

БАҒДАРЛАМАЛАУ ТІЛДЕРІМЕН

ТАНЫСУ

I. Бағдарламалау тілдері

- 1) Бағдарламалау тілдеріне шолу.**
- 2) Бағдарламалаудың негізгі құрылысы.**

II. Бағдарламалау тілдерімен танысу

- 1) Процедуралық бағдарламалау.**
- 2) Модульдік және құрылымдық бағдарламалау.**
- 3) Объектіге-бағытталған бағдарламалау.**

□ Компьютерге түсінікті командалар тізбегі арқылы жазылған алгоритм – бағдарлама (программа) деп, ал бағдарлама құру процесі бағдарламалау (программалау) деп аталады.

□ Бағдарламалау тілі – ЭЕМ-де шешілетін әртүрлі есептердің бағдарламасын компьютерге түсінікті үлгіде жазу тәсілі болып табылады.

□ Бағдарламалау тілдері екі үлкен топқа бөлінеді:

- 1) төменгі деңгейлі;
- 2) жоғарғы деңгейлі.

- Төменгі деңгейлі бағдарламалау тілі – машиналық тілге жақын бағдарламалау тілі. Машиналық командаларды жазу үшін жадыда сақтап белгілеу қолданылады.
- Машиналық код дегеніміз - процессор жұмыс жасайтын сандар тізбегі.
- "Төменгі" деңгейдегі бағдарламалау тіліне "Ассемблер" тілі жатады.

Жоғарғы деңгейлі бағдарламалау тілі қолданбалы мазмұнды есептерді шешуге бағытталған, командалардың жиынтығынан тұратын табиғи тілге ұқсас тіл.

Жоғарғы деңгейлі бағдарламалау тілдерін негізгі 2 топқа бөледі:

- 1) функционалдық мәніне (қызметіне) байланысты;**
- 2) қолданылатын бағдарламалау технологиясына байланысты.**

**I. Функционалдық мәніне байланысты 4
үлкен классқа бөлінеді:**

1. бағдарламалауға үйретуші;

2. жалпы мәнді;

3. проблемалық-бағытталған;

4. параллель бағдарламалаушы.

1. Бағдарламалауға үйретуші тілдер:

- Logo тілі, 60-шы жылдардың аяғында С. Пейперттің басшылығымен құрылған.**
- Basic тілі, 1965 жылы Д.Кемени мен Т.Курц құрған.**
- Pascal тілі, 1971 жылы Н.Вирт құрылымдық технологияға үйретуші Pascal тілін құрған.**

2. Жалпы мәнді жоғары деңгейлі тілдер:

- C тілі, 1972 жылы Д.Ритчи құрды және Unix операциялық жүйесі осы тілде жазылды.**
- Ada тілі. 1978 жылы бағдарламалау тілдерін үйлестіру үшін жасалды.**

3. Проблемалық-бағытталған тілдер:

- Fortran тілі, 1956 жылы бірінші компиляцияланған тіл, IBM фирмасы құрды және ғылыми-техникалық есептерді шешуге арналған.**
- Lisp тілі, логикалық бағдарлама құру үшін 50-жылдары Д.Макартни құрды.**
- Prolog тілі, функционалдық бағдарлама құру үшін құрастырылған тіл.**

II. Бағдарламалау технологиялары

Құрылымдық бағдарламалау

- Бағдарламалау процесін жақсартатын және кең қолданылатын әдістердің бірі – құрылымдық бағдарламалау. Оның 3 бөлігі бар:
 1. Модульдік бағдарламалау;
 2. Құрылымдық кодтау;
 3. Жоғарыдан төменге қарай жобалау;

Модульдік бағдарламалау дегеніміз – бағдарламаны логикалық бөліктерге бөлу процесі.

- **Бағдарлама бірнеше модульдерге бөлінеді. Модуль өлшемі 60 жолдан аспауы керек және модульдер өзара тәуелсіз болуы керек.**
- **Модульдерді қолдана отырып бағдарлама күрделілігін төмендетуге болады.**

Құрылымдық кодтау деп бағдарламада басқарушы құрылымдардың – шартты және қайталану операторларының қолданылуын айтады.

Бағдарламаны жоғарыдан төменге қарай жобалаудың өз иерархиялық құрылымы бар.

- Ол қысқа есептің қойылуынан басталады;
- Одан кейін есеп бірнеше ұсақ ішкі есептерге бөлінеді;
- Әр қадамда ішкі есептің орындайтын

Объектіге-бағытталған бағдарламалау

Объекті-бағытталған бағдарламалау тілдерінің негізі – құрылатын қосымша өзара байланысқан негізгі объектілерден тұрады.

Объектіге-бағытталған технологияда қолданушы үш базалық элементпен жұмыс істейді

- Объектілер;**
- Хабарлар;**
- Класстар.**

Негізгі анықтамалар

- **Бағдарламалау тілі** – таңбалар тізбегінен құрылатын бағдарламаны жазуға қажет ережелер жиыны.
- **Бағдарламалау ортасы** - таңбаларды орындалатын есептеулерге түрлендіруге қажет құралдар жиынтығы.
- **Редактор** - бұл бағдарламалау тілінде жазылған бағдарламадан құрылған таңбалық файл болатын алғашқы файлды құру және өзгерту үшін қажет құрал.

- **Компилятор** - алғашқы файлдағы таңбаларды компьютер үшін машиналық кодтағы командалардан тұратын объектілі модульге аударады.
- **Тестілеу** – бағдарламадағы қателерді іздеу процесі.
- **Тестілеу құралдары** - бағдарламаны тестілеу процесінде тестілеу нәтижесін құруды және орындауды автоматты түрде орындайды.
- **Бағдарламаны өңдеу** - бағдарламадағы қателерді жою процесі.

- **Конфигурациялау құралдары** - бағдарлама жасауды және файлдың бастапқы деңгейіне дейінгі өзгерістерді қадағалауды автоматтандырады.
- **Интерпретатор** - тіл командаларын орындайтын, бағдарламаның бастапқы кодын тікелей орындайтын құрал.

Объектілер дегеніміз - бірнеше рет қолданылатын бағдарламалық модульдерден, яғни байланысқан мәліметтер мен процедуралардан тұрады.

Объект құрылымы екі бөліктен тұрады: 1) айнымалылар; 2) әдістер.

Әдістер объект функциясының алгоритмін анықтайтын процедуралар мен функциялар жиынынан тұрады.

Айнымалылар қарапайым мәліметтерден (сан, массив, текст) және күрделі ақпараттардан (график, дыбыс т.б.) тұрады.

Хабарлар объектілердің өзара байланысуы үшін қолданылады және үш бөлімнен тұрады:

- 1) объект идентификаторынан;**
- 2) ағымдағы объектіде қолданылатын әдіс аттарынан;**
- 3) таңдалған әдіс режимін қалпына келтіретін қосымша ақпараттардан.**

Класс дегеніміз – бір типті объектілерге арналған үлгі және ол объектілі айнымалылар типтері мен әдістерін анықтайтын ақпараттардан тұрады.

- Бағдарламалау тілінің синтаксисі бұл бағдарламалау тілінің мүмкін конструкцияларын айқындайтын ережелер жүйесі
- Тілдің семантикасы – бұл бағдарламалау тіліндегі бағдарлама тұжырымының мағынасы
- Бағдарламалаудың орталық концепциясы – тип: бұл мәндер жиыны және сол мәндермен орындалатын әрекеттер көпшесі

Деректердің түрлері:

- Тип - бұл шамалардың жиыны және оларға жасалатын көптеген операциялар. Типті анықтау тілге байланысты. Деректерді талдау үшін кейбір анықтамалар қажет.
- Шама. Қарапайым анықталмаған түсінік.
- Литерал. Бағдарламада таңбалар тізбегі түрінде берілген нақты мән. Мысалы: 154, FALSE, «бағдарламалау».
- Көрсеткіш (представление). Компьютер ішінде нақты биттер жолымен көрсетілген мән. Мысалы: символдық мән 'x' 01 11 10 00 сегіз бит жолымен көрсетілуі мүмкін.

- Айнымалы. Нақты типтегі мән көрсетілімінен тұратын жад ұяшығына немесе ұяшықтарына берілген ат. Мәні бағдарлама жұмысы үрдісінде өзгере алады.
- Тұрақты. Нақты типтегі мән көрсетілімінен тұратын ұяшық немесе ұяшықтар аты. Программа орындалу барысында мәні өзгермейді.
- Объект- бұл айнымалы немесе тұрақты.