

# ***Systems Development***

---

*Նախնական նախագծում*

*<http://lbgeeks.com/gitc/systemsPrel.php>*

*Ապրիլ 16, 2007*

- Ինչ է նախագծումը
- Նախագծման փուլերը
- Օբյեկտ կողմնորոշված գաղափարներ
- Օբյեկտ կողմնորոշված գաղափարների իրականացում
- Օբյեկտ կողմնորոշված նախագծման գրանցում  
(դոկումենտավորում)
- Մնացած աշխատանքի պլանավորում

- Պատասխան է տալիս այլ հարցի:
  - Պահանջումնեքը հարցնում են “Ինչ ենք կառուցում?”
  - Նախագծումը պատասխանում է “Ինչպես ենք կառուցելու”
- Պահանջումնեքի կիրառում լուծման հատուկ փաթեթի նկատմամբ
- Տվյալ փաթեթի տարրերի ինտեգրացիա արագագործության սահմանափակումների մեջ ֆունկցիաների կատարման համար
- Համակարգային զարգացման միջավայր:
  - Սխալների հայտնաբերում եւ ուղղում
  - Նոր ֆունկցիաների ավելացում

- **Նախնական նախագծում:**
  - Հնդհանուր գործողությունների որոշում
  - Նախագծման տարրերի արտաֆին ներկայացում
  - Տրանզակցիոն կենտրոնացում
- **Մանրամասն նախագծում:**
  - Բավարար մանրամասնություններ կողավորման համար
  - Նախագծման տարրերի ներֆին ներկայացում
  - Ֆունկցիոնալ կենտրոնացում
- **Նախագծման շերտավորումը ֆչացնում է սխալների ֆանակը**
  - Ինտերֆեյսային սխալների հայտնաբերում «նայելով» բարձր մակարդակից
  - Համոզվել, որ չկան բացակա տարրեր

- **Փաթեթավորում (Encapsulation)**
- **Ժառանգում (Inheritance)**
- **Տվյալների աբստրակցիա (Abstraction)**
- **Պոլիմորֆիզմ (բազմաձև) (Polymorphism)**

# Փաթեթավորում (Encapsulation)

- Տվյալ + ֆունկցիա = օբյեկտ
- Մուտքերի, գործողությունների եւ ելքերի խմբավորում գործողություն կատարելու համար
- Դաս - կոմպիլյացիայի ժամանակի ստատիկ հավաքածու
- Օբյեկտ - աշխատանքի ժամանակի օբիեկտ
- Պետք է պահպանի վիճակ:
  - Օբյեկտը կառավարում է սեփական «նախադասությունները»
  - Միայն մուտքերի դեպքում համապատասխան պատասխան գործողություն
  - Երբեք չկատարել վտանգավոր տրանզակցիաներ

# *Ժառանգում (Inheritance)*

---

- Պրոբլեմային հատման կոդավորում, ոչ թե միավորում
- Տալ աղբյուրի(**source**) եւ նպատակակետի(**target**) դելտան
  - Stack - “Last In-First Out” - LIFO
  - Queue - “First In-First Out” - FIFO
  - “push” մեթոդի դելտան եւ ոչ թե “pop”-ինը
- Տեսավորված կոդի վերաօգտագործում
- Սխալ փոխհարաբերությունները որպես սխալ կիրառման պատճառ

# Տվյալների արստրակցիա (Abstraction)

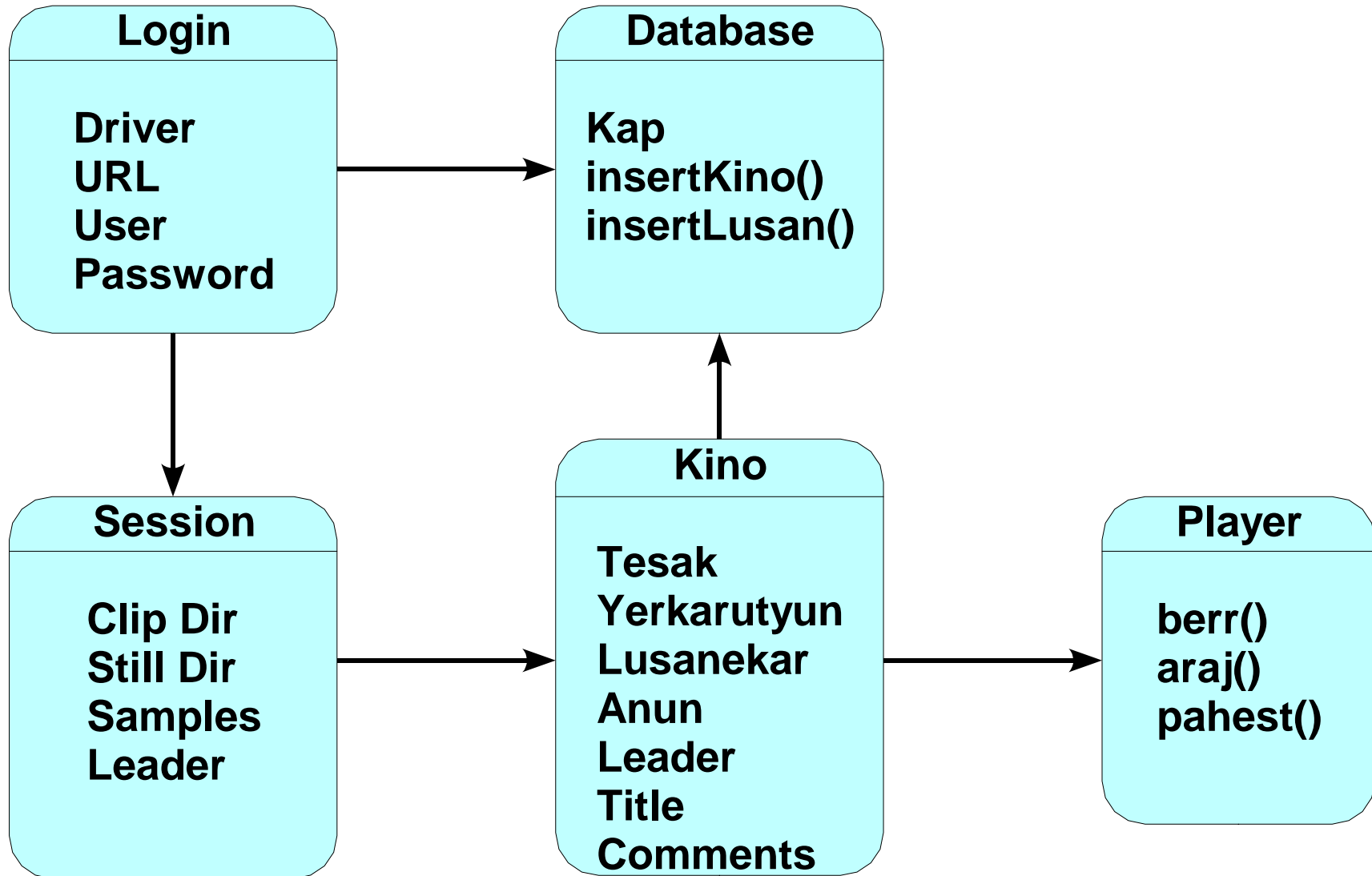
- Կլիենտի եւ սերվերի պատասխանատվությունների բաժանում
- Օբյեկտի օգտագործման բաց ինտերֆեյս
- Հետագա զարգացման փակ ինտերֆեյս
- Ընդհանուր սկզբունքներ:
  - Արգելի դեպի դասի ներքին տվյալները
  - Ունիվերսալ ինտերֆեյսի տրամադրում տարբեր տվյալների տիպերի համար (**templates**)

# Պոլիմորֆիզմ (բազմաձև) (Polymorphism)

- Միեւնույն ձևը դրսևորում է տարբեր վարքեր
- Օրինակ՝ տեխատային դաշտ / ծածկագրի դաշտ
- Բազային օբյեկտի օրինակի ստեղծում տարբեր ածանցյալ դասերից
- Համապատասխան կոդի գեներացում աշխատանքի ժամանակ

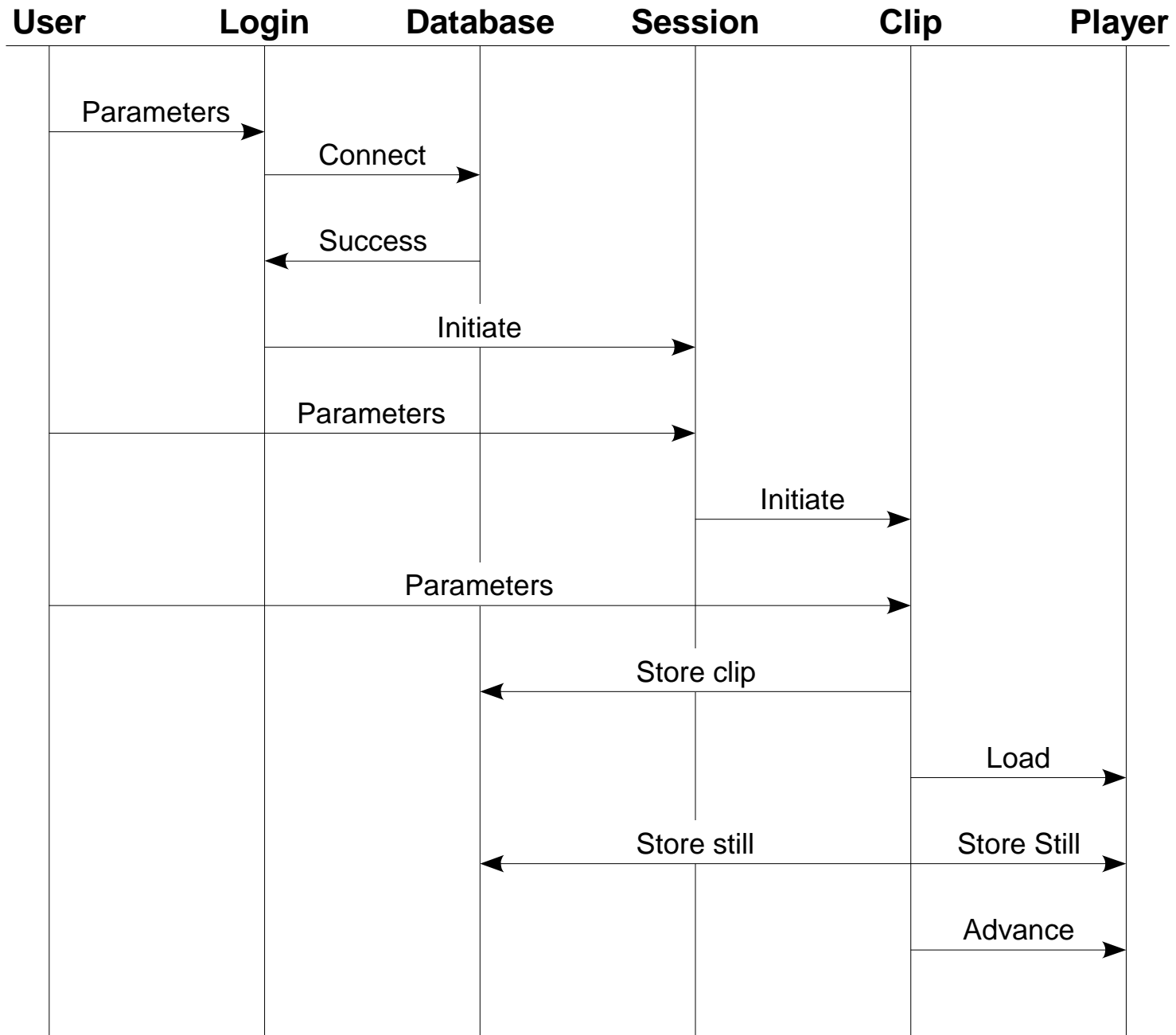
- Օբյեկտի մոդելներ
- Հերթականությունների դիագրամ
- Օբյեկտների փոխներգործության դիագրամ
- Մատրիցայի ձևավորում
- Փսևդոկոդ

- **Դասերի եւ կապերի դիագրամ**
  - Կլորանկյուն ուղանկյունիները - դասեր
  - Անընդհատ գծերը - հարաբերություններ
- **Դասի անունը, հատկություններ եւ մեթոդները դիագրամի վրա**
- **Բառեր, որոնք բացատրում են հարաբերությունների էությունը**
- **Հարաբերությունների 2 հիմնական տիպերը:**
  - “Has-A” պարունակում է
  - “Is-A” ժառանգում



- **Կատարողական հոսք որպես տրանզակցիա**
- **Տվյալների փոխանակում օբյեկտների միջև**
- **Օբյեկտի ներսում ֆայլերի իրականացում**
- **Ժամանակի հերթականությունը վերելից ներքեւ**
- **Ցույց է տալիս համագործակցությունը եւ կախվածությունը**

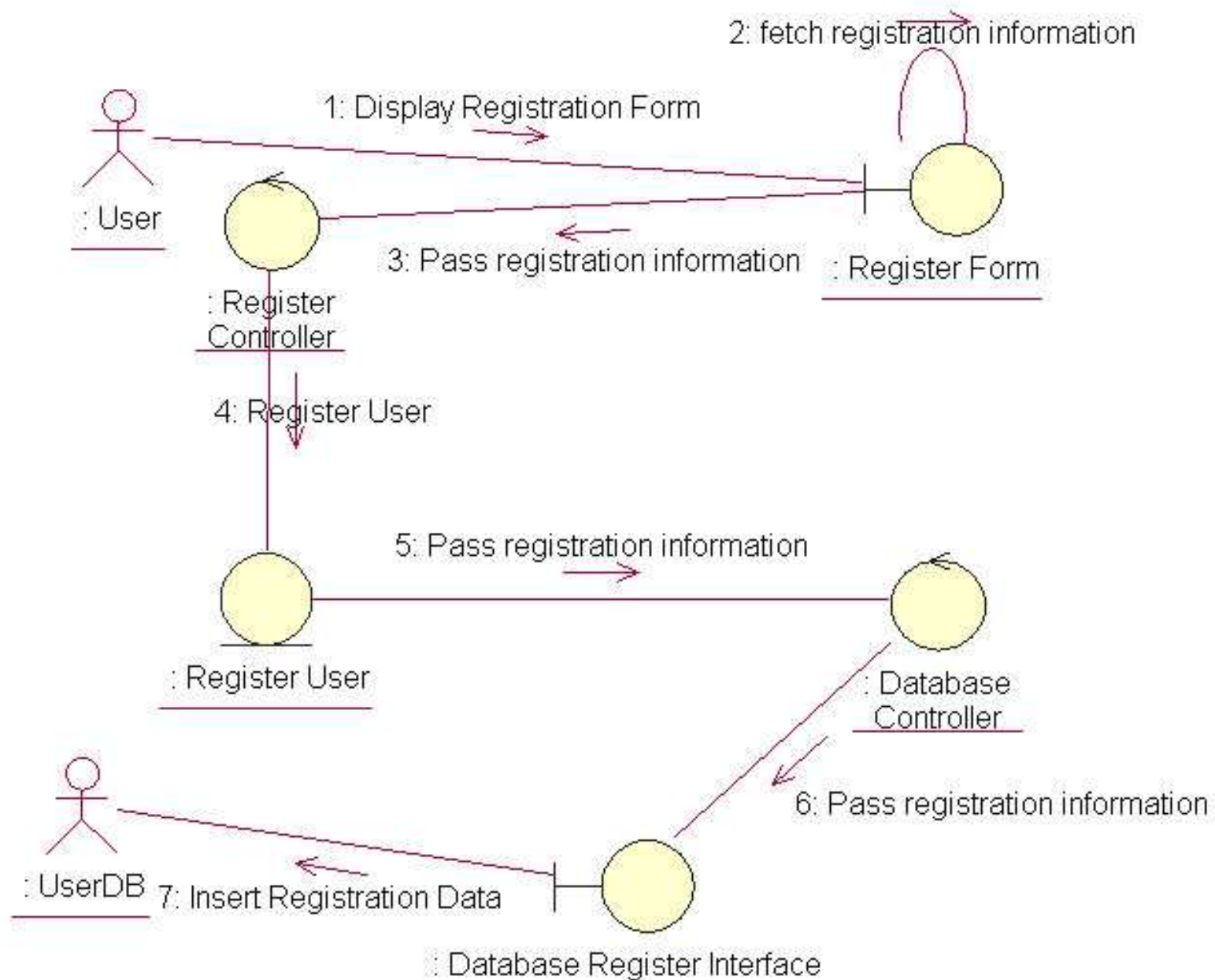
# *Գործողությունների հերթահանության դիագրամ*



# Օբյեկտների փոխներգործության դիագրամ

- Մանրամասնությունների հաջորդ մակարդակը օբյեկտների դիագրամների օգնությամբ
- Օգտագործում է ինֆորմացիա Գործողությունների հերթականության դիագրամից
- Ցույց է տալիս մեթոդների աշխատանքի հերթականությունը տրանզակցիայի ժամանակ
- Կանչերի ամբողջական սպեցիֆիկացիա (մեթոդներ, որոնք կանչում են այլ մեթոդներ)

# Օբյեկտների փոխներգործության դիագրամ



- Պահանջմունքները ունեն ունիկալ ID
- Մի պահանջմունք ենթադրում է նախագծման ժամանակ տարբեր տարրերի կիրառում
- գրանցել “ինչ - ինչպես” ինֆորմացիան որպես աղյուսակ:
- Աղյուսակ մի քանի դաստերով:
  - Պահանջմունքի ID
  - Նախագծման տարրի տիպը
  - Նախագծման տարրի անունը
  - Նպատակը եւ պատճառը

- Իրականացմանը նախորդող վերջին փուլը
- Ալգորիթմների եւ արտահայտությունների կազմակերպում
- Կոդի արտահայտում բնական լեզվով
- Ազատ ոճը կարող է միանշանակ չլինի
- Մի ֆանի ֆայլերի նկարագրում մի նախադասության տեսքով
- Բավական ինֆորմացիա կոդի գեներացման համար

- Փամանակավոր եւ երկարատեւ պահպանում
- Ինկապսուլյացիա - **Encapsulation**
- Տվյալների աբստրակցիա - **Data Abstraction**
- Փառանգում եւ պոլիմորֆիզմ - **Inheritance and Polymorphism**

- **Access** - ծրագրավորում, **Oracle** - տվյալների բազա
- Ինֆորմացիայի երկարատեւ պահպանում - **ORACLE**
- Ժամանակավոր տվյալների պահպանում ֆորմաների մեջ կամ **Visual Basic (VB)** փոփոխականների միջոցով
- Բացառություններ:
  - Ժամանակավոր աղյուսակներ ստեղծված **Jet table space**-ում
  - Փոփոխականներ **PL/SQL** կոդում

- **Access**-ի մակրոսներ կամ **VB** կոդ
- **VB** աշխատանքային հատկություններ
- **Oracle** աղյուսակների երկարատև հատկություններ

- **Variant** տիպ էլ տեստավորման ֆունկցիաներ:
  - **IsDate()**
  - **IsNull()**
  - **IsNumeric()**
  - **IsObject()**
- Ոչ տեսանելի տարրերի օգտագործում ֆորմաներում:  
**visible** հատկություն

- Բազային տվյալներ ընդհանուր աղյուսակում
- Ժառանգված տվյալների կապում **foreign key**-ի միգրացիա
- Վիզուալ տարրերի դինամիկ ստեղծում կամ պատրաստի ֆորմացի բացում՝ **VB If**-ի միջոցով

# ***Class Schedule Reminders***

---

- **Դասաժամը՝ 19:00:**
  - **Չորեփարթի, Ապրիլ 18**
  - **Երկուփարթի, Ապրիլ 23**
  - **Չորեփարթի, Ապրիլ 25**
- **Հաջորդ թեման՝ Ապրիլ 30**
  - **Քննությունը՝ ժամը 16:45**
  - **Հանձնարարության հանձնում մինչև մյուս թեման**