuns cu bine  
[20:54] <Birkoff> si invers?  
[20:54] <@Johane> Birkoff: invers nu am testat dar cred ca merge si invers  
[20:54] <Birkoff> ca eu invers am avut probleme, nu puteam primi  
[20:55] <@Johane> Kopete are o singura problema  
[20:55] <@Andrei> Transferul de fisiere pidgin - messenger si invers functioneaza perfect pe ultima versiune de pidgin, cel putin pe ubuntu. :)  
[20:55] <Birkoff> de fapt primesc dar nu de la toti, de multe ori imi da transfer failed  
[20:55] <@Johane> Aduce in spate multe dependente KDE  
[20:55] <s3v3n> multe fiind Qt ?  
[20:55] <Spock> Birkoff, use rapidshare  
[20:56] <@Johane> s3v3n: nu tot qt ci doar ce ii este lui strict necesar sa functioneze  
[20:56] <Spock> da s3v3n  
[20:56] <Spock> anumite librarii  
[20:56] <Birkoff> :) de obicei primesc coduri... folosesc codepad.org :D  
[20:56] <s3v3n> as recomanda pentru file sharing dropbox, sau un serviciu tare ge.tt  
[20:56] <Spock> la file sharing ati vorbit despre ftp clients? ca io am fost putin plecat  
[20:56] <Birkoff> spock da se poate spune ca sa vb si de ftp  
[20:57] <@Andrei> Spock, citeste te rog arhiva :)  
[20:57] <Spock> ok  
[20:57] <@Johane> Daca cu transfeul de fisiere am rezolvat cum face cu camera web si microfon?  
[20:57] <@Johane> Folosim gyachi  
[20:58] <s3v3n> @Johane, ai reusit sa-l testezi?  
[20:58] <@Johane> s3v3n: da si ba  
[20:58] <s3v3n> mie nu mi-a mers de nici o culoare  
[20:58] == ksyus [webchat@522FBF15.11F35438.8ADF52B3.IP] has left #rocybercon []  
[20:58] <@Johane> Merge cand vrea  
[20:58] <s3v3n> pe mai multe calculatoare  
[20:58] <@Johane> s3v3n: ultima versiune este din 2007 adica foarte veche  
[20:58] <s3v3n> si interfata e foarte prost organizata  
[20:59] <@Johane> s3v3n: este prost organizata pentru ca a fost abandonat destul de repede de dezvoltatori

[20:59] <@Johane> De ce nu stiu  
[21:00] <@Johane> Chiar daca protocolul de care voi vorbi acum nu este foarte folosit la noi in tara el este foarte folosit in Vest si USA  
[21:00] <@Johane> Este vorba de MSN  
[21:00] <@Andrei> incepe sa prinda si la noi MSN, incet dar sigur :)  
[21:01] <@TheJudger> de cand cu xss-urile in yahoo :P  
[21:01] <@Johane> Pentru msn exista mai multe programe dar cel mai bun dintre ele este amsn  
[21:01] <@Johane> Andrei: Asta pentru ca msn nu iti baga 1000 de reclame pe gat cu forta  
[21:01] <@Johane> TheJudger: cu xssurile nu comentez  
[21:02] <@Johane> amsn are o interfata asemenatoare cu cea live messenger  
[21:02] <necuratu> paprerea mea este ca e mult mai sigur gchatul de la google  
[21:02] <@Andrei> portabilitate?  
[21:02] <necuratu> si mult mai accesibil...  
[21:03] <s3v3n> kopete, empathy si pidgin sunt si ele alternative pentru msn.  
[21:03] <@Johane> Andrei: este portabil pe orice sistem de operare  
[21:03] <@Johane> Windows, linux, mac  
[21:04] <@TheJudger> am testat empathy cu un cont de facebook cu 5000 de friends si a crapat :)  
[21:04] <@TheJudger> (ca o paranteza la empathy)  
[21:04] <@Johane> TheJudger: nici nu ma mir :)  
[21:04] <s3v3n> nu sunt nici eu fan empathy.  
[21:04] <bcman> oricum, e greu sa ajungi la 500 friends  
[21:04] <bcman> 5000\*  
[21:05] <@Johane> empathy mai are inca bug-uri nedescoperite  
[21:05] <@TheJudger> bcman, in 2 luni :)  
[21:05] <s3v3n> si pidgin le are. anyway  
[21:05] <@Johane> Sa trecem acum la bine cunoscutul irc :)  
[21:06] <@Johane> cei care folosesc irc, au 3 optiuni  
[21:06] <@Johane> Celor de pe gtk+ le recomand xchat  
[21:07] <@Johane> Este foarte bine integrat si intelege de minune cu el  
[21:07] <@Johane> gtk+ aduca gnome si xfce  
[21:07] <s3v3n> si din nou pidgin, si de data asta si kopete suporta irc.  
[21:07] <@Johane> Pentru cei de pe KDE le recomand konversation  
[21:08] <@Johane> s3v3n: dar mie unu nu imi place nici pidgin si nici kopete pentru irc  
[21:08] <@Nytro> Nu conteaza desktop environmentul, toate distributiile au runtime-ul de la Qt si Gtk  
[21:08] <s3v3n> true  
[21:08] <@Johane> Nytro: toate le au doar nu toate le instaleaza  
[21:09] <@Johane> Si sa aduci jumate din qt doar pentru konversation nu e prea bine :)  
[21:09] <s3v3n> @Johane nu le instaleaza adica nu vin by default cu ele  
[21:09] <s3v3n> instalate  
[21:09] <s3v3n> si chiar nu e mare treaba daca instalezi qt  
[21:09] <s3v3n> oricum e o posibilitate mare ca vei mai folosi si alte programe ce vor folosi qt  
[21:10] <Birkoff> pentru cine lucreaza cu multi clienti din exterior, eu recomand pidgin, deoarece te poti conecta si la yahoo si la msn si la gtalk si la alte servere de mess iar pentru o discutie rapida cu clientul e de ajuns... insa daca vrei ceva gen skype... aici e o problema...  
[21:10] <@Johane> s3v3n: pai nu e dar ocupa destul de mult spatiu  
[21:10] <@Nytro> Pfff, ce distributie nu instaleaza runtime-ul de la Qt si Gtk?  
[21:10] <s3v3n> a vrut sa spuna ca nu vine cu ele instalate  
[21:10] <@Johane> Birkoff: este skype si pentru linux , dar este antic  
[21:10] <s3v3n> eu il folosesc, aproape zilnic



[21:11] <@Johane> Este pe la versiunea 5 skype normal si 2 pe linux  
[21:11] <@TheJudger> si cu buguri :(  
[21:11] <s3v3n> si e o solutie destul de buna (si chiar printre putinele) pentru voce si video  
[21:11] <necuratu> dar o solutie de terminal aveti..functionala ?  
[21:11] <@Johane> TheJudger: exista in orice software :)  
[21:11] <ZeroCold> Salut, am intaziat... :(( despre ce se discuta?  
[21:11] <s3v3n> ce fel de terminal ?  
[21:11] <@Johane> necuratu: irssi  
[21:11] <kNigHt> o mica paranteza, pe langa gimp mai e vreo alternativa a photoshopului pentru linux?  
[21:12] <s3v3n> eu nu cunosc  
[21:12] <kNigHt> sau illustrator/corel, ceva de editare vectoriala...  
[21:12] <necuratu> eu am cautat un ala momentan dar nu suporta protocolul 16 de laymgrs  
[21:12] <ZeroCold> Photoshop CS2 merge instalat cu wine...  
[21:12] <kNigHt> wine cred ca e cel mai buggy software de linux  
[21:12] <s3v3n> inkscape pt corel (?)  
[21:12] <@Johane> ZeroCold: am vazut acum si cs5 cu anumiti pasi la instalare  
[21:13] <kNigHt> mersi s3v3n  
[21:13] <ghitzZza> exista/folositi alt emulator de windows pe linux inafara de wine?  
[21:13] <ZeroCold> playonlinux?  
[21:13] <@Nytro> WINE = Wine Is Not an Emulator  
[21:13] <@Johane> ghitzZza: mai exista un emulator  
[21:13] <@TheJudger> decat sa te chinui sa faci un program de pe windows sa mearga pe linux sau sa cauti alternative care sunt de calitate mai proasta nu mai bine folosesti un sistem dual-boot ? ;;)  
[21:14] <@Nytro> E o implementare a API-ului Windows pe Linux  
[21:14] <s3v3n> sau folosesti programe cross-platform  
[21:14] <ZeroCold> sau instalezi windows in virtualbox :)  
[21:14] <s3v3n> din cate stiu eu, playonlinux nu ar fi bazat pe wine? ca si cedega de altfel ?  
[21:14] <@TheJudger> ZeroCold, da;)  
[21:14] <@Johane> s3v3n: play on linux este wine cu catev ascripturi in plus  
[21:15] <s3v3n> so, yes  
[21:15] <@Johane> Cedega este altceva daca tin minte bine  
[21:15] <bcman> si e si pe bani :|  
[21:15] <s3v3n> e pe wine  
[21:15] <Birkoff> parerea mea e ca pana o sa apara programe compatibile in linux cu cele de pe windows (si la fel de bune) o sa vina google os din urma si o sa puna la dispozitie o itnreaga gama de aplicatii cel putin la fel de bune ca cele de windows... mai asteptati 2-3 ani si vedem :D  
[21:15] <s3v3n> exista versiuni "gratuite" pe torrente  
[21:15] <@Johane> bcman: este pe bani dar poate rula mai multe decat wine  
[21:16] <s3v3n> personal nu am si din principiu nu folosesc wine & similar  
[21:16] <@Johane> s3v3n: nici eu  
[21:16] <cyder> Birkoff,.. discutia asta tine de mai mult de 10 ani  
[21:16] <cyder> :)  
[21:16] <@Johane> Intrebari?  
[21:16] <@Andrei> alte intrebari, daca nu Nytro asateapta sa intre :)  
[21:17] <Birkoff> concluzia zic eu e ca mai bine linux pe post de server si windows pe post de client  
[21:17] <Birkoff> cine lucreaza are tot ce ii trebuie in windos  
[21:17] <Birkoff> si pe server datorita securitatii ridicate de preferat linux  
[21:17] <s3v3n> pentru server, exclusiv linux. pentru client, mai exista alternativa mac

[21:17] <epsilonomega13> Birkoff, si cine stie are tot ce ii trebuie in linux :P  
[21:17] <@TheJudger> +1 Birkoff  
[21:17] <@Andrei> E discutabil, fiecare foloseste o gama de aplicatii care pot fi (sau nu) satisfacute :)  
[21:18] <cyder> eu credeam ca bsd e mai bun pe server decat linux  
[21:18] <tdxev> de acord cu birkoff  
[21:18] <s3v3n> e discutabil  
[21:18] <epsilonomega13> cyder, depinde de utilizator  
[21:18] <sysgh0st> freebsd da e mai securizat decat linux in general  
[21:19] <Birkoff> in cadrul unei firme, de exemplu ai nevoie de office si de un program de fac si eventual un program contabil  
[21:19] <@Andrei> cel mai bine e sa testezi ambele sisteme de operare si sa aflam daca ne plac sau nu. Le pastram pe ambele sau nu? :)  
[21:19] <Birkoff> astea le poti gasi in linux  
[21:19] <cyder> aici vorbim de mediu de productie serios  
[21:19] <cyder> nu retele cu 10-20 de utilizatori  
[21:19] <cyder> :)  
[21:19] <sysgh0st> rm -rf C:\windows\\*  
[21:19] <epsilonomega13> sysgh0st, :)  
[21:19] <s3v3n> +1  
[21:19] <Birkoff> dar pentru un utilizator casnic... fiecare cu preferintele lui... eu sunt mai pretenitos de fel si nu am gasit tot ce imi trebuie in linux asa ca am renuntat la el il folosesc doar pe post de server  
[21:20] <s3v3n> folosesc de aproape 2 ani doar linux  
[21:20] <s3v3n> si sunt multumit  
[21:20] <epsilonomega13> Birkoff, fiecare cu preferintele sale  
[21:20] <Birkoff> nu sa vb insa de care linux e mai ok  
[21:20] <RST> 'sup?  
[21:20] <Birkoff> dar probabil asta pe data viitoare  
[21:20] <sysgh0st> Birkoff nu vb despre oi (casnici) ... sau despre asta era vorba  
[21:20] <sysgh0st> vorba^  
[21:20] <cyder> ultimul server pe care l-am avut acasa la ai mei avea 133 mhz 4 gb hdd 64 mb ram si rula slackware  
[21:21] <Birkoff> era vb de programe de zi cu zi compatibile in linux  
[21:21] <cyder> si era pe post re router/firewall si o mica pagina web  
[21:21] <@Andrei> avand in vedere ca deja trecem pe windows vs linux la sange propun sa lasam dezbaterea asta pentru fondatorii acestora. Noi trebuie sa ne bucuram de ce au creat si sa trecem mai departe. :)

## OOP in PHP

[21:21] <Birkoff> sunt de acord cu andrei  
[21:21] <epsilonomega13> Sa trecem mai departe sa il ascultam pe Nytro  
[21:21] <Birkoff> Andrei \*  
[21:21] <epsilonomega13> Sunt foarte interesat de ce urmeaza  
[21:22] == epsilonomega13 has changed nick to Johane2  
[21:22] <@Nytro> Si eu :D  
[21:22] <RST> mcfloci zi ceva  
[21:22] <s3v3n> e o tema foarte actuala si foarte interesanta :)

[21:22] <Johane2> Deci Nytro ai onoarea :)

[21:22] \* Andrei Nu vom merge pe sincronizare, din nou. Nu vreau sa iesiti afara. De aceea prezentarea lui Nytro e aceasta. In urmatoarele minute ne va plictisi (sau incanta) cu PHP & OOP. :D

[21:22] <@Andrei> Succes

[21:22] <@Andrei> <http://www.slideshare.net/RoCyberCon/ooop-din-perspectiva-php-5>

[21:22] <@Nytro> .mode +m

[21:22] <@TheJudger> .mode +m

[21:22] == mode/#rocybercon [+m] by Guardian

[21:23] <@TheJudger> .op sysgh0st

[21:23] == mode/#rocybercon [+o sysgh0st] by Guardian

[21:23] <@Nytro> Bun, sa trecem la treaba

[21:23] <@Nytro> O sa disctuam despre programarea orientata pe obiecte in PHP

[21:23] <@Nytro> .mode -m

[21:23] <@Nytro> :-?

[21:23] <@TheJudger> .mode -m

[21:23] == mode/#rocybercon [-m] by Guardian

[21:23] <@Nytro> In fine, sa incep

[21:24] <@Nytro> Initial voiam sa folosesc diacritice, dar asta ar dubla timpul si nu vreau sa va plictisesc prea rau

[21:24] <@Nytro> Vreau sa incep cu cateva lucruri teoretice pe care le veti intelege mai bine pe parcurul prezentarii

[21:24] <Birkoff> a?tept?m cu ner?bdare prezentarea ta

[21:24] <@Nytro> Programarea orientata pe obiecte, este dupa cum spune teoria, o paradigma de programare

[21:25] <@Nytro> E cu alte cuvinte doar o modalitate de scriere a codului

[21:25] <@Nytro> Ideea de baza este simpla

[21:25] <@Nytro> Spre deosebire de programarea procedurala

[21:25] <@Nytro> La care datele si functiile erau amestecate si totul era o mare invalmaseala

[21:26] <@Nytro> PRogramarea orientata pe obiect vrea sa faca ordine intre aceste date si functii

[21:26] <@Nytro> Si se bazeaza pe gruparea acestor date in clase care definesc obiecte

[21:26] <@Nytro> Principiul de baza este separarea entitatilor cu care se lucreaza

[21:27] <@Nytro> Hmmm, sa ma gandesc la un exemplu

[21:27] <Birkoff> si evitarea repetitiilor\*

[21:27] <@Nytro> Da, re folosirea codului

[21:27] <@Nytro> In viata reala avem de-a face cu oameni, lucruri...

[21:27] <@Nytro> Putem considera aceste notiuni, in perspectiva POO ca niste obiecte

[21:28] <@Nytro> Astfel vom separa aceste notiuni

[21:28] <@Nytro> Si vom lucra separat cu ele

[21:28] <@Nytro> Nu am ales tocmai un exemplu potrivit, dar veti intelege pe parcurs ce vreau sa spun

[21:28] <@Nytro> Continuand cu putina teorie, la baza programarii orientate pe obiecte stau mai multe notiuni

[21:28] <Birkoff> cred ca cel mai bun exemplu e telefonul cu camera web si alte accesorii

[21:28] <@Nytro> :)

[21:29] <@Nytro> Sa incepem cu incapsularea datelor

[21:29] <@Nytro> Incapsularea, adica ascunderea datelor

[21:29] <@Nytro> Este un principiu extrem de important in POO, poate chiar cel mai important

[21:29] <@Nytro> Si anume, ca datele sa fie ascunse, accesul la aceste date sa se faca prin intermediul unei interfete definite de programator

[21:30] <@Nytro> Nu stiu cui anume sa ma adresez, nu as putea sa ma adresez celor care nu stiu mai deloc programare

[21:30] <@Nytro> O sa ma adresez celor care au cel putin notiuni de baza legate de programare

[21:30] <@Nytro> Chiar daca acestea nu sunt in PHP

[21:30] <@Nytro> Sper sa inteleaga toata lumea cate ceva

[21:30] <@Andrei> Stai linistit, subiectul e oricum mai tehnic. Important e sa dai idei. :D

[21:31] <@Nytro> Aceasta idee de ascundere la informatiilor este de baza din mai multe motive

[21:31] <@Nytro> In primul rand, la acele date pot avea acces mai multe persoane

[21:31] <Birkoff> cred ca aici te pot ajuta - cei incepatori abordeaza programarea procedurala, scriu si gandesc logica codului pe masura ce au nevoie de anumite evenimente si creaza si apeleaza functii pe masura ce au nevoie de acele date

[21:31] <@Nytro> Prin date, ca sa fie mai usor de inteles, ma refer la variabile, la anumite informatii

[21:32] <@Nytro> Datele sunt cele cu care se lucreaza prin intermediul functiilor

[21:32] <@Nytro> Si putem privi acest concept de ascundere a informatiei si ca o problema de securitate

[21:32] <Birkoff> pe cand in oop toata nebunia asta se separa, fiecare actiune e precis separata in cod, preluarea, prelucrarea datelor, verificarea lor si salvarea sau afisarea lor sunt grupate in actiuni separate

[21:32] <Birkoff> tac ?D

[21:32] <@Nytro> De exemplu, avem o variabila (termeni procedurali) si permitem accesul la ea

[21:33] <@Nytro> Ca in programarea procedurala, unde "toata lumea" are acces la date, deci toata lumea are acces la acea variabila

[21:33] <@Nytro> Prin intermediul functiilor de acces la variabile, deci prin prevenirea accesului direct la variabile, putem filtra datele pe care le contine acea variabila

[21:34] <@Nytro> Daca initial nu ati inteles mare lucru, nu va panicati, veti intelege dupace voi termina discutia teoretica

[21:34] <@Nytro> Birkoff, nu ma deranjeaza sa intervii, dimpotriva

[21:34] <Birkoff> prefer sa detaliez dupa ce termini, asa e frumos

[21:34] <@Nytro> Daca e ceva de mentionat, ceva important ce are legatura cu subiectul, nu voi fi deranjat

[21:35] <@Nytro> Sau daca omit sa specific ceva, daca gresesc ceva, sper sa fiu corectat

[21:35] <@Andrei> stai calm, eu astept greseala sa sar pe tine :D

[21:35] <@Nytro> Trecand mai departe, un alt principiu in programarea orientata pe obiecte este abstractizarea

[21:36] <@Nytro> Si anume, putem face anumite actiuni, ca sa ma exprim mai simplu, fara sa tinem cont de cum se realizeaza aceste atuni

[21:36] <@Nytro> Primul exemplu stupid care imi vine acum in minte este sa punem un animal sa vorbeasca de exemplu

[21:36] <@Nytro> Presupunem ca lucram cu animale

[21:36] <@Nytro> Ii putem cere unui animal sa "vorbeasca" fara sa tinem cont de cum face acest lucru

[21:36] <@Nytro> Fara sa ne intereseze cum isi foloseste corzile vocale

[21:37] <@Nytro> Cam asta ar fi ideea de baza

[21:37] <s3v3n> Aici am reprezentat schematic cum mi s-a prezentat mie ideea de encapsulare cand am inceput sa invat oop <http://www.dabbleboard.com/draw?b=Guest606262&i=0&c=31591d0601a0fd35f52f8abf01965733bca959be>

[21:37] <@Nytro> Si ar mai fi important ca mdoficarile ulterioare ale lucrurilor cu care lucram, sa nu afecteze implementarea

[21:37] <+Cyborg> Title: Dabbleboard - Online whiteboard for drawing & team collaboration - Interactive whiteboard software (at [www.dabbleboard.com](http://www.dabbleboard.com))

[21:37] <@Nytro> Adica daca acel animal va "vorbi" altfel, noi sa il putem pune sa vorbeasca in

acelasi mod

[21:38] <@Nytro> Imi cer scuze ca vin cu exemplu stupide, nu am pregatit exemple, dar sper sa intelegeti

[21:38] <@Nytro> Trecem mai departe

[21:38] <@Nytro> Un alt principiu extrem de important este mostenirea

[21:38] <@Nytro> Si anume, se pleaca de la niste lucruri generale

[21:39] <@Nytro> Sa iau un exemplu care mi-a fost prezentat la facultate

[21:39] <@Nytro> Scanunul

[21:39] <@Nytro> Putem defini un scaun, si anume putem specifica ca are 4 picioare, si ce altceva mai au in comun alte scaune

[21:39] <@Nytro> Apoi putem pleca de la acest concept si sa definim scaunul cu spatar

[21:39] <@Nytro> Acest scaun cu spatar are toate proprietatile scaunului definit de noi, si are in plus spatar

[21:40] <@Nytro> Noi, dupa ce am definit scaunul in general, ne putem folosi de el pentru a defini notiuni mai particulare

[21:40] <@Nytro> Fara sa fie necesar sa ii specificam toate proprietatile, le putem "importa" (mosteni) de la scaunul general

[21:41] <@Nytro> Un ultim concep de baza in POO ar fi polimorfismul

[21:41] <@Nytro> Polimorfism inseamna intr-o oarecare traducere: mai multe forme

[21:41] <@Nytro> In alte limbaje de programare este diferit fata de PHP

[21:41] <@Nytro> In alte limbaje prin polimorfism se permite executarea unor actiuni cu diferite date

[21:42] <@Nytro> Hmm, nu stiu daca pot gasi un exemplu concludent din viata reala care sa ilustreze acest concept

[21:42] <Birkoff> scaun de plastilina :D

[21:42] <@Nytro> :-?

[21:42] <@Nytro> Da, putem gandi si asa

[21:42] <@Nytro> Putem crea un scaun

[21:42] <@Nytro> Acesta poate fi de metal, de plastic, de lemn, sau de ce nu, de plastilina

[21:42] <@Nytro> Deci: mai multe forme

[21:43] <BGS> nu cred ca e corect spus mai multe forme mai degraba mai multe tipuri :D

[21:43] <@Nytro> Bun, cam astea ar fi principiile generale ale POO-ului

[21:43] <Birkoff> un obiec in mai multe forme si in mai multe tipuri

[21:43] <@Nytro> morfism = forma, cel putin in latina sau din ce limbaj provine

[21:44] <@Nytro> Si nu am ales degeaba "forma", nu e neaparat vorba de tipuri

[21:44] <@Andrei> BGS, defapt putem daca exemplu virusilor polimorfici - isi schimba forma dar fac acelasi lucru :)

[21:44] <@Nytro> Da, in alte limbaj ca C++ e vorba de tipuri

[21:44] <@Nytro> Deoarece polimorfismul se realizeaza prin supraincarcarea functiilor cu parametri diferiti

[21:44] <@Nytro> Eu vorbesc in termeni mai generali

[21:44] <Birkoff> comparativ... ai o clasa care lucreaza cu bazele de date... nu te intereseaza daca baza de date e mysql, mssql, sqlite sau oracle... pasezi si te folosesti de acea clasa care iti returneaza datele in forma care ai nevoie indiferent ce tip de sql e in spate

[21:44] <@Nytro> Corect

[21:45] <@Nytro> Bun, sa ne apropiem mai mult de programare

[21:45] <@Nytro> Si sa incercam sa definim notiunile cu care se lucreaza in programare si anume clasele si obiectele

[21:45] <@Nytro> O clasa reprezinta doar o definitie pentru o anumita notiune

[21:45] <@Nytro> Adica defineste acea notiune

[21:46] <@Nytro> Iar un obiect este o implementare a acelei notiuni  
[21:46] <@Nytro> Sa luam un exemplu  
[21:46] <@Nytro> hmmm  
[21:46] <Birkoff> imagineaza-ti o clasa ca o cutie de scule... si functiile din acea clasa ca fiind sculele efective  
[21:46] <@Nytro> Schita unei case, de exemplu, planul arhitectural ar putea fi o clasa  
[21:46] <@Nytro> Iar clasa propriu-zisa, construita dupa acea schita ar fi obiectiul  
[21:46] <@Nytro> Aceasta e ideea de baza  
[21:47] <@Nytro> Asa, sa ne legam acum de limbajul PHP  
[21:47] <@Nytro> Si sa vedem cum putem lucra cu aceste notiuni in PHP  
[21:47] <@Nytro> Datele cu care vom lucra sunt variabile  
[21:47] <@Nytro> Sa luam exemplul cu casa  
[21:47] <@Nytro> Ba nu, sa luam de exemplu un om  
[21:48] <@Nytro> Datele cu care am putea lucra ar putea fi: inaltime, greutate, varsta...  
[21:48] <@Nytro> In PHP vom lucra cu variabile ca: \$inaltime, \$greutate, \$varsta  
[21:48] <bluexaren> cum am facut eu Girl Generator  
[21:48] <@Nytro> Si actiunile pe care le poate face un om sa fie: mananca(), bea()...  
[21:48] <@Nytro> Acestea ar fi functii in PHP  
[21:49] <@Nytro> Si anume actiunile pe care le poate executa un om  
[21:49] <@Nytro> Legat de ce am povestit mai sus  
[21:49] <@Nytro> Si un animal ar putea avea aceste trasaturi  
[21:49] <@Nytro> Insa principiile OOP grupeaza aceste date  
[21:49] <@Nytro> Sa definim un om in PHP  
[21:49] <@Nytro> Va trebui deci sa definim o clasa  
[21:49] <@Nytro> Acea clasa va defini un om  
[21:50] <bluexaren> uitati aici un exemplu <http://pastebin.com/YiZ8cc2b>  
[21:50] <+Cyborg> Title: bluexaren - girl generator - Pastebin.com (at pastebin.com)  
[21:50] <@Nytro> Iar implementarile clasei, instantierile in termeni POO ar fi oameni propriu zis  
[21:50] <@Nytro> In PHP, o clasa se defineste folosind cuvantul cheie "class"  
[21:50] <@Nytro> Urmat de acolade in care sunt incadrate datele si functiile cu care se lucreaza  
[21:51] <@Nytro> Ex. class Om { }  
[21:51] <@Nytro> Iar intre acolade vom specifica datele si actiunile omului  
[21:51] <necuratu> <http://pastebin.com/53jnkB9R>  
[21:51] <+Cyborg> Title: class site { function bla() { } } - Pastebin.com (at pastebin.com)  
[21:51] <@Nytro> In termeni POO, datele se numesc proprietati (sau attribute), iar actiuniile care de fapt sunt functii se numesc metode  
[21:51] <@Nytro> De acum voi folosi acesti termeni  
[21:52] <@Nytro> Sa fac si o mica legatura cu PHP 4  
[21:52] <@Nytro> In PHP 4, o versiune mai veche de PHP, datele se defineau folosind cuvantul cheie "var"  
[21:52] <necuratu> in 5 se face cu define  
[21:52] <@Nytro> Adica pentru definirea datelor noastre am fi procedat astfel:  
[21:52] <@Nytro> Nu  
[21:52] <s3v3n> @necuratu: constantele  
[21:52] <necuratu> scz  
[21:52] <s3v3n> si nu in poo  
[21:52] <bluexaren> \$  
[21:53] <@Nytro> class Om { var \$inaltime, \$latime, \$varsta; }  
[21:53] <@Nytro> Asa am defini acele date pentru un om

[21:53] <@Nytro> Dar de la PHP 5 modelul obiectelor s-a schimbat foarte mult  
[21:53] <@Nytro> De fapt nici nu putem spune ca in PHP 4 am avut OOP  
[21:53] <@Nytro> Decat la un nivel foarte redus  
[21:53] <@Nytro> totul a fost rescris, imbunatatit si poate fi numit OOP  
[21:54] <@Nytro> In PHP 5, definirea datelor se face folosind un specificator de acces: public sau private  
[21:54] <@Nytro> Pentru inceput atat, nu am omis nimic  
[21:54] <Andrey> protected, internal...  
[21:54] <@Nytro> Da, voi specifica la mostenire  
[21:54] <necuratu> care sunt diferentele ?  
[21:54] <@Nytro> De protected  
[21:55] <@Nytro> Nu pot sa scriu 2000 de caractere pe secunda  
[21:55] <@Nytro> Asta incerc sa fac  
[21:55] <@Nytro> Deci, pentru inceput, putem defini datele ca public sau private  
[21:55] <s3v3n> Nytro: te descurci foarte bine :)  
[21:55] <@Nytro> Ex. class Om { public \$latime, \$inaltime; }  
[21:55] <@Nytro> Mersi  
[21:55] <@Nytro> Sau  
[21:56] <@Nytro> Ex. class Om { private \$latime, \$inaltime; }  
[21:56] <@Nytro> Este extrem de importanta diferente dintre aceste definitii  
[21:56] <@Nytro> Offtopic: nu pot raspunde la mesajele private  
[21:56] <@Nytro> Diferenta este de acces la aceste date  
[21:56] <@Nytro> Sa vedem cum se definesc si metodele si o sa specific diferenta  
[21:56] <bluexaren> uite  
[21:56] <bluexaren> <http://pastebin.com/bRan2Xgc>  
[21:57] <+Cyborg> Title: bluexaren // php OOP example - Pastebin.com (at pastebin.com)  
[21:57] <bluexaren> am facut eu exemplu  
[21:57] <@Nytro> Am specificat datele cu care se lucreaza  
[21:57] <@Nytro> Acum sa specificam si actiunile, adica metodele  
[21:57] <@Nytro> class Om  
[21:57] <@Nytro> P  
[21:57] <@Nytro> {  
[21:57] <@Nytro> private \$latime, \$inaltime;  
[21:57] <@Nytro> function mananca() { /\* mancare \*/ }  
[21:57] <@Nytro> }  
[21:58] <@Nytro> O metoda este o functie, deci se defineste ca si o functie  
[21:58] <@Nytro> De asemenea, ar trebui specificat un modificator de acces adica public sau private  
[21:58] <@Nytro> Si vom avea  
[21:58] <ZeroCold> private, adica se pot folosi doar in clasa respectiva?  
[21:58] <@Nytro> private function mananca() { /\* mancare \*/ }  
[21:58] <@Nytro> Ajung si la asta  
[21:58] <@Nytro> Sau  
[21:58] <@Nytro> public function mananca() { /\* mancare \*/ }  
[21:59] <@Nytro> Bun, am definit datele si metodele omului  
[21:59] <@Nytro> Acum sa si lucrăm cu ele  
[21:59] <@Nytro> Nu putem lucra asa cu clasa  
[21:59] <@Nytro> Va trebui sa cream un obiect plecand de la acea clasa  
[21:59] <@Nytro> Un obiect se creeaza folosind cuvantul cheie "new"  
[21:59] <@Nytro> Urmat de numele clasei

[21:59] <@Nytro> Adica:  
[22:00] <@Nytro> \$obiect = new Om;  
[22:00] <@Nytro> Sau  
[22:00] <@Nytro> \$obiect = new Om();  
[22:00] <@Nytro> Astfel, \$obiect este un ompropriu-zis  
[22:00] <@Nytro> Care poate sa manance, care are inaltime si greutate  
[22:00] <paxnwo> cei aici?  
[22:00] <@Nytro> Acum putem vorbi despre accesul la datele si metodele noastre  
[22:01] <@TheJudger> paxnwo taci si priveste  
[22:01] <@Nytro> Avem clasa Om, in care inaltimea si greutatea sunt publice  
[22:01] <@Nytro> Accesul la ele prin intermediul obiectului creat se face usor  
[22:01] <@Nytro> \$obiect -> inaltime  
[22:01] <@Nytro> Astfel avem acces la inaltime  
[22:01] <@Nytro> Pentru apelul unei metode, pentru a face omul sa manance:  
[22:01] <@Nytro> \$obiect -> mananca();  
[22:02] <@Nytro> Acum sa specific diferenta dintre public si private  
[22:02] <@Nytro> Daca datele noastre sunt publice  
[22:02] <Birkoff> o paralela... o functie dintr-o clasa se poate apela si fara instantare cu  
NumeClasa::nume\_functie(parametrii)  
[22:02] <s3v3n> static  
[22:02] <@Nytro> Da, mai tarziu asta  
[22:02] <@Nytro> Probabil mult mai tarziu  
[22:02] <@Nytro> Daca datele sunt publice  
[22:02] <@Nytro> Putem avea acces la acele date din obiectul proaspat creat  
[22:03] <@Nytro> De exemplu, putem seta varsta astfel  
[22:03] <@Nytro> \$obiect -> varsta = 19;  
[22:03] <@Nytro> Sau putem afisa varsta astfel:  
[22:03] <@Nytro> echo \$obiect -> varsta;  
[22:03] <@Nytro> Daca datele noastre sunt private, asa cum ar trebui sa fie  
[22:03] <@Nytro> class Om { private \$varsta; }  
[22:03] <bluexaren> nytro scuzama ca ma bag in seama dar exact ce explici tu este aici  
<http://pastebin.com/bRan2Xgc> poate se intelege mai bine din intregul cod  
[22:03] <+Cyborg> Title: bluexaren // php OOP example - Pastebin.com (at pastebin.com)  
[22:04] <@Nytro> \$obiect = new Om;  
[22:04] <@Andrei> bluexaren se intelege minunat :) shht.  
[22:04] <@Nytro> bluexaren, nu am timp sa ma uit la acele link-uri  
[22:04] <@Nytro> Incerc sa merg cat mai repede deoarece am o banuiala ca o sa mai dureze ceva timp  
[22:04] <@Nytro> Dar vreau sa se inteleaga  
[22:04] <@Nytro> Vreau sa inteleaga toata lumea  
[22:04] <@Nytro> Bun, am creat obiectul nostru  
[22:05] <@Nytro> Folosind codul de putin mai sus  
[22:05] == paxnwo [webchat@B0DBC4B5.171C72E0.F4214F71.IP] has left #rocybercon []  
[22:05] <@Nytro> Daca vom incerca sa accesam varsta  
[22:05] <@Nytro> \$obiect -> varsta  
[22:05] <@Nytro> Cand aceasta este privata, nu ni se va permite, si vom primi o eroare  
[22:05] <@Nytro> Probabil va intrebati de ce am face acest lucru  
[22:05] <@Nytro> De ce am defini varsta ca private  
[22:05] == hkj has changed nick to TinKode  
[22:06] <@Nytro> Am specificat mai sus, la teorie: incapsularea (ascunderea) datelor



[22:06] <@Nytro> Un bun mod de lucru (si asa se lucreaza in POO) este sa definim datele ca private

[22:06] <TinKode> test

[22:06] <@Nytro> Accesul la ele se poate realiza prin intermediul metodelor

[22:06] <@Nytro> Astfel, clasa noastra va fi definita astfel

[22:06] <@Nytro> class Om

[22:06] <@Nytro> {

[22:07] <@Nytro> private \$varsta;

[22:07] <@Nytro> public function arata\_varsta() { return \$this -> varsta; }

[22:07] <@Nytro> public function seteaza\_varsta(\$varsta\_noua) { \$this -> varsta = \$varsta\_nou; }

[22:07] <@Nytro> }

[22:08] <@Nytro> Nu va sperati de \$this

[22:08] <@Nytro> \$this este un pointer catre obiectul curent

[22:08] <@Nytro> Adica se refera la datele din obiectul curent

[22:08] <@Nytro> In interiorul clasei, astfel vom avea acces la datele noastre

[22:08] <@Nytro> Deci il vom folosi in metodele noastre pentru a accesa datele noastre

[22:09] <@Nytro> In interiorul clasei este permis accesul la date, chiar daca acestea sunt private

[22:09] <@Nytro> Daca sunt private, accesul nu este permis doar din obiectele instantiate

[22:09] <@Nytro> Astfel, am definit clasa noastra

[22:09] <@Nytro> Nu putem accesa \$this -> varsta

[22:10] <@Nytro> Dar pentru a citi varsta putem folosi metoda arata\_varsta()

[22:10] <@Nytro> De exemplu, sa setam o valoare pentru omul nostru

[22:10] <@Nytro> \$obiect = new Om;

[22:10] <@Nytro> \$obiect -> seteaza\_varsta(20);

[22:10] <@Nytro> Am setat varsta la 20

[22:10] <@Nytro> La apelul functiei, aceasta seteaza \$this -> varsta la 20, aceasta fiind variabila noastra

[22:11] <@Nytro> Iar pentru a o citi

[22:11] <@Nytro> echo \$obiect -> arata\_varsta();

[22:11] <@Nytro> Probabil acum urmeaza intrebarea: Dar de ce sa nu schimbam direct valoarea variabilei?

[22:11] <@Nytro> Cum spuneam la incapsularea datelor, daca s-ar putea face asta direct, mai multe persoane ar putea avea acces la acea variabila

[22:11] == TheJudge [the@judge.info] has left #rocybercon []

[22:12] <@Nytro> Si se pot seta valori stupide

[22:12] == mode/#rocybercon [+o TheJudge] by TheJudge

[22:12] <@Nytro> Si din motive de securitate

[22:12] <@Nytro> Adica putem filtra valorile care pot fi atribuite acelei variabil

[22:12] <@Nytro> Deci, putem defini functia noastra astfel

[22:13] <@Nytro> public function seteaza\_varsta(\$varsta\_noua) { if(\$varsta\_noua > 0 && \$varsta\_noua < 120) \$this -> varsta = \$varsta\_noua; }

[22:13] <@Nytro> Adica putem specifica de exemplu niste limite pentru varsta

[22:14] <@Nytro> Daca am permite accesul direct, s-ar putea specifica valoarea 2000, ceea ce nu este un lucru foarte normal

[22:14] <@Nytro> Si astfel, luand lucrurile pe ocolite, avem o stabilitate mai mare

[22:14] <@Nytro> Un control mult mai bun al datelor cu care lucram

[22:14] <@Nytro> Cam astea ar fi principiile cu datele si metodele

[22:14] <@Nytro> Ma dor mainile :))

[22:15] <@Nytro> Totusi sa trec mai departe

[22:15] <Birkoff> se vede ca nu esti programator care scrie 8-10 ore pe zi :D

[22:15] <s3v3n> pevedeam asta. maine vei avea febra musculara la degete! :D  
[22:15] <Johane2> te inteleg perfect Nytro  
[22:15] <@Nytro> :D  
[22:15] <@Nytro> Bun, sa trecem la constructori si la destructori  
[22:15] <@Nytro> POO ne permite apelarea unor functii speciale la crearea si stergerea unor obiecte  
[22:15] <@Nytro> Aceste functii speciale poarta denumirea de constructori si destructori  
[22:16] <@Nytro> Singurul lucru diferit (deocamdata) fata de metodele normale  
[22:16] <@Nytro> Este ca sunt apelate automat la crearea unui obiect  
[22:16] <@Nytro> Aceste metode, au un format special  
[22:17] <Birkoff> nu poti distruge o clasa si cu \$ceva = new Clasa; unset(\$ceva); ?  
[22:17] <@Nytro> In PHP 4, o functie era constructor, adica se apela automat la crearea unui obiect  
daca avea acelasi nume ca si clasa  
[22:17] <@Nytro> Ba da  
[22:17] <@Nytro> Sa luam un exemplu, atentie, in PHP4!!!  
[22:17] <@Nytro> class Om  
[22:17] <Andrey> nu se distruge, se elima doar poiteru :]  
[22:17] <@Nytro> Se sterge obiectul  
[22:18] <@Nytro> Se apeleaza destructorul  
[22:18] <Andrey> si obiectul se baga in colectorul de gunoaie  
[22:18] <@Nytro> Sa luam exemplu  
[22:18] <@Nytro> class Om  
[22:18] <@Nytro> {  
[22:18] <@Nytro> :)  
[22:18] <Andrey> unde asteapta sa fie sters.,  
[22:18] <Andrey> scz de interuperi.  
[22:19] <@Nytro> public function Om { echo 'Obiectul a fost creat'; }  
[22:19] <@Nytro> }  
[22:19] <@Nytro> Aceasta functie, va fi apelata automat la crearea unui obiect  
[22:19] <@Nytro> Adica  
[22:19] <@Nytro> \$obiect = new Om;  
[22:19] <@Nytro> Va apela acea metoda, va apela constructorul  
[22:19] <@Nytro> Deci la \$obiect = new Om; va fi afisat acel mesaj pe ecran  
[22:20] <@Nytro> Cat despre destructori in PHP 4, acestia nu prea exista. :)  
[22:20] <@TheJudger> .op sysgh0st  
[22:20] == mode/#rocybercon [+o sysgh0st] by Guardian  
[22:20] <@Nytro> Acum sa trecem in secolul 21, la PHP 5  
[22:20] <@Nytro> PHP 5, din motive de compatibilitate admite si acest tip de constructor  
[22:20] <@Nytro> Adica si in PHP 5 se va putea folosi o metoda cu acelasi nume pe post de  
constructor  
[22:20] <@Nytro> Si va fi apelata automat  
[22:21] <@Nytro> Insa PHP 5 vine cu un mod special de creare al constructorilor  
[22:21] <@Nytro> Si anume, un constructor trebuie sa aiba numele  
[22:21] <@Nytro> \_\_construct  
[22:21] <@Nytro> Astfel, pentru a crea un constructor pentru clasa Om  
[22:21] <@Nytro> class Om  
[22:21] <@Nytro> {  
[22:21] <@Nytro> public function \_\_construct() { echo 'Obiect'; }  
[22:21] <@Nytro> }  
[22:22] <s3v3n> Spock va crea o eroare la parsare :|

[22:22] <@Nytro> La instantierea obiectului, la crearea acestuia va fi apelat constructorul  
[22:22] <@Nytro> Ce eroare?  
[22:22] <s3v3n> offtopic. scz :P  
[22:22] <@Nytro> A :)  
[22:22] <@Nytro> La  
[22:22] <@Nytro> \$obiect = new Om;  
[22:23] <@Nytro> Va fi afisat acel mesaj  
[22:23] <@Nytro> Iar destructorul are numele special: \_\_destruct  
[22:23] <@Nytro> Si este apelat cand obiectul este distrus  
[22:23] <@Nytro> Un lucru important cu acesti constructori este ca pot avea parametri  
[22:24] <@Nytro> Si pot fi folositi la crearea obiectului  
[22:24] <@Nytro> public function \_\_construct(\$varsta\_noua) { \$this -> varsta = \$varsta\_noua; }  
[22:24] <@Nytro> Daca vom defini astfel constructorul, vom putea specifica varsta la crearea obiectului  
[22:24] <@Nytro> Si vom crea obiectul astfel:  
[22:25] <@Nytro> \$obiect = new Om (20);  
[22:25] <@Nytro> Parametrul 20, va fi varsta obiectului nostru  
[22:25] <@Nytro> Atentie, trebuie specificat acest parametru  
[22:25] <@Nytro> Sau trebuie definit ca parametru optional ( \$varsta\_noua = 20 ) pentru a putea fi folosita o instantiere ca  
[22:25] <@Nytro> \$obiect = new Om;  
[22:26] <@Nytro> Cam asta momentan despre obiecte si clase  
[22:26] <@Nytro> Vad ca dureaza cam mult, nu stiu cum sa fac sa explic mai pe larg si sa inteleaga toata lumea  
[22:27] <@Nytro> Trecem mai departe  
[22:27] == Billy [webchat@WIT-87228A36.eregie.pub.ro] has left #rocybercon []  
[22:27] <@Nytro> Slide-ul 4  
[22:27] <Johane2> Nytro, eu inteleg destul de bine  
[22:27] <@Nytro> <http://www.slideshare.net/RoCyberCon/oop-din-perspectiva-php-5>  
[22:27] <@Nytro> Pentru cei care nu mai au link-ul  
[22:27] <@Nytro> Trecem la mostenire  
[22:27] <@Nytro> Am definit clasa Om, care este o clasa destul de generala  
[22:28] <@Nytro> De la aceasta clasa putem defini clasa Barbat, sau clasa Femeie  
[22:28] <kNigHt> cand se apeleaza \_\_destruct(), la unset(\$clasa) ?  
[22:28] <@Nytro> Si barbati si femeile sunt oameni  
[22:28] <@Nytro> Deci putem pleca de la acea definitie pentru a defini ceva mai genral  
[22:28] <@Nytro> Astfel, Barbat-ul va avea toate proprietatile omului  
[22:28] <@Nytro> Da  
[22:29] <@Nytro> Sau la finalul executiei scriptului  
[22:29] <@Nytro> Adica, daca ai un print la destructor  
[22:29] <@Nytro> Mesajul va fi afisat la rulara scriptului  
[22:29] <kNigHt> aha :)  
[22:29] <@Nytro> Cred ca si la \$obiect = NULL se apeleaza  
[22:29] <@Nytro> Deoarece este distrusa referinta la acel obiect  
[22:29] <@Nytro> In PHP 4 obiectele erau tratate ca tipurile int, float...  
[22:29] <@Nytro> In PHP 5 obiectele sunt tratate ca referinte  
[22:30] <kNigHt> deci cand dau \$clasa = new Clasa, \$clasa e defapt un pointer?  
[22:30] <@Nytro> Sa trecem la mostenire  
[22:30] <s3v3n> o intrebare plz ?

[22:30] <@Nytro> O referinta  
[22:30] <@Nytro> Spune  
[22:30] <@Nytro> Adica \$ clasa nu e obiectul propriu-zis  
[22:30] <@Nytro> Ci o referinta catre acel obiect  
[22:30] <s3v3n> in ce conditii se face o copie sau o referinta la obiect?  
[22:30] <@Nytro> Adica?  
[22:30] <s3v3n> adica daca am un obiect \$ob  
[22:31] <s3v3n> si fac \$ob2 = \$ob  
[22:31] <@Andrei> s3v3n, depinde de versiune. In php 4 doar \$x =& ...  
[22:31] <@Nytro> Cred ca urma sa spun spre final si de asta  
[22:31] <s3v3n> se cloneaza sau se copiaza  
[22:31] <@Nytro> Da, in PHP 4 asa trebuia folosit  
[22:31] <@Andrei> e doar copie la referinta  
[22:31] <@Nytro> Adica \$ob = &new clasa;  
[22:31] <@Nytro> De clonare vom vorbi mai tarziu  
[22:31] <@Nytro> Legat de ce ai spus tu  
[22:31] <@Nytro> \$ob = new Clasa;  
[22:32] <@Nytro> \$ob2 = \$ob;  
[22:32] <@Nytro> \$ob2 nu va fi un nou obiect  
[22:32] <@Nytro> Asa cum \$ob e o referinta catre un obiect  
[22:32] <@Nytro> \$ob2 va fi referinta catre ACELASI obiect  
[22:32] <@Nytro> Deoarece i se atribuie valoarea lui \$ob care e el o referinta  
[22:32] <@Nytro> Astfel, daca avem  
[22:32] <@Nytro> \$ob = new Om(20);  
[22:32] <@Nytro> \$ob2 = \$ob;  
[22:33] <@Nytro> \$ob2 -> seteaza\_varsta(30);  
[22:33] <@Nytro> Atunci si \$ob -> arata\_varsta() va fi tot 30  
[22:33] <@Nytro> Deoarece sunt referinte  
[22:33] <s3v3n> ok, dar daca cum spui tu in php5 si int-urile sunt tot cu referinte tratate, atunci cand lucram cu variabile simple, nu lucreaza la fel.  
[22:33] <@Nytro> Adica doar niste nume pentru acelasi obiect  
[22:33] <cyder> eu nu inteleg nimic  
[22:33] <kNigHt> asta-i in php5 da?  
[22:33] <s3v3n> da  
[22:33] <@Nytro> Da  
[22:34] <@Nytro> Nu am spus ca int-urile sunt tratate ca referinte  
[22:34] <s3v3n> cyder, spune plz ce anume nu intelegi  
[22:34] <@Nytro> Am spus ca obiectele sunt tratate asa  
[22:34] <cyder> pai nimic.. ca nu sunt programator  
[22:34] <@Nytro> Nelamuriri?  
[22:34] <@Nytro> nu va fie rusine  
[22:34] <@Andrei> cyder atunci nu intram in discutii :D  
[22:34] <@Nytro> Ok, trec mai departe atunci  
[22:34] <@Andrei> good :D  
[22:35] <@Nytro> Trebuie sa definim un Barbat plecand de la un Om  
[22:35] <@Nytro> Orice Barbat e om, deci orice Barbat are varsta  
[22:35] <@Nytro> O sa iau doar acest exemplu, pentru a detalia mai usor  
[22:35] <@Nytro> Si orice barbat, fata de omul nostru are in plus...  
[22:35] <@Nytro> Singurul lucru care imi vine in cap e penis :))

[22:35] <cyder> si superioritate in fata femeilor  
[22:35] <@Nytro> Gata, sa fiu mai subtil, o sa spun barba  
[22:35] <cyder> :D  
[22:35] <@Nytro> :))  
[22:36] <@Nytro> Bun, ne vom referi la barba ca la o proprietate particulara omului  
[22:36] <@Nytro> Nu toti oamenii au barba  
[22:36] <@Nytro> Dar toti barbatii au barba  
[22:36] <@Nytro> Sa presupunem...  
[22:36] <@Nytro> Astfel, POO ne permite sa mostenim toate datele si metodele de care avem nevoie din clasa Om, in clasa Barbat  
[22:37] <@Nytro> Spunem ca a noastra clasa Barbat, extinde clasa Om si ii adauga o noua functionalitate  
[22:37] <@Nytro> Si anume: barba si... rasul acestetia sa spunem  
[22:37] <@Nytro> Pentru a prelua toate datele unei clase vom folosi cuvantul cheie "extends"  
[22:37] <@Nytro> Astfel, vom defini clasa barbat astfel:  
[22:37] <@Nytro> class Barbat extends Om  
[22:37] <@Nytro> {  
[22:38] <@Nytro> private \$barba;  
[22:38] <Andrey> public length\_sula;  
[22:38] <@Nytro> public function rade\_barba();  
[22:38] <@Nytro> }  
[22:38] <@Nytro> Acum sa discutam cateva lucruri de baza la mostenire  
[22:38] <@Nytro> Si anume ce se mosteneste si ce anume nu se mosteneste de la clasa de baza  
[22:39] <@Nytro> Daca datele din clasa de baza sunt publice, acestea vor fi mostenite  
[22:39] <@Nytro> Daca datele si metodele din clasa de baza, adica clasa Om, sunt private, nu vor fi mostenite  
[22:39] <@Nytro> Astfel, in exemplul nostru, nu vom mosteni \$varsta de la clasa Om  
[22:39] <@Nytro> Deoarece aceasta este privata  
[22:39] <@Nytro> Insa avem o posibilitate  
[22:40] <@Nytro> Si aici apare modifierul de protectie: protected  
[22:40] <@Nytro> Daca in clasa noastra definim niste date ca protected  
[22:40] <@Nytro> NU vom putea accesa acele date folosind un obiect instantiat  
[22:40] <@Nytro> Daca avem:  
[22:40] <@Nytro> protected \$varsta; in clasa noastra  
[22:41] <@Nytro> Si \$ob = new Om;  
[22:41] <@Nytro> Nu vom avea acces la \$ob -> varsta;  
[22:41] <@Nytro> Insa protected e folosit la mostenire  
[22:41] <@Nytro> De aceea l-a lasta pentru acum  
[22:41] <@Nytro> Desi din afara clasei nu este permis accesul  
[22:41] <@Nytro> Datele protected vor fi mostenite  
[22:42] <@Nytro> In POO, clasele de la care se pleaca, clasele mai generala poarta denumirea de clase parinte  
[22:42] <@Nytro> Iar clasele care mostenesc acele date poarta denumirea de clase copil  
[22:42] <@Nytro> Ca sa ne obisnuim cu acest notatii  
[22:42] <@Nytro> Astfel, daca in clasa paritne avem  
[22:42] <@Nytro> protected \$varsta;  
[22:42] <@Nytro> Aceasta va fi mostenita de clasa copil  
[22:43] <@Nytro> Si vom avea si in clasa copil \$varsta  
[22:43] <@Nytro> De asemenea, daca in clasa copil \$varsta este public

[22:43] <@Nytro> Parinte, pradon  
[22:43] <@Nytro> Aceasta va fi mostenita in clasa copil  
[22:43] <@Nytro> Metodele sunt publice, deci vom avea acces la ele  
[22:43] <@Nytro> Deci si clasa copil le vom avea  
[22:44] <@Nytro> Si prin mostenire, datele sunt mostenite asa cum au fost declarate in clasa parinte  
[22:44] <@Nytro> Daca avem o metoda publica in clasa parinte, aceasta va fi publica si in clasa copil  
[22:44] <@Nytro> Si cu datele la fel  
[22:44] <@Nytro> Dar cum spuneam la incapsularea datelor, desi le putem declara ca publice ca sa le avem si in clasa copil  
[22:44] \* Alex slaps Kappa around a bit with a large fishbot  
[22:45] <@Nytro> Din principiile incapsularii, nu vrem sa permitem accesul direct la ele  
[22:45] <@Nytro> Si le vom declara ca protected, pentru a fi mostenite, si pentru a nu fi permis accesul direct la ele  
[22:45] <@Nytro> Daca vi se pare greu, puteti sa nu folositi protected  
[22:45] <@Nytro> Si sa folositi public  
[22:45] <@Nytro> Dar cum spuneam mai sus, va recomand sa ganditi astfel, asa se lucreaza in general  
[22:46] <@Nytro> Astfel, vom avea clasa om  
[22:46] <@Nytro> class Om  
[22:46] <@Nytro> {  
[22:46] <@Nytro> protected \$varsta;  
[22:46] <@Nytro> public function arata\_varsta() { return \$this -> varsta; }  
[22:46] <@Nytro> public function seteaza\_varsta(\$x) { \$this -> varsta = \$x; }  
[22:46] <@Nytro> }  
[22:47] <birkof> Seara buna  
[22:47] <@Nytro> Iar clasa barbat, o vom declara extinzand clasa Om  
[22:47] <@Nytro> si anume:  
[22:47] <@Nytro> class Barbat  
[22:47] <@Nytro> class Barbat extends Om  
[22:47] <@Nytro> {  
[22:47] <@Nytro> protected \$barba;  
[22:47] <@Nytro> public function rade\_barba() {}  
[22:47] <@Nytro> }  
[22:48] <@Nytro> Si clasa Barbat va avea proprietatile clasei Om  
[22:48] <@Nytro> Doar daca clasa Om avea proprietati/metode private  
[22:48] <@Nytro> Acestea nu erau mostenite de clasa copil  
[22:48] <@Nytro> Astfel, daca vom crea un barbat propriu-zis  
[22:48] <@Nytro> \$b = new Barbat;  
[22:48] <@Nytro> Vom putea  
[22:49] <@Nytro> \$b -> rade\_barba();  
[22:49] <@Nytro> Dar si \$b -> seteaza\_varsta(20);  
[22:49] <@Nytro> Deoarece aceasta metoda a fost mostenita de la clasa Om  
[22:49] <@Nytro> Deci aceasta metoda este prezenta si in clasa Barbat  
[22:49] <@Nytro> Cam acestea ar fi ideile pe care se realizeaza mostenirea  
[22:50] <@Nytro> Cateva lucruri ar mai fi de spus si anume  
[22:50] <@Nytro> Pentru cei care au programat in C++, stiu ca la mostenirea unui clase  
[22:50] <@Nytro> Este apelat automat constructorul clasei respective  
[22:51] <@Nytro> In PHP nu se intampla asa, va trebui apelat manual  
[22:51] <@Nytro> La fel ca si in C++, in PHP nu se permite mostenirea multipla  
[22:51] <@Nytro> Adica nu putem mosteni decat o singura clasa

[22:51] <@Nytro> Un alt lucru de remarcat fata de C++

[22:51] <@Nytro> Este ca din moment ce constructorii oricarei clase au numele `__construct` acesta se mosteneste

[22:52] <@Nytro> Adica, daca in clasa Om avem un `__construct` care afiseaza un mesaj

[22:52] <@Nytro> Chiar daca in clasa Barbat nu definim un constructor

[22:52] <@Nytro> Daca nu il definim

[22:52] <@Nytro> Se va mosteni constructorul din clasa Om, clasa parinte

[22:53] <@Nytro> Si la `$b = new Barbat;` va fi apelat acel constructor si se va afisa acel mesaj

[22:53] <@Nytro> De asemenea, in PHP 4, datorita numelor metodelor se putea ajunge la probleme

[22:53] <@Nytro> Si anume, daca aveam clasa A, in care defineam o metoda B()

[22:53] <@Nytro> Iar apoi defineam o clasa B care mostenea o clasa A

[22:54] <@Nytro> Se mostenea si metoda B si juca rol de constructor

[22:54] <@Nytro> Diferente fata de C++ ar mai fi ca in PHP putem apela constructorul de cate ori vrem

[22:54] <@Nytro> In C++ (parca am incercat) nu putem

[22:54] <@Nytro> Adica in PHP putem scrie de cate ori vrem

[22:54] <@Nytro> `$b -> __construct()`

[22:55] <@Nytro> Tot legat de mostenire, un lucru extrem de important e suprascrierea metodelor

[22:55] <@Nytro> In clasa barbat am mostenit metoda `arata_varsta()`

[22:55] <@Nytro> Dar o putem suprascrie sa afiseze altceva, sau sa lucreze altfel

[22:55] <@Nytro> Dar PHP impune anumite limitari la ceste suprascrieri

[22:55] <@Nytro> Prototipul functiei trebuie sa fie acelasi

[22:56] <@Nytro> Deci va trebui sa definim functia la fel, fara parametri

[22:56] <@Nytro> Singurul caz in care nu se aplica aceste limitari sunt constructorii

[22:56] <@Nytro> Constructorii ii putem scrie in ce modalitate vrem noi, cu cati parametri vrem

[22:56] <@Nytro> Bun, dar daca suprascriem o functie, cum putem apela functia din clasa parinte?

[22:57] <@Nytro> Putem face asta, in PHP exista niste clase speciale: parent, self, static

[22:57] <@Nytro> Ne vom referi doar la parent si la self

[22:57] <@Nytro> In interiorul unei clase, parent reprezinta clasa parinte

[22:57] <@Nytro> Iar self clasa curenta

[22:57] <@Nytro> Bine, legaturile sunt facute in mod static, dar vom ajunge si la asta

[22:57] <@Nytro> Pentru acces la datele din clasa parinte

[22:58] <@Nytro> Vom folosi "parente" si operatorul de rezolutie "::`"`

[22:58] <@Nytro> Astfel, putem apela metoda `afiseaza_varsta()` din clasa parinte

[22:58] <@Nytro> In caz ca in clasa noastra copil am suprascris-o

[22:58] <@Nytro> Folosind:

[22:59] <@Nytro> `parent::afiseaza_varsta();`

[22:59] <@Nytro> Iar self face referire la clasa curenta

[22:59] <@Nytro> Voi reveni cu detalii cand voi spune despre datele statice

[22:59] <@Nytro> Ce ar mai fi important la mostenire este prevenirea

[23:00] <@Nytro> Atat prevenirea suprascrierii functiilor, cat si prevenirea mostenirii clasei

[23:00] <@Nytro> Daca vrem o clasa care sa nu poata fi mostenita, o definim folosind cuvantul cheie "final"

[23:00] <@Nytro> `final class Barbat {}`

[23:00] <@Nytro> Astfel clasa noastra Barbat nu va mai putea fi mostenita

[23:00] <@Nytro> Iar o metoda, daca vrem sa nu poata fi redefinita in clasa copil

[23:00] <@Nytro> O vom defini, la fel, ca final

[23:01] <@Nytro> `final public function ZZZ() {}`

[23:01] <@Nytro> Si va fi mostenita, dar nu va putea fi suprascrisa

[23:01] <@Nytro> Am obosit :D  
[23:01] <s3v3n> ia o pauza! plz ! )  
[23:01] <@Andrei> good, pauza de cafea. :D 5 minute? :D  
[23:01] <@Nytro> Cam asta legat de mostenire  
[23:01] <@Nytro> V-ati plictisit?  
[23:02] <@Nytro> Cred ca e bine sa luam o pauza.  
[23:02] <s3v3n> no, e great :)  
[23:02] <TinKode> da da, continua  
[23:02] <TinKode> :))  
[23:02] <s3v3n> a spus TinKode :))  
[23:02] <Andrey> @Andrey pe cine minti e ora 11 tu acum fumezi vreo 6 tigari nu bei cafea :]  
[23:02] <tdxev> ..dap..  
[23:02] <@Andrei> Andrey nu. S-a terminat :D  
[23:02] <@Andrei> :)  
[23:02] <s3v3n> apropo Andrey  
[23:02] <@Andrei> not anymore ;)  
[23:03] <tdxev> la scoala pentru a capta atentia elevilor..  
[23:03] <s3v3n> sincer sa fiu, imi pare rau pentru tine daca ai \$length\_sula public :]  
[23:03] <dragon> as fi bucuros daca am discuta si de functii pt manipularea erorilor  
[23:03] <tdxev> profesorii pun intrebari  
[23:03] <tdxev> so :))  
[23:03] <tdxev> ..  
[23:03] <@Andrei> dragon, e cu totul alt subiect :D  
[23:03] <@Andrei> tdxev maxim :))  
[23:03] <TinKode> app, despre ce a fost vb in toate randurile de mai sus?  
[23:03] <bluexaren> PHP5 / PHP4 OOP  
[23:04] <@Andrei> Tinkode esti un cretin :))  
[23:04] <tdxev> tinkode treci la tabla:))  
[23:04] <TinKode> :))  
[23:04] <tdxev> zii despre ce am vorbit noi azi :))  
[23:04] <TinKode> nu am fost la ora  
[23:04] <TinKode> :))  
[23:04] <bluexaren> andrei , o conferinta about sqli o sa fie ?  
[23:05] <@Andrei> va fi, dar nu cu tinkode, daca asta era a doua intrebare :))  
[23:05] <TinKode> :)))  
[23:05] <bluexaren> nuu  
[23:05] <TinKode> iti dai seama ce ar iesi?:))  
[23:05] <bluexaren> mie tot una  
[23:05] <@Andrei> nu vreau sa-mi pun imaginatia la contributie :))  
[23:05] <bluexaren> ce ar iesi ?  
[23:05] <@TheJudger> pe tinkode il dam exemplu : "asa sa nu faceti acasa !"  
[23:05] <@Andrei> exact :))  
[23:06] <bluexaren> \$eu = new om();  
[23:06] <bluexaren> \$eu->merge\_la\_buda();  
[23:06] <@Andrei> :))  
[23:06] <BGS> eu za new om merge la flush dns :))  
[23:06] <s3v3n> @bluexaren: foloseste \$this  
[23:07] <bluexaren> de ce sa folosesc this ?  
[23:07] <bluexaren> daca is inafara clasei



[23:07] <bluexaren> apelez din alt fisier  
[23:07] <bluexaren> clasa aceea  
[23:07] <s3v3n> tu esti obiect de tip om. asta conteaza  
[23:09] <@Andrei> cat mai avem timp, ceva propuneri pentru urmatoarele editii? :D  
[23:09] <BGS> a cazut judecatoru sariti  
[23:09] <bluexaren> SQLi  
[23:09] <@Andrei> am retinut, altceva?  
[23:09] <TinKode> SQLi e imens  
[23:09] <TinKode> poate MySQLi  
[23:09] <BGS> Andrei maine cand incep inscrierile sa imi incerc si eu norocul  
[23:09] <bluexaren> Header Injection  
[23:09] <Birkoff> diferentele dintre browsere, html5  
[23:09] <@Andrei> bai hackere, ai sa ajungi la parnaie  
[23:10] <bluexaren> cine ?  
[23:10] <s3v3n> ce interesata e lumea de hacking...  
[23:10] <bluexaren> eu ?  
[23:10] <@Andrei> da, tu bluexaren  
[23:10] <BGS> :))  
[23:10] <@Andrei> Birkoff am notat  
[23:10] <bluexaren> n-o sa ajung  
[23:10] <bluexaren> folosesc prezervativ  
[23:10] <BGS> future blackboy1337 :D  
[23:10] <TinKode> Sa nu faci prostii sa intri pe nu stiu unde  
[23:10] <TinKode> E rau  
[23:10] <BGS> vezi ca prezervativele se mai si rup :D  
[23:10] <TinKode> Nu e ingaduit de la dumnezeu sa penetrezi servere  
[23:10] <bluexaren> nu se rup  
[23:10] <bluexaren> ca nu se misca cablu in router  
[23:10] <bluexaren> :))))  
[23:10] <BGS> la mine in unitate da :))  
[23:10] <BGS> de aia mai cad din cand in cand la datorie  
[23:11] <@Nytro> Reincepem?  
[23:11] <bluexaren> desigur  
[23:11] <@Andrei> cool :D  
[23:11] <BGS> mostenirea?  
[23:11] <bluexaren> da, sa vad cemi iese mie  
[23:11] <@Nytro> Ce mostenire?  
[23:11] <@Andrei> nu, abstractizare :)  
[23:11] <bluexaren> sa stiu ce am de vandut  
[23:11] <bluexaren> :))))  
[23:11] <@Andrei> bluexaron shht  
[23:11] <bluexaren> ok  
[23:11] <BGS> Nytro: imi este greu sa urmaresc nu am fost atent de la inceput dar continua imi voi da silinta :D  
[23:11] <@Nytro> Mai intai precizez cateva detalii apoi trec la abstractizare  
[23:12] <@Nytro> Bun, reincep  
[23:12] <@Nytro> Vreau sa reincep cu cateva lucruri pe care am uitat sa le precizez  
[23:12] <@Nytro> Proprietatile pot fi initialziate  
[23:12] <@Nytro> Adica \$varsta = 20;

[23:12] <@Nytro> Insa initializarea se poate face doar cu date constante, nu si cu expresii

[23:13] <@Nytro> Adica nu putem apela o functie, sa sa concatenem un sir sau mai stiu eu ce

[23:13] <@Nytro> Legat de var, de la PHP 5.00

[23:13] <@Nytro> Pana la PHP 5.1.x daca se folosea se primea un avvertisemnt

[23:13] <@Nytro> Un avvertisment mai special E\_STRICT

[23:14] <@Nytro> Dar apoi, var a devenit un ales pentru public

[23:14] <@Nytro> Si se poate folosi "var" ca si "public" fiind de fapt acelasi lucru

[23:14] <@Nytro> Sa trecem la urmatorul capitol

[23:14] <@Nytro> Si anume la abstractizare

[23:15] <@Nytro> Uneori, avem nevoie de niste prototipuri

[23:15] <@Nytro> Vream ca o anumita implementare sa urmeze anumite "standarde"

[23:15] <@Nytro> Spre exemplu, vrem ca o clasa sa implementeze automat anumite facilitati

[23:16] <@Nytro> Vrem ca o clasa sa implementeze neaparat o anumita metoda de exemplu

[23:16] <@Nytro> Legat de clasa Om, sa spunem ca vrem neaparat ca o clasa care o mosteneste sa implementeze o actiune numita...

[23:16] <@Nytro> Sa vedem ce ar putea fi

[23:16] <@Nytro> Nu stiu, sa zicem alearga()

[23:16] <@Nytro> Adica sa fortam clasele care extind clasa Om sa implementeze aceasta metoda

[23:17] <@Nytro> Pentru a face acest lucru, definim acea metoda folosind cuvantul cheie "abstract"

[23:17] <@Nytro> Vom avea in clasa Om

[23:17] <@Nytro> abstract public fugi()

[23:17] <@Nytro> Iar clasele care vor mosteni clasa Om, vor trebui sa implementeze neaparat aceasta metoda

[23:18] <@Nytro> Insa noi, in clasa Om nu trebuie sa o definim

[23:18] <@Nytro> Din moment ce urmeaza sa fie definita in clasa copil

[23:18] <@Nytro> Deci nu va trebui decat sa ii specificam prototipul

[23:18] <@Nytro> Si vom avea in clasa noastra Om

[23:18] <@Nytro> abstract public fugi();

[23:18] <@Nytro> Fara sa specificam nici un corp pentru metoda, doar definitia, prototipul

[23:19] <@Nytro> De fapt nici nu putem defini un corp pentru o metoda abstracta, vom primi un mesaj de eroare

[23:19] <@Nytro> Dar, de indata ca am definit o metoda abstracta intr-o clasa

[23:19] <@Nytro> Inseamna ca si acea clasa este abstracta

[23:19] <@Nytro> Si va trebui sa o definim ca abstracta

[23:19] <@Nytro> abstract class Om {

[23:20] <@Nytro> Iar clasa noastra Barbat, care mosteneste clasa Om

[23:20] <@Nytro> Trebuie neaparat sa implementeze aceasta metoda, daca nu va rezulta o eroare

[23:20] <@Nytro> abstract class Om

[23:20] <@Nytro> {

[23:20] <@Nytro> abstract public function fugi();

[23:20] <@Nytro> }

[23:20] <@Nytro> Si o vom mosteni in clasa Barbat

[23:20] <@Nytro> Class Barbat extends Om

[23:20] <@Nytro> {

[23:21] <@Nytro> public function fugi() { /\* alearga \*/ }

[23:21] <@Nytro> }

[23:21] <@Nytro> Trebuie sa facem acest lucru obligatoriu

[23:21] <@Nytro> Un lucru important legat de mostenirea acestor metode abstracte

[23:22] <@Nytro> Este modul in care trebuie ele definite in clasa copil

[23:22] <@Nytro> Prototipul functiei trebuie sa fie acelasi

[23:22] <@Nytro> Chiar si daca avem parametri optionali, metoda care trebuie implementata va trebui sa aiba parametri optionali

[23:23] <@Nytro> Si, din cate imi aduc aminte, pana la PHP 5.3 va trebui ca valorile parametrilor optionali sa fie aceleasi

[23:23] <@Nytro> Ca in metoda abstracta

[23:23] <@Nytro> Extrem de important este ca o clasa abstracta nu se poate instantia

[23:23] <@Nytro> Adica nu putem crea un obiect de la o clasa abstracta

[23:24] <@Nytro> Ar fi anormal, din moment ce o clasa abstracta are metode abstracte, deci metode care nu sunt definite

[23:24] <@Nytro> Vin apoi interfetele

[23:24] <@Nytro> Interfetele, la fel ca si clasele abstracte definesc modul in care trebuie implementata o clasa

[23:24] <@Nytro> Adica definesc metodele pe care trebuie sa le aiba o clasa

[23:25] <@Nytro> Pentru definirea unei interfete se foloseste cuvantul cheie "interface"

[23:25] <@Nytro> interface Om { }

[23:25] <@Nytro> Apoi putem specifica ce anume trebuie implementat de catre clasele ce vor implementa acea interfata

[23:25] <@Nytro> interface Om

[23:25] <@Nytro> {

[23:25] <@Nytro> public function fugi();

[23:25] <@Nytro> }

[23:25] == mode/#rocybercon [+o TheJudger] by TheJudger

[23:26] <@Nytro> In clasa pe care urmeaza sa o definim, va trebui sa declaram acea metoda

[23:26] <@Nytro> Pentru a implementa acea interfata in clasa noastra folosim cuvantul cheie "implements"

[23:26] <@Nytro> Si vom defini clasa astfel

[23:26] <@Nytro> class Barbat implements Om

[23:26] <@Nytro> {

[23:26] <@Nytro> public function fugi() { /\* Aciune... \*/ }

[23:26] <@Nytro> }

[23:27] <@Nytro> spre deosebire de mostenire

[23:27] <@Nytro> La implementarea unor interfete putem specifica mai multe interfete

[23:27] <@Nytro> Adica putem implementa mai multe interfete

[23:27] <@Nytro> class Barbat implements Fiinta, Om { }

[23:28] <@Nytro> Legat de specificatorul de acces din interfata

[23:28] <@Nytro> Adica daca definim o metoda in interfata (sau in clasa abstracta) ca public, private sau protected

[23:28] <@Nytro> In clasa care implementeaza interfata (sau mosteneste clasa abstracta)

[23:29] <@Nytro> Vizibilitatea cu care definim metoda trebuie sa fie >= (mai vizibila sau la fel) cu cea din definitia initiala, din care trebuie sa implementam

[23:29] <@Nytro> Adica, daca in interfata definim o metoda protected, in clasa care o implementeaza va trebui sa o definim ca protected sau public, deci nu ca private

[23:29] <@Nytro> Deoarece private este mai putin vizibila

[23:30] <@Nytro> Alt lucru important pentru interfete este ca ele se pot mosteni una pe alta

[23:30] <@Nytro> Adica putem extinde interfata Om de la interfata Fiinta

[23:30] <@Nytro> Si vom face acest lucru astfel

[23:30] <@Nytro> interface Om extends Fiinta { }

[23:31] <@Nytro> Iar clasa care va implementa interfata Om va trebui sa implementeze metodele

definite in ambele interfete

[23:31] <@Nytro> Cred ca acestea ar fi de spus despre clase abstracte si interfete

[23:31] <@Nytro> Folosirea lor ar fi urmatoarea: la inceputul unui proiect, se specifica un sablon care va fi folosit

[23:32] <@Nytro> Adica se poate sti de la inceput cu ce metode se va lucra

[23:32] <@Nytro> Si vei sti de la inceput ca vei avea metoda bea() cu un singur parametru: cati litri de bere

[23:32] <@Nytro> Ceva nelamuriri aici?

[23:32] <@Nytro> MA simt ciudat sa vorbesc asa singur :))

[23:32] <@Andrei> nu, e ok :)

[23:32] <@Nytro> Noroc cu robotelul care ma ajuta moral

[23:32] <@Andrei> :)))

[23:32] <tdxev> :)

[23:33] <@Nytro> Ar mai fi de mentionat un lucru important, si anume datele statice

[23:33] <@Nytro> Uneori poate avem nevoie sa folosim elemente dintr-o clasa fara sa fie nevoie sa o instantiem

[23:33] <@Nytro> Aici putem defini date statice

[23:33] <@Nytro> De exemplu, niste date

[23:34] <@Nytro> Pentru a le defini ca statice, deci pentru a putea avea acces la ele fara sa cream un obiect

[23:34] <@Nytro> Vom folosi cuvantul cheie "static"

[23:34] <@Nytro> class Om

[23:34] <@Nytro> {

[23:35] <@Nytro> static public \$definitie = "Om";

[23:35] <@Nytro> }

[23:35] <@Nytro> Putem avea acces la \$definitie fara sa cream un obiect Om, folosind operatorul de rezolutie

[23:35] <@Nytro> echo Om::\$definitie;

[23:35] <@Nytro> Folosim numele clasei, urmat de variabila statica

[23:36] <@Nytro> De asemenea putem defini si metode statice

[23:36] <@Nytro> Pentru a putea fi folosite fara instantiere

[23:36] <@Nytro> class Om

[23:36] <@Nytro> {

[23:36] <@Nytro> static public \$definitie = "Om";

[23:37] <@Nytro> public static function afiseaza() { echo self::\$definitie; }

[23:37] <@Nytro> }

[23:37] <@Nytro> Sincer, nu stiu daca e vreo diferenta intre public static si static public

[23:37] <@Nytro> Dar nu ar trebui sa conteze ordinea

[23:37] <@Nytro> Metoda afiseaza o vom apela tot static, folosind operatorul de rezolutie si numele clasei

[23:37] <@Nytro> Om::afiseaza();

[23:37] <@Nytro> Probabil ati observat ca am folosit self in interiorul metodei

[23:38] <@Nytro> Self indica clasa curenta si se foloseste in mod static, deci astfel vom avea acces la date statice

[23:38] <@Nytro> Important este faptul ca in metodele statice nu avem pointerul \$this

[23:38] <@Nytro> Ceea ce este foarte normal, din moment ce nu exista un obiect instantiat

[23:39] <@Nytro> Cam asta ar fi de spus pentru moment

[23:39] <tdxev> ty:)

[23:39] <@Andrei> de remarcat e ca se foloseste foarte des :D

[23:39] <@Nytro> Sunt si lucruri mai complicate, la unele vom ajunge  
[23:39] <@Nytro> Da :)  
[23:39] <@Nytro> Voi reveni putin asupra acestui subiect spre final  
[23:40] <@Nytro> Intrebari?  
[23:40] <BGS> prea socat sa pun intrebari ...  
[23:40] <@Andrei> :))  
[23:40] <@Nytro> :))  
[23:40] <@TheJudger> fetele, aveti ceva nelamuriri ?  
[23:40] <BGS> xD  
[23:40] <@TheJudger> bun, se pare ca ati inteles tot  
[23:40] <@TheJudger> ne bucuram  
[23:40] <@TheJudger> :))  
[23:41] <BGS> clar Nytro for president!  
[23:41] <BGS> :)  
[23:41] <@Nytro> Bun, trec mai departe atunci  
[23:41] <@Nytro> :))  
[23:41] <BGS> omg :/  
[23:41] <@Nytro> Trecem la slide-ul 7  
[23:41] <BGS> pana la 12 juma termini:P?  
[23:41] <@Nytro> La polimorfism si functii magice  
[23:41] <@Nytro> :))  
[23:41] <tdxev> n2 sec..  
[23:41] <TinKode> tocilare :)))  
[23:41] <@Nytro> :->  
[23:41] <tdxev> ce slide ca la mine nu a aparut nimic :|  
[23:41] <@TheJudger> .ban TinKode  
[23:42] <@Nytro> Dati si voi la slie-ul 7 manual  
[23:42] <@Nytro> <http://www.slideshare.net/RoCyberCon/oop-din-perspectiva-php-5>  
[23:42] <tdxev> asa ms  
[23:42] == mode/#rocybercon [+b \*!\*@6E553F0F.F0FC5B65.96EEA9AC.IP] by TheJudger  
[23:42] <@Nytro> Npc  
[23:42] <BGS> Nytro esti o enciclopedie umana :P  
[23:42] <@Nytro> Bun, trecem la polimorfism  
[23:42] <@Nytro> :))  
[23:42] <@Nytro> Si la asa-zisul overloading in PHP  
[23:42] <@Nytro> Cum spuneam si la inceput, in PHP nu avem polimorfismul din alte limbaje de programare  
[23:43] <@Nytro> In C++ si in alte limbaje, polimorfismul inseamna supraincarcare: a metodelor, a operatorilor  
[23:43] <@Nytro> In PHP se refera la crearea dinamic a proprietatilor si metodelor  
[23:43] == mode/#rocybercon [-b \*!\*@6E553F0F.F0FC5B65.96EEA9AC.IP] by TheJudger  
[23:43] <@Nytro> Astfel, functiile magice \_\_get si \_\_set ne permit un control foarte bun al proprietatilor cu care se lucreaza  
[23:44] <@Nytro> Ideea care sta la baza este simpla si foarte utila, si nu stiu daca sunt si alte limbaje care o implementeaza  
[23:44] <@Nytro> In clasa noastra putem defini niste metode speciale, numite metode magice  
[23:44] <@Nytro> Care se apeleaza automat in anumite cazuti  
[23:44] <@Nytro> Luam mai intai \_\_get si \_\_set pentru lucrul ca proprietati  
[23:45] <@Nytro> Avem clasa Om cu proprietatile \$varsta si \$greutate de exemplu

```

[23:45] <@Nytro> Presupunem ca sunt publice, si le vom putea accesa dintr-un obiect
[23:45] <@Nytro> $ob = new Om;
[23:45] <@Nytro> echo $ob -> varsta;
[23:46] <@Nytro> Ei bine, cand vom incerca sa citim valoarea unei proprietati care nu exista
[23:46] <@Nytro> Sau care nu este vizibila (private, protected)
[23:46] <@Nytro> Se va apela automat metoda magica __get ($nume)
[23:46] <@Nytro> Si vom sti ce proprietati s-a incercat sa se citeasca
[23:46] <@Nytro> Vom defini metoda in clasa
[23:46] <@Nytro> class Om
[23:46] <@Nytro> {
[23:47] <@Nytro>     public $varsta;
[23:47] <@Nytro>     public function __get ($name) { echo 'Se incerca citirea proprietatii: ' . $name; }
[23:47] <@Nytro> }
[23:47] <@Nytro> Cand vom incerca sa accesam
[23:47] <@Nytro> $ob -> ceva pentru citire
[23:47] <@Nytro> Adica de exemplu afisarea
[23:48] <@Nytro> echo $ob -> ceva;
[23:48] <@Nytro> Se va apela automat acea metoda
[23:48] <@Nytro> Si se va afisa acel mesaj
[23:48] <@Nytro> La fel si cu __set
[23:49] <@Nytro> __set se va apela automat cand se incerca atribuirea unei proprietati care nu exista
[23:49] <@Nytro> De exemplu
[23:49] <@Nytro> class Om
[23:49] <@Nytro> {
[23:49] <@Nytro>     public function __set($name, $value) { echo 'S-a incercat setarea proprietatii ' .
$name . ' cu valoarea ' . $value; }
[23:49] <@Nytro> }
[23:50] <@Nytro> Se poate folosi de exemplu la crearea unui vector asociativ cu proprietati
[23:50] <@Nytro> Adica sa avem: array('prop1' => 'val1'...) si asa mai departe
[23:50] <@Nytro> Atat despre ele, trecem la __call si __callStatic
[23:51] <@Nytro> La fel ca mai sus, sunt metode magice care sunt apelate:
[23:51] <@Nytro> __call($nume, $argumente) atunci cand se incerca apelul unei metode care nu
exista
[23:51] <@Nytro> $name va fi numele metodei care se incerca apelare
[23:51] <@Nytro> Iar $arguments un array cu argumentele
[23:52] <@Nytro> __callStatic ($nume, $argumente) se apeleaza pentru apeluri statice
[23:52] <@Nytro> Adica cand avem Clasa::functie_statica()
[23:52] <@Nytro> Trecem la celelalte functii magice
[23:52] <@Nytro> __autoload se foloseste pentru autoincarcarea claselor
[23:52] <@Nytro> Nu este o metoda, este o functie ce nu se declara intr-o clasa
[23:53] <@Nytro> Este apelata automat cand se incerca instantierea unei clase care nu exista
[23:53] <@Nytro> Daca avem
[23:53] <@Nytro> $ob = new MySQL;
[23:53] <@Nytro> Si clasa MySQL nu a fost definita
[23:53] <@Nytro> Si o avem definita intr-un fisier
[23:53] <@Nytro> vom putea proceda astfel
[23:53] <@Nytro> function __autoload($nume_clasa)
[23:53] <@Nytro> {
[23:54] <@Nytro> include_once('clase/' . $nume . '.php');

```

[23:54] <@Nytro> return new \$nume;  
[23:54] <@Nytro> }  
[23:54] == Johane2 has changed nick to Johane  
[23:54] <@Nytro> Ceva ce am uitat sa specific la inceput, un mic amanunt posibil util  
[23:54] <@Nytro> Incepadn cu PHP 5 se poate crea un obiect folosind numele clasei la rulare  
[23:55] <@Nytro> Adica, sa avem \$nume\_clasa = 'MySQL';  
[23:55] <kNigHt> pentru functia autoload numele clasei trebe sa fie acelasi cu numele fisierului  
[23:55] <@Nytro> Da, am dat exemplu  
[23:55] <@Nytro> \$nume = 'MySQL';  
[23:55] <@Nytro> \$ob = new \$nume;  
[23:55] <@Nytro> Sa folosim numele clasei ca un sir de caractere care poate fi regasit la runtime  
[23:56] <@Nytro> De exemplu tipul unei baze de date: MySQL, MSSQL... Cu care vom lucra  
[23:56] <@Nytro> \_\_toString  
[23:56] <@Nytro> Aceasta metoda magica ne permite sa folosim obiectul ca sir de caractere  
[23:56] <@Nytro> Nu trebuie sa aiba parametri incepadn cu nu stiu ce versiune de PHP  
[23:57] <@Nytro> Si va returna un sir de caractere care va reprezenta obiectul cand acesta va fi folosit  
ca sir de caractere  
[23:57] <@Nytro> Un fel de conversie la sir de caractere  
[23:57] <@Nytro> public function \_\_toString() { return 'Eu sunt Om'; }  
[23:57] <@Nytro> Daca o vom avea ca metoda in clasa Om  
[23:57] <@Nytro> Cand vom folosi clasa Om ca sir, se va returna acel sir  
[23:58] <@Nytro> \$ob = new Om;  
[23:58] <@Nytro> print \$ob;  
[23:58] <@Nytro> Va afisa acel mesaj  
[23:58] <@Nytro> Sau daca concatenam ca sir  
[23:58] <@Nytro> \$ob . 'Un alt sir'  
[23:58] <@Nytro> Va fi sirul de mai sus cu acesta concatenat  
[23:58] <@Nytro> \_\_invoke  
[23:58] <@Nytro> \_\_invoke(\$x)  
[23:59] <@Nytro> E o metoda magica care va fi apelata atunci cand se foloseste obiectul pe post de  
functie  
[23:59] <@Nytro> \$ob = new Om;  
[23:59] <@Nytro> \$ob(123);  
[23:59] <@Nytro> Va apela \_\_invoke cu parametru 123  
[00:00] <@Nytro> public function \_\_invoke(\$x) { print 'S-a apelat cu :'. \$x; }  
[00:00] <@Nytro> \_\_set\_state  
[00:00] <@Nytro> Am uitat o liniuta de subliniere in slideshow  
[00:00] <@Nytro> Se apeleaza in context cu functia var\_export  
[00:01] <@Nytro> Functia returneaza reprezentarea unor date sau o returneaza "ca cod" exebutabil  
[00:01] <@Nytro> Cod PHP  
[00:01] <@Nytro> Complicat si nu foarte util, nu vreau sa intru in detalii  
[00:01] <@Nytro> Huh, intrebari?  
[00:01] <@Nytro> Mai e cineva pe aici? :))  
[00:01] <@Andrei> :))  
[00:02] <BGS> ce ce cum??  
[00:02] <BGS> da da prezent  
[00:02] <BGS> eu este aci  
[00:02] <BGS> ^^  
[00:03] <@Nytro> Vreo intrebare?

[00:03] <tdxev> nup  
[00:03] <tdxev> :)  
[00:03] <BGS> prea neinitiat in tainele programarii ca sa pun vre ointrebare :D  
[00:03] <@Nytro> Omul cat traieste invata  
[00:04] <@Andrei> Nytro, e ok. Unii citesc si tac, altii vor citi maine pe wit :)  
[00:04] <@TheJudge> in ce situatii e nevoie mare de oop ?  
[00:04] <@Nytro> Ah!  
[00:04] <@Nytro> Thanks  
[00:04] <BGS> cred ca in aplicatiile mari  
[00:04] <@Nytro> OOP se foloseste la aplicatiile complexe  
[00:04] <@Nytro> La care e nevoie de o buna separare a datelor cu care se lucreaza  
[00:05] <@Nytro> Si in special la aplicatiile la care lucreaza mai multe persoane  
[00:05] <@Andrei> in general se prefera la proiectele team based :D  
[00:05] <BGS> cred ca se pot realiza proiecte de genu fara oop :)  
[00:05] <@Nytro> Astfel o persoana face o calsa pentru ceva, alta pentru altceva...  
[00:05] <@Nytro> Nu poate sa fie totul o invalmaseala de date si functii  
[00:05] <@Nytro> S-ar ajunge la haos  
[00:05] <@Nytro> Da  
[00:05] <BGS> asta atribuint fiecaruia un lucru bine stabilit  
[00:05] <BGS> de facut  
[00:05] <@Nytro> OOP e doar o modalitate de programare  
[00:05] <BGS> si la final sa uneasca codurile :D  
[00:06] <@Nytro> Ce se poate in OOP se poate si procedural  
[00:06] <@Nytro> Doar ca sa este mai curat, mai strict, nu e haos...  
[00:06] <@Andrei> pe langa asta, procedural e mai rapid, doar ca mai greu de organizat :)  
[00:06] <@Nytro> Adevarat  
[00:06] <@Nytro> Bun, trec la serializarea si clonarea obiectelor  
[00:07] <@Nytro> Si cred ca ajunge pentru aceasta seara, ok?  
[00:07] <BGS> clar ;))  
[00:07] <BGS> dar cu siguranta mai dorim sa ne mai prezinti  
[00:07] <@Nytro> Ok, sa trec la subiect  
[00:07] <BGS> aspecte d genu :D  
[00:07] <Dra3Q> :) seara  
[00:07] <@Nytro> :)  
[00:07] <@Nytro> Serializarea obiectelor  
[00:07] <@Andrei> salut ;)  
[00:07] <@Nytro> Uneori, dupa ce folosim un obiect e posibil sa avem nevoie sa il folosim si ulterior  
[00:08] <@Nytro> Astfel, putem serializa un obiect, adica ii putem salva "continutul" pentru utilizari ulterioare  
[00:08] <@Nytro> Pentru serializare se foloseste functia "serialize(\$obiect)";  
[00:08] <@Nytro> Aceasta functie returneaza un sir de octeti care e de fapt reprezentarea acelui obiect  
[00:09] <@Nytro> BRB sa fac rapid un exemplu  
[00:09] <@TheJudge> brb sa iau un algocalmin  
[00:09] <BGS> serveru meu x din creier tocmai a prins semnalul 11  
[00:09] <BGS> :S  
[00:09] <Dra3Q> :))  
[00:09] <@Nytro> Daca avem clasa  
[00:09] <@Nytro> class Test  
[00:10] <@Nytro> Cu public \$date = "Date";



[00:10] <@Nytro> Rezultatul serializarii  
[00:10] <@Nytro> Adica print serialize(\$obiect); va fi  
[00:10] <@Nytro> O:4:"Test":1:{s:4:"date";s:4:"Date";}  
[00:10] <@Nytro> Foarte important ca sunt serializate decat datele  
[00:10] <@Nytro> Proprietatile  
[00:10] <@Nytro> Metodele nu sunt serializate  
[00:11] <@Nytro> Apoi, de la acest sir pe care il putem scrie intr-un fisier apoi il putem folosi in alt fisier  
[00:11] <@Nytro> Putem recrea obiectul initial  
[00:11] <@Nytro> Folosind functia unserialize();  
[00:11] <@Nytro> La apelul functiei serialize()  
[00:11] <@Nytro> Pentru obiectul in cauza se cauta functia magica \_\_sleep()  
[00:11] <@Andrei> De asemenea, daca tot esti aici as sublinia ca functiile astea doua trebuie folosite cu atentie - sa nu depinde de user input ca prezinta niste vulnerabilitati destul de grave :)  
[00:12] <@Andrei> ma refer la serialize/unserialize  
[00:12] <@Nytro> :)  
[00:12] <@Nytro> Functia serialize, va trebui sa returneze un vector cu numele proprietatilor care sa fie serializate  
[00:12] <BGS> ce fel de vulnerabilitati andrei?  
[00:12] <s3v3n> si \_\_autoload, si \_\_set si \_\_get  
[00:12] <@Nytro> Adica un vector cu numele proprietatilor ale caror valori sa fie salvate la serializare  
[00:12] <s3v3n> \_\_autoload chiar in special  
[00:13] <@Nytro> Da...  
[00:13] <@Andrei> BGS, dupa prezentare :)  
[00:13] <BGS> ./k  
[00:13] <@Nytro> De exemplu, pentru o clasa pentru baze de date  
[00:13] <@Nytro> Presupunem ca avem datele \$host, \$user, \$password  
[00:13] <@Nytro> Si vrem sa le salvam in obiectul serializat  
[00:13] <@Nytro> Vom defini atunci metoda \_\_sleep astfel  
[00:14] <@Nytro> public function \_\_sleep() { return array('host', 'user', 'password'); }  
[00:14] <@Nytro> Si in sirul serializat se vor pastra aceste valori  
[00:14] <@Nytro> Apoi, la unserialize(\$sir\_serializat) vom putea reface conexiunea cu baza de date  
[00:14] <@Nytro> Pentru ca unserialize va apela metoda magica \_\_wakeup  
[00:15] <@Nytro> PE care o vom putea scrie astfel pentru a putea reface conexiunea  
[00:15] <@Nytro> public function \_\_wakeup() { mysql\_connect(\$this->host...); }  
[00:15] <Dra3Q> :)  
[00:15] <@Nytro> Important, deoarece nu se salveaza si metodele, este ca in fisierele in care se apeleaza unserialize  
[00:15] <@Nytro> Sa se afle si definitia clasei obiectului  
[00:16] <@Nytro> Deoarece ati vazut ca numele clasei apare in sirul serializat  
[00:16] <@Nytro> Si PHP va sti cum sa creeze acel obiect la unserialize();  
[00:16] <@Nytro> Daca nu e prezenta definitia clasei  
[00:16] <@Nytro> Se va crea o clasa standart in PHP numita \_\_PHP\_Incomplete\_Class\_name  
[00:17] <@Nytro> Care nu are nici o functionalitate, deci nu am facut nimic  
[00:17] <@Nytro> Inca un lucru, observat intr-un post pe RST: daca un obiect are date protected, in sirul serializat vom avea NULL-uri, caractere cu codul ASCII 0  
[00:17] <@Nytro> Insa vom putea parcurge sirul folosint strlen, nu se va opri la acele caractere  
[00:18] <@Nytro> Asta asa, ca idee  
[00:18] <@Nytro> Sa trecem si la clonarea obiectelor

[00:18] <@Nytro> Uneori avem nevoie de o copie a unui obiect, da sa spunem ca avem nevoie de un identificator unic in fiecare obiect

[00:18] <s3v3n> intrebare. daca folosim autoloading, si clasa inca nu a fost inclusa, la deserializare, se va apela \_\_autoload ?

[00:19] <@Andrei> hmm, se apeleaza \_\_wakeup, de \_\_autoload nu cred

[00:19] <@Nytro> \_\_autoload se apeleaza cand vrei sa creezi obiectul

[00:19] <@Nytro> Da, buna intrebare :)

[00:19] <@Nytro> Dar nu cred

[00:19] <@Andrei> ma interesez si eu :)

[00:19] <s3v3n> pai tocmai, daca folosesc lazy loading, se va autoinclude clasa prin \_\_autoload?

[00:19] <@Nytro> Nu e logic sa se apeleze, trebuie sa fie prezenta definitia clasei

[00:19] <@Nytro> Nu ar trebui

[00:19] <s3v3n> s-ar include cu conditia ca in serializare se pastreaza numele clasei

[00:20] <@Nytro> :-?

[00:20] <s3v3n> presupun

[00:20] <s3v3n> trebuie testata chestia

[00:20] <@Nytro> 2 minute sa testez

[00:20] <s3v3n> si sunt aproape sigur ca este pastrat tipul clasei. pentru ca stie ce sa aleaga din mai multe definitii de clasa

[00:20] <@Andrei> Nytro, mai bine la sfarsit :D

[00:21] <s3v3n> da )

[00:21] <BGS> e un adevarat test de anduranta prezentarea lui nytro :)

[00:21] <s3v3n> mai pe la sfarsit, sa nu se intinda chiar spre dimineata

[00:21] <s3v3n> vorbeste de 3 ore ;)

[00:21] <@Nytro> Ok, sa termin, mai am putin si putem discuta

[00:21] <s3v3n> dar sa fiu sincer, imi place :)

[00:22] <@Nytro> Cum spuneam, uneori avem nevoie de un obiect, dar poate vrem sa avem un identificator unic (de exemplu) in fiecare obiect

[00:22] <@Nytro> Aceasta clonare se face folosind cuvantul cheie "clone"

[00:22] <@Nytro> Daca avem \$ob = new Om;

[00:23] <@Nytro> Pentru a clona acest obiect vom folosi

[00:23] <@Nytro> \$ob2 = clone \$ob;

[00:23] <@Nytro> Acest apel va face copierea lui \$ob in \$ob2

[00:23] <@Nytro> Iar daca clasa cu care lucram are metoda magica \_\_clone()

[00:23] <@Nytro> Aceasta va fi apelata dupa clonare si pentru \$ob2

[00:24] <@Nytro> Deci pentru clona obiectului

[00:24] <@Nytro> Astfel, putem seta ceva aleator de exemplu in \$ob2

[00:24] <@Nytro> public function \_\_clone()

[00:24] <@Nytro> {

[00:24] <@Nytro> \$this -> id\_unic = uniqueid();

[00:24] <@Nytro> }

[00:25] <@Nytro> Aceasta metoda va fi apleata pentru obiectul clona

[00:25] <@Nytro> Cam asta ar fi ideea de baza

[00:25] <@Nytro> Un lucru de mentionat ar fi ca metoda \_\_clone nu se poate apela direct

[00:25] <@Nytro> Adica nu putem apela \$ob -> \_\_clone();;

[00:26] <@Nytro> Cam astea ar fi de spus

[00:26] <@Nytro> Mai ramane slide-ul 8 car mi-ar mai lua vreo ora :D

[00:26] <@Nytro> Si nu am acoperit toate lucrurile importante legate de OOP, cred ca ar mai fi multe de spus

[00:27] <@Andrei> Nytro, eu zic ca e foarte ok. Propun discutii libere de acum :)

[00:27] <@Nytro> Legat de ce am povestit, in afara de intrebarea cu serializarea, mai sunt intrebari?

[00:27] <@Andrei> Bravo, ai un like mare de la mine. :D

[00:27] <@Andrei> :))

[00:27] <s3v3n> :D

[00:27] <s3v3n> +111

[00:27] <s3v3n> \_

[00:27] <s3v3n> )

[00:27] <@TheJudger> great job !!!

[00:28] <@Andrei> BGS, referitor la intrebare

[00:28] <BGS> da prezent :)

[00:28] <@Andrei> acel \_\_wakeup care se auto executa poate da nastere unor probleme:

[00:28] <@Nytro> Multumesc :D

[00:28] <@Andrei> spre exemplu, tu poti (ca user) sa oferi un input. Asta inseamna ca ai putea manipula clasa serializata

[00:28] <@Andrei> si el cand va deserializa va executa si ceva din ce ii dai tu (la wakeup)

[00:29] <@Nytro> La intrebarea cu serializarea

[00:29] <@Nytro> Daca la deserializare se executa \_\_autoload, daca clasa nu e prezenta

[00:29] <@Nytro> DA, se executa

[00:29] <s3v3n> :)

[00:29] <@Andrei> asta inseamna ca poti ajunge la memory corruption, local exploit, executare comenzi cu user de apache (cel putin)

[00:29] <@Nytro> Thanks

[00:29] <s3v3n> dc?

[00:29] <@Andrei> ce de ce?

[00:29] <@Nytro> Da, pare logic sa se execute, se incearca crearea unui obiect plecand de la o clasa care nu exista

[00:29] <BGS> am inteles Andrei

[00:29] <BGS> tx de lamurire

[00:29] <@Andrei> anytime :)

[00:29] <@Nytro> E normal sa se apeleze

[00:30] <s3v3n> a, nimic )) era alt fir de discutie )

[00:30] <@Andrei> Nytro, da e logic. ca pana la urma tot \$x = new class e :D

[00:30] <@Nytro> Da :)