AIAlogo.bmpUniversitatea Politehnica Timişoara

Facultatea de Automaticǎ şi Calculatoare

Departamentul de Automatică şi Informatică Aplicată

**Aplicatie Desktop și Mobliă pentru fotografiere,editarea și afișarea pozelor la predarea cursurilor**

**PROIECT DE LICENŢ**Ă

Coordonator știinţific: Autor:

Ş. l. Dr. Ing. Ovidiu BANIAŞ Paul-Florin TARCE

Timișoara,

Iunie 2018

CUPRINS

[Capitolul 1.Introducere 3](#_Toc514226379)

[1.1 Motivatie și Context 3](#_Toc514226380)

[1.2 Descrierea, domeniul și utilitatea temei 4](#_Toc514226381)

[1.3 Structura lucrării 6](#_Toc514226382)

[Capitolul 2.Studiul Bibliografic 6](#_Toc514226383)

[Capitolul 3.Tehnologii folosite 7](#_Toc514226384)

[3.1 Platforma Microsoft .NET Framework 7](#_Toc514226385)

[3.1.1 Limbajul C# 10](#_Toc514226386)

[3.1.2 Multithreading ,Programarea Concurentă și Async 12](#_Toc514226387)

[3.2 Programarea orientată pe eveniment 13](#_Toc514226388)

[3.3 Biblioteci speciale: AForge ,OpenCV 15](#_Toc514226389)

[[8]Event –based programming 18](#_Toc514226390)

Prezentarea temei proiectului: 17%-25% Se prezintă tema propriu-zisă, modul în care ea este dezvoltată pe parcursul proiectului, legătura dintre capitole precum şi o documentare bibliografică (orice lucrare trebuie încadrată într-un referenţial de cunoştinţe, referenţial pe care autorul şi-l crează prin apelare la surse bibliografice; credibilitatea unei lucrări este strâns legată de acest referenţial care arată cât este de informat şi de avizat autorul cu privire la actualitatea şi necesitatea lucrării pe care o prezintă).

# Capitolul 1.Introducere

## Motivatie și Context

Automatizarea unor procese și realizarea de software care să vină in ajutorul omului în diferite domenii și situații, reprezintă unul din cei mai importanți factori ,piloni, de altfel ai evoluției tehnologice și științifice din ultimii ani.Cum rolul acestor programe este de a face viata omului mai ușoară și munca acestuia mai eficientă , acest proiect reprezintă un început pentru un nou mod de lucru a posibililor utilizatori.

Proiectul abordat în această lucrare vine în ajutorul cadrelor didactice și studenților din cadrul facultății de Arhitectură și Design din Timișoara ,dar aplicația poate să fie folosită în viitor în domeniul educațional și la universități de alt profil sau alte școli.Proiectul acesta este un început a ceea ce se dorește a fi în viitor un suport pentru un nou mod de predare ,care să eficientizeze mai mai mult munca profesorilor care implicit va avea efect asupra randamentului elevilor în timpul orelor.

Contextul acutal arată că pe măsură ce tot mai multe aspecte din arhitectură, design și fabricație sunt transferate în mediul digital , modul în care acestea sunt predate devine un subiect de importanță mare.Tranziția de la vechile modalități de predare și a celor noi este un subiect discutabil.Una din cauzele principale pentru care acest proiect a început a fost trecerea de la tablele negre(cu cretă) la cele albe(cu markere).În ultimii 15 ani în Facultatea de Arhitectura si Urbanism din „Universitatea Politehnica Timișoara” modul în care se predă s-a schimbat mult daorită evoluției tehnologice [1] .Schimbarea tablelor la a avut unele efecte negative în cadrul cusurilor de desing.Cateva din problemle apărute în urma acestei schimbări vor fi prezentate în continuare , pentru a aduce o justificare clară a necesității unei aplicații care să vină în ajutorul celor afectați de această schimbare, dar să aducă și alte beneficii.

Din comportamentul studentilor la aceste cursuri de design s-a observat că tot mai puțini din ei iau notițe la cursuri ,ei bazându-se pe discursul profesorului și pe pozele cu notițele de pe tablă.Aceste poze sunt distribuite între ei pe rețelele de socializare.Pe de alta parte s-a observat la profesori obiceiul de a îngrămădi multe informații referitoare la o problemă pe o singură tablă ceea ce arată nevoia de a fotografia tabla și de a salva pozele.O altă problemă este faptul că o tablă albă pe un perete alb nu mai reprezintă centrul de interes la fel cum il reprezenta o tablă neagră.[1].De asemenea tablele albe au o suprafată care din anumite unghiuri reflecă lumina ,cele negre relectând doar când sunt ude.Grosimea și densitatea liniei cretei pot fi modificate mult mai usor decât cea a markerului.Foarte important pentu aceste cursuri este lucrul cu mai multe variante de desen.Pe tabla neagră acest lucru era mai usor de făcut pentru că

Deși există încercări de a muta tablele în zona digitală interactivă acestea s-au dovedit a fi mai încete decât echivalentul lor analogic.De aceea se dorește o metodă mai convenabilă și eficientă de lucru posibilă doar prin mijloace hibride(analogic – digital).Schițatul figurilor poate să rămână și este mai de folos să fie făcut manual în timp partea digitală(aplicația) completează procesul de predare[1].

Luând în considerare cele de mai sus , se doreste o digitalizare partială a procesului de predare.Modul în care se va realiza acest lucru este cu o aplicație desktop(PC) legată cu o aplicație mobilă prin care se vor face poze la tabla ,edita,și apoi afișa.

## Descrierea, domeniul și utilitatea temei

Sistemul care a fost construit pentru situațiile prezentate în capitolul anterior face parte din mai multe domenii din IT .Ea este alcatuită din 3 părți sau subsisteme care sunt: aplicația desktop(PC),aplicația mobila și aplicația pe microcontroler. Domeniul în care se încadrează în este preponderent cel al aplicațiilor desktop care se foloseste de anumite periferice ,precum camera foto,microcontrolerul Arduino, pentru a avea o funcționalitate specifică și pentru a fi de folos și usor de folosit.

Tema constă în realizarea fotografiilor cu ajutorul unei camere foto sau video conectate la PC, editarea acestora și afișarea lor prin videoproiector pe tabla pe care se predă un anumit curs.Aplicația desktop va putea fi utilizată în mare parte ,pentru ușurință, folosind câteva taste prin care se pot face urmatoarele : realizarea pozelor, afișarea ,salvarea sau ștergerea lor, zoom in/out, rotirea lor,modificarea contrastului sau a luminozității și decuparea pozelor(folosind mouse-ul).Pentru ca această aplicație să fie mai ușor de folosit și să nu fie necesară prezența permanentă la laptop sau PC în timpul predării orelor , în completarea ei s-a construit și o aplicație mobilă care integrează cele mai importante comenzi și funcționalități ale aplicației. Aplicația mobilă va comunica cu aplicația desktop prin Bluetooth.

Aplicația permite controlul direct și de la distanță a camerei foto , calibrarea , procesarea imaginilor și afișarea acestora în timp real pe proiector.În timpul prezentării lecțiilor sau proiectelor, utilizatorul poate să deruleze pozele făcute sau importate, asfel folosind aplicația ca o diaporamă.De asemenea utilizatorul poate comuta ușor între afisarea fișierelor ce țin de curs și pozele realizate cu ajutorul aplicației.

Prin urmare acest proiect rezolvă unele din problemele prezentate anterior și aduce unele plusuri care nu țin neaparat aceste probleme dar care sunt importante pentru ca aplicația să fie usor de folosit și să aibă funcționalități bune asemănătoare unui editor de poze.Astfel profesorul care o folosește poate ca atunci când nu mai are loc pe tablă să scrie , să fotografieze tabla și să revină oricand la pozele anterioare.De altfel, proiectul fiind realizat în special pentru Facultatea de Arhitectură și Design aplicația va fi utilă în a afișa mai multe variante de desen( figuri și planuri arhitecturale) suprapuse și acestea vor putea fi rotite cu un anumit număr de grade și vor putea fi micșorate sau mărite.Folosirea acestei aplicații nu necesită un lucru de mare precizie tocmai din considerentul de a nu consuma mult timp în timpul utilizării ei fapt ce ar lua din timpul predării.[1]

Aportul personal la acest proiect îl reprezintă realizarea de la zero aplicațiilor necesare funcționării acestui sistem ceea fapt ce implică mai multe etape din procesul de dezvoltare software.

## Structura lucrării

Această lucrare este alcătuită din 7 capitole.Acestea sunt :

* Capitolul 1. Introducere : cuprinde câteva cuvinte despre domeniul,utilitatea temei , motivația și contextul proiectului.
* Capitolul 2.Studiu bibliografic : documentare bibliografică care are ca scop fixarea referenţialului temei.
* Capitolul 3. Tehnologii folosite : documentarea cunoștiințelor necesare realizării proiectului.Acestea reprezintă fundamentarea teoretică a lucrării.
* Capitolul 4. Specificațiile aplicației: Descrierea interfetelor dintre aplicația software și mediu(sistemele cu care ineracționează aplicația).
* Capitoul 5. Utilizarea aplicației : Prezentarea modului de utilizare a aplicației
* Capitolul 6.Conzluzii : Ce s-a realizat în acest proiect , problemele întâmpinate și anumite direcții de dezvoltare
* Capitolul 7.Bibliografie

# Capitolul 2.Studiul Bibliografic

# Capitolul 3.Tehnologii folosite

În acest capitol vor fi prezentate tehnologiile folosite pentru realizarea acestui proiect , ele fiind uneltele de bază în dezvoltarea oricărui sistem software.Vor fi abordate tehnologiile,criteriile și metodele utilizate pentru dezvoltarea celor 3 subsisteme ale acestui proieect ,si anume : Aplicația Desktop, Aplicatia Mobilă și Aplicația pe microcontrolerul Arduino.Toate aceste 3 aplicații au fost construite folosind tehnologii diferite ,limbaje de programare diferite și au fost interconectate prin anumite protocoale.

## 3.1 Platforma Microsoft .NET Framework

Pentru a construi aplicația de bază a acestui proiect , și anume aplicația desktop , aceasta s-a realizat utilizând platforma de dezvoltare .NET creată de Microsoft. Aplicația a fost construită pentru sistemul de operare Windows și a fost construită cu ajutorul tehnologiilor .NET precum : Windows Forms, C# și Windows Presentation Foundation (WPF). Mediul de dezvoltare folosit a fost Microsoft Visual Studio .NET 2017.

Platforma .NET este o platformă de dezvoltare open source de uz general,folosită la dezvoltarea aplicațiilor pe sistemele de operare: Windows,Linux,iOS și Android.Implementările platformei .NET depind de sistemele de operare menționate anterior și de tipul aplicației dezvoltate(server,desktop,mobilă ). .Net este o mulțime de framework-uri care au o infrastuctură comună cu mici diferențe între acestea.[3].Aceasta a apărut în anul 2002 și a devenit rapid o parte de bază în dezvoltarea software din prezent.Introducerea lui a avut ca scop de a oferi programatorilor un model de programare mai puternic , mai flexibil și mai simplu față de ce oferea predecesorul acestuia ,COM.Cele mai importante funționalități și caracteristici oferite de .NET sunt :

* Oferă suport pentru mai multe limbaje de programare: C# ,Visual Basic,F# ,C, C ++ și asa mai departe.
* Biblioteci de bază cuprinzătoare( *.NET Framework Class Librasry)*.Această bibliotecă cuprinde mii de tipuri predefinite care premit programatorului să construiască : alte biblioteci ,aplicații de tip consolă, aplicații grafice destkop și aplicații web la nivel de întreprinderi.
* Un nucleu de runtime comun tuturor limbajelor .NET.Aceste este *Common Language Runtime* sau CLR și are rolul de a localiza , încărca și a gestiona obiecte .NET în numele programatorului.De asemenea CLR se ocupă de gestionarea firelor de exctutie și de verificări de securitate
* Integrarea limbajelor : .NET suportă moștenirea, debugging-ul tratarea erorilor cross-language. [4]



Fig 3.1 .Implementări ale .NET[img 1]

**.NET Core** este o versiune .NET care este cross-platform,adică ea poate să ruleze pe orice sistem de operare(Windows,Linuk,MacOS).

**Xamarin** este o platformă cu ajutorul căreia pot f i dezvoltate aplicații native Android ,iOS,MacOS și Windows folosind limbaje .NET.Această platformă este bazată pe Mono( o comunitate susținută de de Xamarin/Microsoft.

.**NET Framework** dupa cum se vede în Figura 1, e o platformă special construită pentru dezvoltarea aplicațiilor specifice familiei sistemelor de operare Windows.NET dispune de funcționalități specific Windows dar are și biblioteci care extind aceste funcționalități.Multe dintre aceste biblioteci sunt folosite și pentru dezvoltarea aplicației desktop din acest proiect. . [2].



Fig 3.2 Arhitectura .NET [img2]

Printre entitățile de bază în .NET ,după cum se poate vedea și în Fig 3.2 se numără: CLR: *Common Language Runtime ,* CTS: *Common Type System* și CLS: *Common Language Specification.* CTS definește toate tipurile de date posibile ,cum entitățile interacționează între ele și în care un tip poate moșteni un alt tip de dată. CLS este un set de specificații pe care un limbaj de programare trebuie să le aibă pentru a fi un limbaj .NET. CLR este o mașină virtuală prin care sunt gestionate programele la rulare în ceea ce privește necesitățile acestora.Astfel,programatorul nu trebuie să țină cont de tipul procesorului pe care rulează aplicația lui.CLR e primul nivel din .NET Framework.. În figura 3.3 este descris procesul de rulare a unui program scris în unul anumite limbaje .NET. [2] [4]



Fig 3.3 Diagrama CLR[img3]

CLR execută un cod octet numit *Common Intermediate Language(*CIL) sau *Microsoft Intermediate Language* (MISL).Acest cod nu poate să fie executat direct de procesor , codul nativ fiind creeat de către Compilatorul Just-in-Time(JIT Compiler) din codul CIL. [2]

Ca și IDE s-a folosit **Microsoft Visual Studio 2017.**Acesta este un IDE pentru dezvoltarea aplicațiilor desktop ,consolă, web ,mobile etc. Acesta foloseste platforme de dezvoltare Microsoft precum Windows Forms,WPF sau Windows Store suportând 36 de limbaje de programare.[5]



Fig. 3.3 Visual Studio 2017[img4]

### 3.1.1 Limbajul C#

După cum s-a menționat și mai sus **limbajul C#** și platforma .NET au fost aduse în folosință in anul 2002 cu scopul de a oferi programatorilor ceva mai mult decât oferea COM. Limbajul de programare C# are o sintaxă de bază care seamănă foarte mult cu *Java* dar C# face parte din familia limbajelor C ( C, Objective C ,C ++). C# este un limbaj orientat pe obiecte la fel ca și Java amândoua având sintaxă este bazată pe C și care au evoluat din C++. C# , la fel ca si C++ permite supraîncărcarea operatorilor , crearea structurilor de date,enumerărilor și a funcțiilor callback(cu delegați) [6][7] .

În plus , C# are unele funcționalități speciale regăsite în diferite limbaje funcționale ( LISP sau Haskell) cum ar fi : expresiile lambda și tipurile anonime.Odată cu aparișia LINQ (*Language Integrated Query),* C# devine tot mai mult un limbaj unic în universul programării.Din cauză că C# este un hibrid a mai multor limbaje,rezultatul este un limbaj curat din punct de vedere sintactic (ca și Java) ,simplu ca si Visual Basic, și puternic și flexibil ca și C++.O caracteristică important care reiese din Fig 3.3 este că C# produce cod care se poate executa doar în runtime.[6][7]



Fig.3.4 C# .NET[img5]

Câteva dintre caracteristicile mai importante ale limbajului C# sunt:

* Nu este necesară folosirea pointerilor.
* Gestionarea memoriei se face automat cu ajutorul asa numitului colector de gunoaie(“Garbage collector”).
* C# oferă suport pentru programarea bazată pe atribute.Sunt permise adnotări ale tipurilor și ale membrilor acestora.
* Operatorul lambda ( => ) ,care simplifică mai mult lucrul cu tipurile *delegate* din .NET Framework.
* Suport pentru căutarea dinamică a membrilor la rulare(runtime) folosind cuvântul cheie *dynamic.*
* Permite programarea asincronă și cu fire de execuție care au fost simplificate mult prin apariția cuvintelor cheie : *async* și *await.*
* Posibilitatea de a costrui tipuri generice și membrii generic.Folosing membrii generic codul scris poate să fie type-safe care definește numeroși substituenți specificați atunci când se interacționează cu tipul generic.
* Un nou fel de inițializare a obiectelor ,care permite setarea valorii proprietaților la momentul creării acestora.
* Parametrii opționali în cadrul metodelor.[7]

În Martie 2017 a apărut curenta versiune majoră de C# și anume C# 7.Acesta a adus noi funcționalități pentru a eficientiza baza de cod și a mai adus tuplii și variabilele și parametrii *ref.*Alte funcționalități :

* Existenta tupliilor( tipuri de categorie ușoară fără nume) și a variabilelor locale și returnate *ref*(returnarea referinței unei valori în loc de valoarea în sine ).
* Tipurile returnate de tip async generalizate
* Un nou cuvânt chei *default* care permite inițializarea oricarui tip

Posibilibilitatea ca metoda *main* a unui program sa fie *async* [7]

În anul data de 7 Mai 2018, C# a ajuns la vresiuna stabilă 7.3. În acest proiect a fost folosită versiunea 4.7.1 a platformei .NET care a fost instalată odată cu instalarea IDE-ului Visual Studio 2017(versiunea 15.4.4), iar versiunea C# folosită a fost 7.2. ... wikipedia

### 3.1.2 Multithreading ,Programarea Concurentă și Async

Aplicațiile care funcționează bine sunt cele care au un timp de răspuns foarte bun și în cadrul cărora atunci când se pornește un task, acesta nu împiedică alte părți din program să funcționeze așa cum trebuie.Realizarea acestui lucru necesita operații mai complexe in C++ dar platforma .NET a adus cu ea căi de realizare a multithreading-ului mult mai ușoare.Un fir de execuție este definit ca o cale de execuție într-o aplicație.Prin crearea firelor de execuție adiționale se pot construi aplicații mult mai receptive.Spațiul de nume *System.Threading* a apărut odată cu .NET 1.0 și oferă suport pentru a construi aplicații cu fire de exectuție[7]

În acest proiect recepționarea datelor de la portul serial(în cadrul aplicației desktop) s-a realizat în mod asincron și a fost necesar un alt fir de execuție decât cel al interfeței grafice.Pentru a transmite datele de la firul nou creat la firul interfeței grafice este nevoie de un funcție speciala de tip **delegate.**Un delegat .NET este practic un pointer la o funcție type-safe. Delegații au rolul de a transmite metode ca și parametrii pentru alte metode. Delegatului îi pot fi transmise metode care au aceeași semnătură cu a sa(include și valoare returnată).Delegații sunt ideali pentru metodele de tip *callback*. [7][2].

## 3.2 Programarea orientată pe eveniment

Programarea orientată pe eveniment este o paradigmă de programare din programarea calculatoarelor în care cursul unui program este determinat de anumite eveniment precum primirea informațiilor de la senzori , click-ul unui mouse, o tastă apăsată, trecerea unui anumit timp etc.[9] Ceea ce diferențiază programarea orientată pe eveniment de alte paradigme de programare este că această paradigmă este un mod de a gândi problemele și soluțiile lor.Furnizează abstractizări.Limbajele sunt cunoscute ca și limbaje funcționale și orientate pe obiecte pentru că ele atribuie abstracțiuni unor structuri de program.La fel se întâmplă și cu programarea bazată pe eveniment, modelul evenimentelor fiind abstractizarea principală.[9].

La baza programării orientată pe eveniment stă conceptul de eveniment.Trei tipuri de obiecte computaționale sunt associate fiecărui eveniment : sursa evenimentului(*event source*), obiectul eveniment(*event object*) și unul sau mai multe proceduri eveniment(*event handlers*).

În realizarea acestui proiect , pentru aplicația Desktop și cea mobilă în mare parte s-a folosit paradigm programării orientate pe eveniment.Elementele cele mai importante din această paradigmă sunt prezentate mai jos:

* *Sursele evenimentului(Event Sources)*.

Acestea reprezintă originea evenimentului.O sursă declanșează un eveniment atunci când creează un obiect eveniment și îl pregătește pentru procedurile de evenimente.Sursele de de evenimente pot să fie obiecte de tip software,hardware sau firmware.

* *Obiectele sursă(Event Objects)*

Un eveniment obiect încapsulează datele importante cu privire la un eveniment.Clickul unui mouse ,de exemplu, este printre cele mai des întâlnite evenimente.Datele corespunzătoare fiecărui click sunt:coordonatele ecranului sau ferestrei curente și butonul care a fost apăsat.Orice procedură eveniment care “ascultă” aceste clickuri va avea nevoie de aceste informații așa că orice procedură,indiferent de limbaj va avea nevoie de ele.

* *Procedurile Eveniment(Event Handlers)*

Procedurile eveniment răspund la evenimente prin executarea actiunilor impuse de programator. Aceste proceduri sunt factorul care unește diferite obiecte pentru a întocmi un program funcțional.Procedurile sunt asociate cu sursele lor(event sources),apoi asteaptă ca să sursa să declanșeze



Fig 3.5 Cursul unui program orientat pe evenimente[img 7]



Fig 3.6 Modelul de procesare al evenimentelor

pentru programarea orientata pe obiecte[img 8]

Intrfețele grafice(GUI) sunt o bună cale de a ilustra baza programării pe eveniment.Partea fundamentală a programării orientate pe obiecte sunt clasele și obiectele.Același lucru se aplică și pentru cea orientată pe eveniment.Figura 3.5 și Fig 3.6 arată relașia dintre diferitele părți prezentate mai sus.

## 3.3 Biblioteci speciale: AForge ,OpenCV

Pentru lucrul cu camera foto/web și cu imaginile primite de la aceasta s-a optat pentru folosirea framework-ului **AForge .NET** și **Emgu CV**( wrapper pentru **Open CV**).

AForge .NET este un framework scris în C#, open-source , creeat pentru dezvoltatori și certetători din domeniile : Inteligență Artificială,procesare de imagini,rețele neuronale,computer vision.De asemena mai poate fi utilizat în logică fuzzy ,machine learning și robotică.În acest proiect acest framework a fost folosit pentru a detecta camerele web conectate la PC , pentru a prelua frame-urile și pentru a realiza poze(a extrage un frame la un momentdat).A mai fost folosit și pentru a edita anumite imagini.

Această platformă este alcătuită din multe librării : AForge.Imaging, AForge.Vision, AForge.Neuro, AForge.Fuzzy. Pentru proiect au fost folosite Aforge.Video și Aforge.Imaging.Pentru a capta informații de la camera web s-au folosit clasele FilterInfoCollection și VideoCaptureDevice și evenimentul de NewFrame. [10]

**Emgu CV** este o înfășurătoare .NET a frameworkului **OpenCV(**folosit la procesarea imaginilor,C++)care permite utilizarea funcțiilor OpenCV în cadrul plaformei .NET. Emgu CV poate fi compilat în Visual Studio,Xamarin ,Unity și poate rula pe Windows ,Android, MacOS etc.EmguCV este scris în totalitate în C# [11]

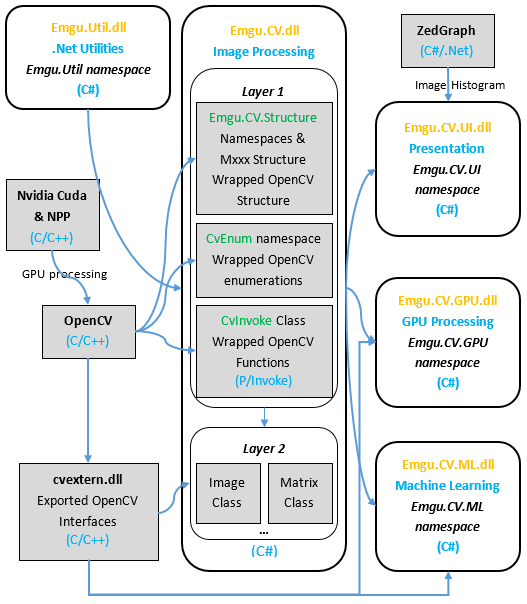


Fig 3.7 Arhitectura OpenCV și wrapper-ul EmguCV[img9]

3.1.5 WPF, Forms

3.1.6-biblioteci folosite: Aforge , OpenCV

3.2 Arduino cu c ++

3.3 Android Studio si Java

3.4 Partea Hardware : Bluetooth]

3.5 Comunicare intre componente

3.6 Testare

Software - > rezolva probleme din lumea reala . Face viata mai usoar ..Aduce solutii pentru probleme .Imbunatateste modul de lucru

-inlocuire bb cu wb

-contrast schimbat , reflexii (lucesc)

-grosimi de linii ->amfiteatre mari

-grosime greta :1 cm ...marker : 3-4 mm

- variatie gorsimi de linie : pe marker nu trebuie

-Stergerea : aici intervine aplicatia mea .

- in desen de arhitectura ..trag linie ,dau cu burete uscat si ramane in partea din spate urma si apoi fac varianta 2 (pe langa , peste) si apoi 3 si 4 ( in arhitectura se lucreaza mult cu variante) .Partea de calc -> sa pun un desen mai palid pe talbal . Sa pot sa il micsorez sau maresc (am o cladire -. Sa detaliez casa ) . Rotirea planului : N in sus (clasic ) ..dar cand se detaliaza cladirea , poza trebuie rotita

-pe white board – l-am sters ..RIP – nu mai pot lucra cu variante

-in predare : pauze de copiat de pe tabla , elevii intreaba ce fost inainte pe tabla .Se inghiesuie pe tabla tot cursul.

-A se face pdf cu tablele din cursul X si sa se trimita pe mail

-La predare ... profesorul foloseste app de pe calculator

-Camera obturata -> jaluzele

-proiectare imagine Negru

FOLOSIRE : Prezentare Teme ale elevilor , ei vin cu solutii foarte variate au mai multi acceeasi greseala. Trebuie tinut un curs cu PDF( Alt + Tab -> schimb de la pdf la aplicatie) nu stimultan

Aplicatia trebuie sa nu necestie lucru mare de precizie ( ca sa nu se intrerupa cursul

- architecture and design

Min 11

-



...pag 749 - 3659

-Carte Berian : [2]

-https://en.wikipedia.org/wiki/.NET\_Framework[3]

[4] ..carte .Net Framework 4.6

[https://en.wikipedia.org/wiki/Microsoft\_Visual\_Studio [5](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio%20%5b5)]

[6] C# 7.0 book all in one for dummies

[6] C# 7.0 book all in one for dummies

[7] Pro C# 7

[8] [https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies (nuu](https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies%20(nuu) am scris de aici)

## [8]Event –based programming

[9] <https://en.wikipedia.org/wiki/Event-driven_programming>

[10] <http://www.aforgenet.com/framework/>

[11] http://www.emgu.com/wiki/index.php/Main\_Page

Imagini :

[img 7] <https://www.slideshare.net/ashwiniawatare/event-oriented-programming>

[img 8] Carte

[ [img 9] <https://www.google.ro/search?biw=1280&bih=893&tbm=isch&sa=1&ei=a9L7Wtu_H5SjmwXP2ILABQ&q=opencv+architecture&oq=OpenCV+architec&gs_l=img.3.0.0i19k1l2.9555.11066.0.11901.9.6.0.3.3.0.140.383.0j3.3.0....0...1c.1.64.img..3.6.404....0.4-hQfthNgbg#imgrc=mbENGINorYG7QM>