

Detailed Design

<http://lbgeeks.com/gitc/systemsDeta.php>

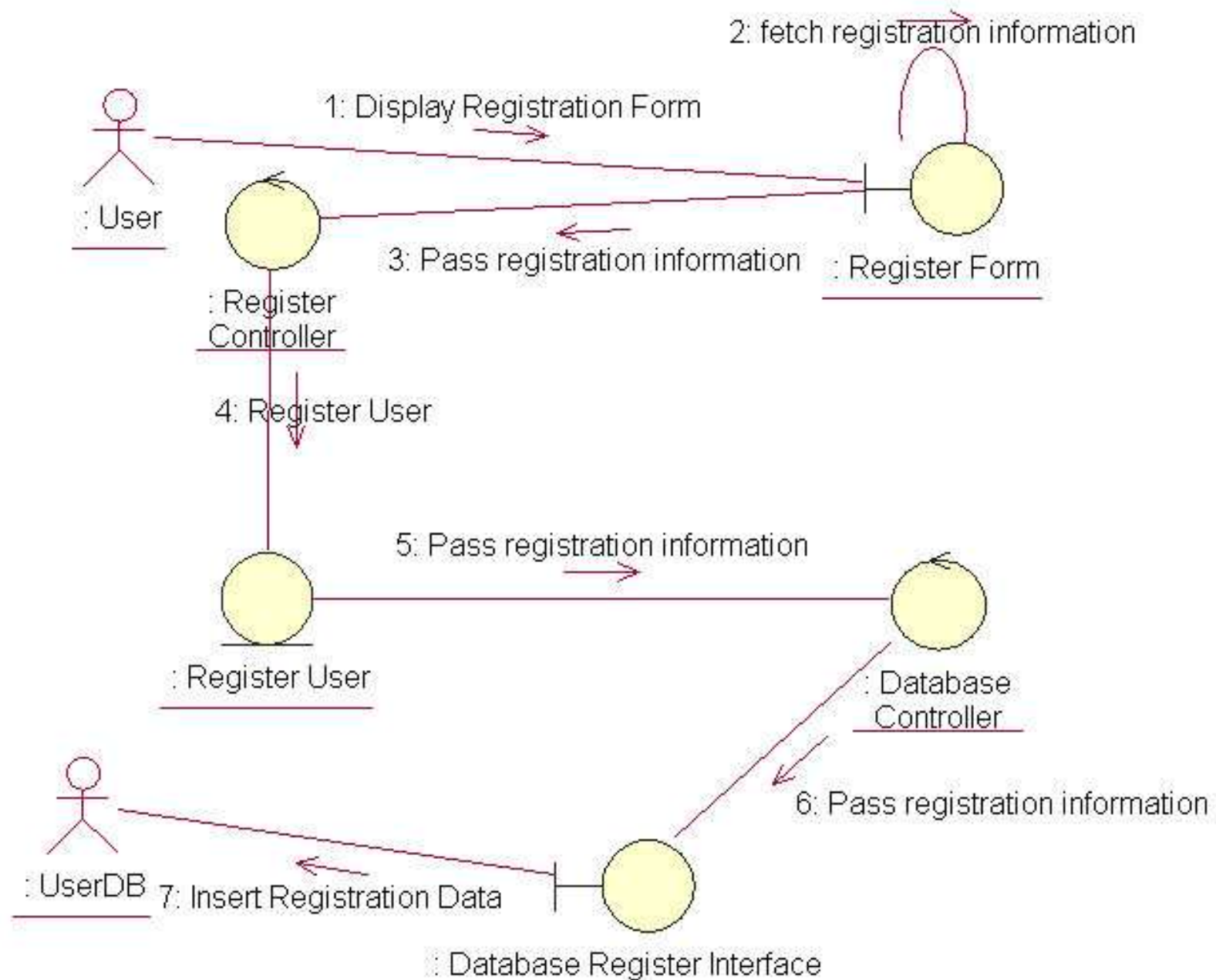
April 30, 2007

- **Նախագծման ենթափուլեր**
- **Դոկումենտավորում**
- **Աղյուսակների նախագծում**
- **Մարդկային ֆակտորներ**
- **Պլանավորում**

- **Նախնական նախագծում:**
 - Հնդհանուր կազմակերպման ստեղծում
 - Նախագծման տարրերի արտաքին ներկայացում
 - Տրանզակցիա
- **Մանրամասն նախագծում:**
 - Բավարար տեղեկություններ կողմնորման համար
 - Նախագծման տարրերի ներքին ներկայացում
 - Ֆունկցիաներ
- **Նախագծման ենթափուլերի բաժանումը նվազեցնում է սխալները**
 - Բարձր մակարդակի դիագրամները տեսանելի են դարցնում ինտերֆեյսի հետ կապված սխալները
 - Դիագրամի օգնությամբ համոզվել, որ չկան բաց թողնված տարրեր

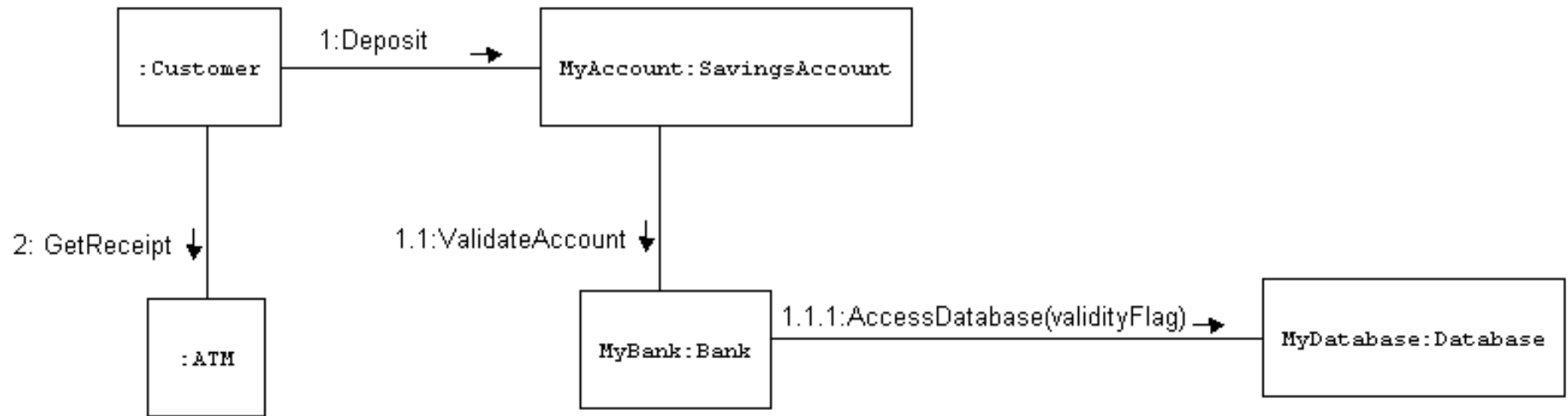
- **Համագործակցության դիագրամներ**
 - **Նպատակ (կրկնություն)**
 - **Օրինակ (կրկնություն)**
 - **Պահանջվող ցանկ**
 - **Պայմանական նշանների կիրառում**
 - **Այլ օրինակներ**
- **Փսևվդոկոդ**
 - **Քննարկում (կրկնություն)**
 - **Պահանջվող ցանկ**
 - **Կազմելու մեթոդներ**
 - **Փսևվդոկոդի եւ ուսալ կոդի օրինակներ**

- Օբյեկտների դիագրամների հաջորդ փուլը (տրամաբանական շարունակություն)
- Իմի է բերում հերթականությունների դիագրամի ինֆորմացիան
- Ցույց է տալիս մեթոդների կանչերի հերթականությունը տրանզակցիաների ժամանակ
- **Executives**-ների ամբողջական սպեցիֆիկացիա

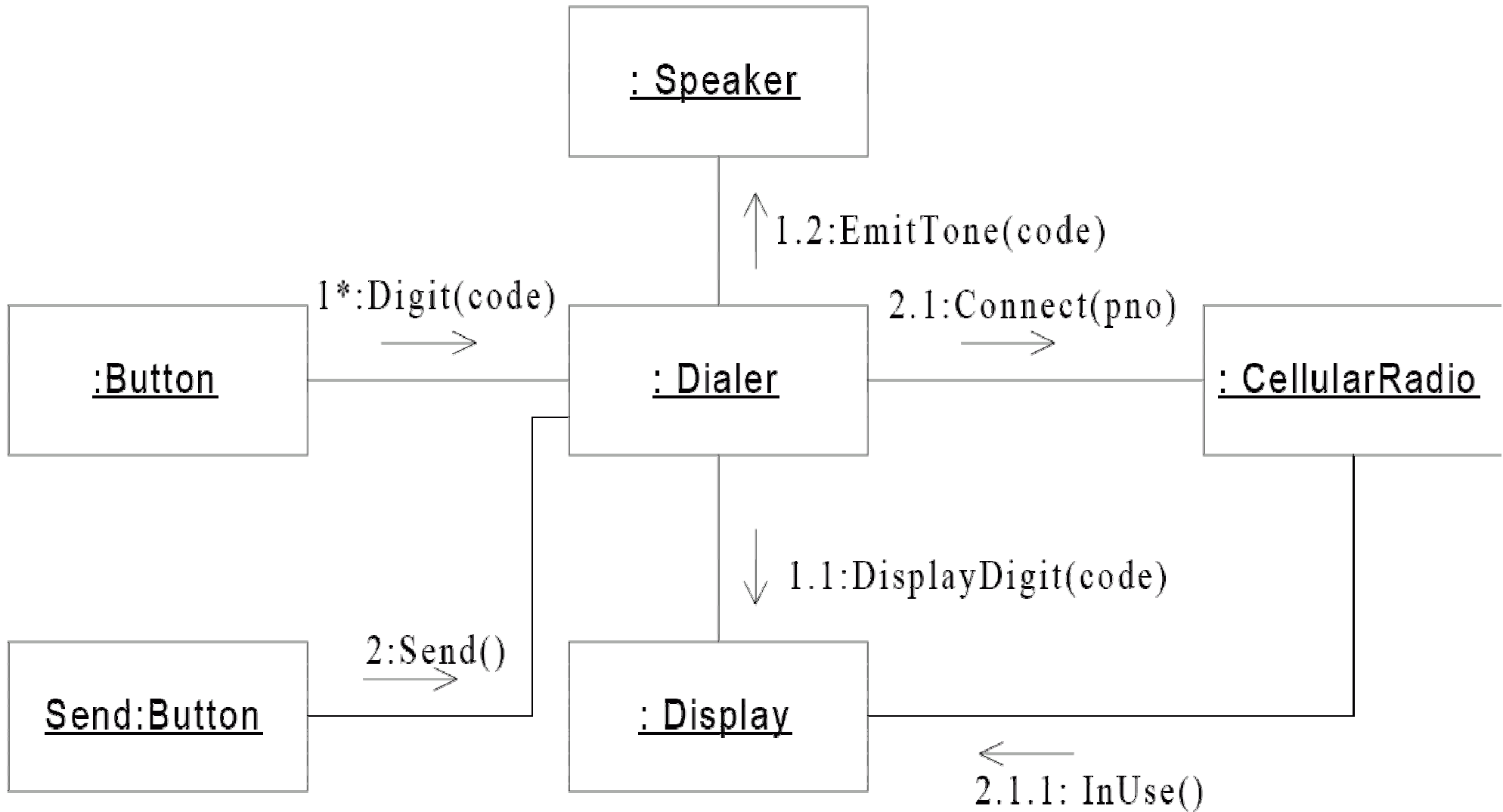


- **Դասեր (class):**
 - Անուն եւ ֆանակ
 - Ներառել օբյեկտի անունը, եթե տեղի ունի անորոշություն
- **Հաղորդագրություններ:**
 - Անուններ, հերթականություն, ուղղություն
 - Պարամետրեր եւ վերադարձվելիք արժեքներ, եթե տեղի ունի անորոշություն
- **Արտաֆին տարրերի մոդելավորում որպես դասեր կամ actors-ներ (գործիչներ)**
- **Համակակալում:**
 - Հերթականությամբ, օրինակ՝ 1, 2, 3...
 - Հիերարխիկ, օրինակ՝ 1.1, 1.2, 1.3, 2.1, 2.2, ...

Համագործակցության դիագրամի օրինակ



Համագործակցության դիագրամի օրինակ



- Իրականացմանը նախորդող վերջին ձուլը
- Նպատակն է կազմակերպել եւ ընտրել ալգորիթմներ եւ արտահայտություններ
- Կոդի արըահայտում կամ նկարագրում բնական լեզվով
- Ձեւի անհամակարգվածությունը կարող է անորոշության պատճառ դառնալ
- Մի ֆանի ֆայլեր կարելի է ներկայացնել մի նախադասությամբ
- Բավարար ինֆորմացիա փորձառու ծրագրավորողի համար գեներացնելու համակարգչային կոդ

- Հիմնական ներքին փոփոխականներ
- Գործողությունների հերթականություն
- Հոսքի կառավարում: ընտրություն (if), ցիկլեր (while)
- Տունկցիաների եւ պրոցեդուրաների կանչեր
- Սխալների հայտնաբերում եւ հետագա մշակում
- Բավարար տեղեկություններ նորեկ ծրագրավորողին կողմէն գեներացնելու համար

Փսևդոկոդի գեներացման մեթոդներ

- Օգտագործել թե համակարգչային, թե բնական լեզուն
 - Հստակ բառերի (բառախմբերի օգտագործում)օրինակ՝ **if-then-else**
 - Չավարտված նախադասությունների կիրառում
- **Significant details**
 - Մանր-մունր մանրամասների իրականացում կոդավորման ժամանակ
 - Ակնհայտ բաների ներառումը խանգարում է պարզությանը
- Անհրաժեշտ աշխատանքի նկատմամբ կենտրոնացում
 - Կարի՛ք չկա նկարագրելու, թե **sort**-ը ոնց է իրականացված
 - Իսկ եթե նկարագրվում է **sort**-ի ֆունկցիան?
- Բալանս պատվիրատուների եւ պրոֆեսիոնալ ծրագրավորողների միջև

Եթե կրեդիտ ֆարտի քալանսը վալիդ է

կատարել տրանզակցիա օգտագործելով ֆարտի համարը, պատվերը

այլապես

ցույց տալ գործողության ընդհատման հաղորդագրությունը

վերջ եթե

```
if credit card number is valid
```

```
    execute transaction using number, order
```

```
else
```

```
    show generic failure message
```

```
end if
```

```
<?php
if (is_valid($cc_number)
    {
    execute_transaction($cc_number, $order);
    }
else
    {
    show_failure();
    }
?>
```

- Ռեկյացիոնալ մոդել
- **Primary** եւ **Foreign** բանալիներ
- Նորմալիզացիայի ձեւեր
 - Նորմալիզացիայի առաջին ձեւ, մակ **1NF**
 - Նորմալիզացիայի երկրորդ ձեւ, մակ **2NF**
 - Նորմալիզացիայի երրորդ ձեւ, մակ **3NF**
- Օրինակը տրամադրված է **study guide**-ում եւ ընթերցման նյութի մեջ

- **Տերմիններ:**
 - Տվյալ
 - Սյունյակ
 - Տող
 - Աղյուսակ
- **Տեսություն:**
 - Աղյուսակը որպես հավաքածուների հավաքածու
 - Սյունյակները որպես տարրերի եւ դասերի ներկայացում
 - Տողերը ներկայացնում են օրինակներ, օբյեկտներ, վեկտորներ
- **Ռեկլյադիոն հանրահաժիվը կիրառվում է հարցումների ընթացքում ստացված տվյալների մանիպուլյացիայի համար**

Primary ու Foreign բանալիներ

- **Primary key-ն** ապահովում է աղյուսակներում չկրկնվող տողերի առկայությունը
- **Foreign key-ն** համապատասխանում է այլ աղյուսակում որոշակի տողի
- **Բանալիները** ինդեքսավորվում են արագ որոնման համար

Նորմալիզացիայի առաջին ձեւ (1NF)

- Յուրաքանչյուր տող պետք է լինի ունիկալ
- Յուրաքանչյուր դաժտ պետք է լինի անբաժան եւ պարունակի հստակ ինֆորմացիա

<u>Anun</u>	<u>Or</u>	<u>Dram</u>
Vartan	Yerkushapti	1.900, -2.820
Vartan	Chorekshapti	-8.400
Lucine	Urpat	10.000, 15.000, -4.000

Նորմալիզացիայի երկրորդ ձև (2NF)

- Համապատասխանում է 1NF
- Բոլոր ոչինդեֆապորված սյունյակները ֆունկցիոնալ տեսակետի կախված են բոլոր ինդեֆապորված սյունյակներից

<u>Anun</u>	<u>Hemtutyun</u>	<u>Hastse</u>
Vartan	Nekarich	114 Main Street
Vartan	Kinonekarich	114 Main Street
Vartan	Gerogh	114 Main Street
Lucine	Inzhener	73 Industrial Way
Shushanik	Tentesaget	73 Industrial Way
Shushanik	Usutsich	73 Industrial Way

Նորմալիզացիայի երրորդ ձև (3NF)

- Համապատասխանում է եւ 1NF, եւ 2NF
- Բոլոր չինդեֆսավորված սյունյակները վերաբերում են ինդեֆսավորված սյունյակին
- Սնյունյակի արժեքը պետք է հնարավոր չլինի հաշվարկել ելնելով այլ սյունյակներում գտնվող տվյալներից

<u>Banali</u>	<u>Anun</u>	<u>Gin</u>	<u>Orer</u>	<u>Inchkan?</u>
1000	Vartan	3.500	3	10.500
1001	Lucine	2.500	2	5.000
1002	Shushanik	2.500	3	7.500

- Սահմանում եւ նպատակ
- Հիմնական սկզբունքներ:
 - Վիզուալ տարրերի հիշողություն եւ կիրառում
 - Սխալների հիշողություն
 - Օգնության (HELP) տրամադրում
 - Ֆորմաների պարզ նախագծում

- **Սահմանում:**
 - Մարդկային ֆակտոր - մարդկային հոգեբանություն, համաձայն որի մարդը գործ է ունենում տարբեր բաների հետ
 - Ինչու կան բաներ որոնք հեշտ է կամ դժվար օգտագործել,
- **Նպատակ:**
 - Հասկանալ ֆորմաների եւ գեկույցների (**report**) նիշտ նախագծումը, արտաֆին տեսքը
 - Նախագծման եղանակներ

- Յուրաքանչյուր վ. տարր ունի հստակ նշանակություն:
 - Կոճակ (command button)
 - Հնարության կոճակ (radio button)
 - Նշեչու դաշտ (check box)
 - Pitak (label)
 - Տեքստային դաշտ(text field)
- Վ.տարրերի նիշտ օգտագործում
- Հակառակ դեպքում պարզ ֆորման կվերաձվի բարդ ֆորմայի

- Օբյեկտները պատասխանատու են սեփական վիճակի համար:
 - Հնդուները սխալ տվյալներ բայց «չթոներ» կամ «չկախվել»
 - Գործողությունների կատարում միայն երբ ամեն ինչ նորմալ է, հակառակ դեպքում՝ ոչ
- Վորտեղ հնարավոր է նախատեսել **default value-ներ**
- Օգտագործողի տեղեկացում բացակա տվյալի մասին
- Մուտքային տվյալների դիապազոնային ստուգում
- Համակարգային սխալների լոկալիզացում եւ մշակում
- Պրոբլեմի վերացմանը ուղղված լուծումների առաջարկում
- Ծրագրին եւ տվյալների շտեմարանին վերաբերող համատեղ սխալներ

- Հնարավորության դեպքում օգնության տրամադրում:
 - “Tool tips”-ներ
 - «Status bar» - ում տեղեկության ցուցադրում
 - Help կոնակներ
 - Hyperlink կամ ձեռնարկ
- Վ.տարրի նշվածության (focus) փոփոխման ժամանակ ինֆորմացիայի տրամադրման 2 կամ 3 պարզագույն հնարքներ
- Ուսուցողական ուժիմներ:
 - Լրացուցիչ ինֆորմացիայի տրամադրում պատուհանների միջոցով
 - Ինֆորմացիայի սահմանափակում՝ ներառնելով միայն ամենաանհրաժեշտը

- Խայտառակ վեբ-կայքի օրինակ:
 - http://www.ueditvideo.com/site2006/index_2.html
- Համեմատե՛ք մեր վեբ-կայքի հետ (lbgeeks.com/gitc)
- Որն է ավելի պարզ?
- Նախագծումը պետք է լինի մատչելի:
 - Գույների նիշտ ընտրություն
 - Տեսողության վատթարացման բացառում
- Պարզ եւ «մաքուր» դիզայնը կարող է եւ մրցանակ չսահի, բայց կժախի ֆանատներ :))

- **Դասը ժամը 19:00:**
 - **Զորեփաբթի, մայիս 2**
 - **Երկուփաբթի, Մայիս 7**
 - **Զորեփաբթի, Մայիս 9**
- **Հաջորդ մասի սկիզբը՝ Մայիս 14**
 - **Թեսթ ժամը 16:45**
 - **Հանձնարարությունը պատրաստել մինչև մասիս 14**