

ЛЕКЦИЯ №1

**Ұғымдардың атауы: программа,
программалаудың деңгейлері және
дәрежелері (бағыттары),
программаларды өңдеу және
аспаптары**

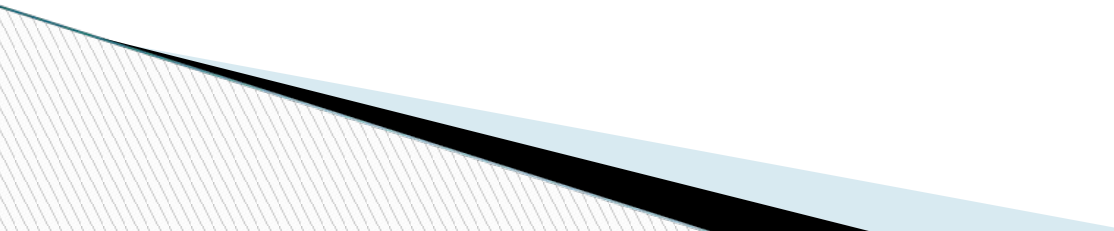
Жұмыстарды орындау үшін құралдар, аспаптар керек болады, әрі программаларды өңдеу және іске асыру **аппараттық және **бағдарламалық** болып бөлінеді.**

Аппараттық – микропроцессор және қосылатын (сыртқы) құрылғылар.

Программалық – барлық жұмыстарды орындауға мүмкіндік беретін, жобалау әдісімен анықталған программалар.

Ұсынылатын пәнді оқыту программалық аспаптық құралдарға бағытталған, компьютерде программаларды өңдеу және орнату үшін қолданылады.

- Программалық өнімді өңдеу (ПӨ) көптеген байланысқан фрагментарлы оқиғаларды көрсетеді:**
- мәліметтердің моделін құру және есептеу әдісі;**
 - есептеуді қамтамасыз ететін функционалдықты сипаттау;**
 - мәліметтердің құрылымын анықтау – компьютерде және алгоритмдегі ұсыныстар моделі;**
 - есептің іске асыру әдісін анықтау және сипаттау (шешу алгоритмі және тесттер);**
 - қолданушы интерфейсін анықтау және сипаттау;**
 - ПӨ құралдарын қолдауын анықтау;**

- есептің спецификациясы;
 - тестілеу бағдарламасын қосып, бағдарламаның текстін жазу;
 - жөндегіш және талқылау, бағдарламаларды тестілеу
 - қолдау кітапханаларын байланыстыру және қосу;
 - орындау ортасын құру; бастапқы модульды орналастыру және жүктеу;
 - кірістірілген көмекті құру және өңдеуді құжаттау;
 - ПӨ орнатылатын пакетті құру.
- 

Rational Unified Process (RUP) шеңберінде программаларды өңдеу бойынша оқиғалардың жиыны келесі кезеңдерде шоғырландырылған:

- талаптарды анықтау;
- жобалау;
- бағдарламалау;
- тестілеу;
- енгізу.

Көрсетілген жұмыстарды орындау үшін үлкен программалар жиыны өңделген және әрдайым толықтырылып отырады – **программаларды өңдеудің процесін** формаға келтіріп және автоматтандыруға рұқсат беретін аспаптар бар. Бұл құралдарды қолдану өңдеудің уақытын қысқартады және программалық өнімдерді енгізеді.

Компьютер үшін программа – салынған - жазылған немесе салынған - құрастырылған программалау тіліндегі тапсырманың ұйғарымы.

Программалау тілі тиімді, сенімді, тез, үнемді жобалау және қойылған есептің іске асыру мақсатымен қолданылады.

Тілдің деңгейі – программалау деңгейі. Төмен деңгейлі программалау – командалар мен кодалық және мнемоникалық формадағы микропроцессор мәліметтерінің форматтарын қолдануымен берілген программалау (ассемблер).

Объектілі–бағытталған (алгоритмдік) (текстуальді) программалау – командалар, функциялар, объектілі-бағытталған тілдердің мәліметтер форматтарын қолданып программалау.

Визуальды программалау – программаларды өңдеудің визуальды құралдарында командалар, компонент, мәліметтер форматы анықталған программалау.

Командалық (атомарлы), құрылымды, модульді программалау – тіл конструкциялары типімен анықталған, программаларды өңдеуде қолданылатын дәрежелер.

Программада құрылатын мәселе облысы программалаудың бағытын анықтайды – бизнес есептері үшін ғылыми, объектілер мен процестерді басқару үшін, ақпараттарды басқару және ұсыну үшін, Интернетке қарым-қатынас үшін, мәліметтер қоймаларымен қарым-қатынас үшін және т.б. пайдаланылады.

Құрылымдық программалау – амал әдістемесі бойынша талдауға, жобалауға және іске асыруға негізделген бағдарлама. Ол 70-ші жылдардың басында пайда болып, қабілетті түрде қолданып бүгінгі күнге дейін тіршілігін көрсетті.

Технология негіздері келесі жағдайларды құрайды:

- Күрделі есеп көптеген ұсақ, функциональды басқарылатын есептерге бөлінеді. Әрбір есеп бір кіріспен бір шығысты қамтиды. Бұл жағдайда программаның басқарушы ағыны ішкі есептердің элементарлы анық функциональды орнынан тұрады.

- Қолданылатын есептерде басқарушы құрылымдардың атомарлығы мен оңайлығы.

Бұл логикалық есептің оңай болуын білдіреді. Функционалды толық жиынтық басқарушысына құрылымдардың бос тұруы жеткілікті болады.

Бағдарлама әзірлеуі кезең бойынша дайындалады. Әрбір кезеңде айқын түсінуі бар, айқын қойылған есептер мәні, шектелген саны мен барлық есепті контексттер ұйғарылуы керек. Егер мұндай түсінік ұйғарылмаса, онда бұл кезең өте ауыр, яғни оны қарапайым кадамдарға бөлуі керектігін айтады.

Модулдік - программалауды тұжырымдау

Дәл осылай құрылымдық технология қатарында, модулдік программалау тұжырымдамасын бірнеше ұғымдар мен жағдайлар түрінде сипаттауға болады:

Есептің функционалдық декомпозициясы –

ауыр есептерді оңай қатар есептерге бөлуі, функционалдық дербес есептердің зор қатарға міндетті бөліктеуі бұл - модулдар.

Модулдар өзара тек қана кіріс және шығыс мәліметтерден байланған.

Модуль - модульдік программалау тұжырымдамасының негізі.

Функционалдық декомпозицияда әрбір модульдің бір кіруі мен бір шығуы бар "қаражәшік" болады. Модульдік жолды пайдалануды, процесстегі жаңартылған бағдарламаны өндіріп алуға, ауыртпалықсыз мүмкіндік бере отырып, оның бақылап отырауын жеңілдетеді.

Қосымша модульдік жол әртүрлі программалау тілдерінде бір жобалық бағдарламаның бір бөлігін өңдеуге мүмкіндік бере отырып құрастыру құралдары көмегімен, кейін олардың біртұтас жүктеуші модулін бірлестіреді.

Жүзеге асырылатын шешімдер айқын және бос тұру қажет. Егер модульдың тағайындауы түсініксіз болса, онда ол бастапқы немесе аралық есептің декомпозициясын сапасыз, жеткіліксіз жүргізгені. Осы жағдайда есепті тағы бір рет талдап, есепті қосымша бөліктерге бөліп өткізуі мүмкін.

Жобада күрделі орындар болған жағдайда, түсініктердің ойластырылған жүйесі көмегімен, **құжаттауды** толықтыру керек.

Барлық айнымалы модулдардың тағайындалуы, олардың анықталуы, шара бойынша, түсініктер арқылы сипатталуы керек. Бұл есеп процесін жалғастырамыз сол уақытқа дейін, ол уақыт олардың ұтымды тіркесін, барлық модулдар тағайындауын, айқын түсініп қол жеткізбегенше.

Объектілі - бағытталған парадигма

ОБП ой талпыныс мәліметтері өңделетін процедуралар мен біртұтас байлауға болатын— объект. ОБП үш маңызды қағидаларға негізделген. Бұл қағидалар **инкапсуляция, мұрагер және полиморфизмдар** болып табылады.

Инкапсуляция – мәліметтер мен бұл мәліметтерді өңдеуінің алгоритімін біріктіруі.

ОБП мәліметтер шеңберінде – **объекттің өрісі**, алгоритмдер – **объектінің әдістері**.

Өріс пен алгоритмдер - сырттай ,- ортақ, сыртқы немесе тек қана объектiнiң iшi - жеке, iшкi немесе сабақтас объекттердiң топтары – қорғалған болып келеді.

Мұрагер –объекттердің қасиеті өз ұрпақтарын тудырып және үнсіздік бойынша өз мүмкіндіктерімен үлестіруді білдіреді.

Объект - ұрпақ, бұл мүмкіндіктерді меншікті толықтыруға немесе оларды (қайта жабу) алмастыра алады.

Полифорфизм – ағайындас объекттердің қасиеті (яғни, объектілер, бір ортақ ата-анасы бар) мағынасы бойынша ұқсас мәселелерді орыннан және қолданудың уақытына қарамастан әр түрлі тәсілдермен шешу.

Программаларды өңдеу аспабы таңдаулы деңгейдің негізінде, бағыт, өңдеудің дәрежелері және дәлме дәл немесе визуальды қасиетпен анықталынады.

Қазіргі программалау – компонентті (объектілі), оқиғалы және визуальды. (Ұғымды анықтау).

Программаларды өңдеу – республикалық және халықаралық, бір үлгідегі немесе технология өңдеушімен ұсынылатын стандарттармен сәйкес келеді.

Аспаптық құралдардың классификациясы

Өңдеу процесінің орны және іске асыруы, уақытша принципі, қолданатын технологиялар мен әдістері, өнімнің сапасы бойынша және т.б.

Пән және тәртіп есептері

Программаларды өңдеудің процедурасында аспаптық құралдардың рөлі және орны.

Программаларды өңдеудің әрбір кезеңі өзінің аспаптар жиынынан тұрады. Пәндік оқу кешенінде өңдеудің барлық қадамдары көрсетілген. Әрбір қадам өңдеудің нәтижесі ретінде – негізгі құжат және кезеңді жабумен қамтамасыз ететін – қосымша құжаттармен анықталынады. Аспап **негізгі** немесе **қосымша** болып келеді.

Сапаның мінездемесі және құрал-жабдықтың қолданылуы

Кеңістікті, уақытша, орнықты, сенімді, қазіргі, интуициялық түсінікті, визуальды, статикалық және динамикалық, өзгертуді қамтамасыз ететін және өңдеуді күйге келтіру, нәтиженің мінездемелері бойынша, стандарттарға және технологияларға сәйкес, мақсаттардың артықшылықтары бойынша және т.б. анықталынады.

Аспаптық жүйелердің дамуының қысқаша тарихи шолуы

1960–1975–1985–2000–2005 жылдарындағы кезеңдері. Тілдер, компиляторлар, құрастырушылар, салушылар, жүктеушілер, операциялық жүйелер, өңдеудің құралдары және тестілеу, енгізу құралдары және бағдарламаны қолдау – тағайындау, қолданудың реті, функционалды және құрылымдық құрамы (конструктивтер). *Программалау* тілдерінің даму сұлбасы соңғы **20 жылдар бойы** даму үстінде.

СРС сұрақтары:

- 1.Программа дегеніміз не?**
 - 2.Аспаптар деген не?**
 - 3.Программалаудың деңгейлері мен дәрежелерін атаңыз.**
 - 4.Модульдік программалаудың тұжырымдамасы?**
 - 5.Объектілі-бағытталған парадигма деген не?**
- 