



DASTURLASH :KOMPUTERLARNING TILIDA GAPIRISH SAN'ATI

Xabibullayev Jasurbek Xamidulla o'g'li

Axborot tizimlari va texnologiyalari

ATT –AU-25 Yo'nalishi 1-kurs talabasi

Ilmiy maslahatchisi : Ubaydullayeva Diloromxon Mahmudjon qizi

Raqamli ta'lim texnologiyalari kafedrasi o'qituvchisi

Namangan davlat universiteti

xabibullayevjasurbek581@gmail.com

Annotatsiya (qisqacha tavsif):

"Dasturlash — Kompyuter tiling sanati" kitobi kompyuter fanlari va dasturlash sohasidagi eng muhim va klassik asarlardan biridir. Ushbu ko'p jildli monografiyada Donald Knuth algoritmlarni chuqur tahlil qiladi, ularning matematik asoslarini, samaradorligini va amaliy qo'llanilishini batafsil yoritadi.

Dasturlash – bu zamonaviy dunyoning asosiy quroli bo'lib, kompyuterlar va qurilmalarga nima qilish kerakligini ko'rsatib berish jarayoni. Bu nafaqat texnologiya mutaxassislarining ishi, balki ijodkorlik, mantiq va muammo yechish qobiliyatini birlashtirgan fan va san'atdir. Ushbu maqolada dasturlashning asoslari, afzalliklari, mashhur tillari va kelajagi haqida gaplashamiz. Agar siz boshlovchi bo'lsangiz yoki qiziqsangiz, bu maqola sizga yaxshi boshlanish nuqtasi bo'ladi.

Dasturlash Nima?

Dasturlash – kompyuterlarga inson tilida emas, balki maxsus "dasturlash tillari" orqali buyruq berish. Kompyuterlar faqat 0 va 1 lardan iborat ikkilik tizimni tushunadi (binary code), lekin dasturchilar bu jarayonni soddalashtirish uchun



yuqori darajadagi tillardan foydalanadi. Masalan, oddiy bir buyruq orqali veb-sayt yaratish, o'yin ishlab chiqish yoki sun'iy intellektni o'rgatish mumkin.

Dasturlashning asosiy qismlari:

- Algoritmlar: Muammoni bosqichma-bosqich yechish usuli. Masalan, qidiruv tizimini yaratish uchun algoritmlar kerak.

- Ma'lumotlar tuzilmalari: Ma'lumotlarni saqlash va boshqarish usullari, masalan, ro'yxatlar (lists), daraxtlar (trees) yoki grafiklar (graphs).

- Sintaksis va semantika: Tilning qoidalari va ma'nosi. Xato sintaksis dasturni ishlamay qoldiradi.

Dasturlashning tarixi 19-asrda Ada Lovelace va Charles Babbage bilan boshlangan, ammo zamonaviy shakli 1950-yillarda Fortran va COBOL tillari bilan rivojlangan. Bugun dasturlash hayotning har bir sohasida: smartfon ilovalaridan tortib, kosmik missiyalargacha.

Nima Uchun Dasturlashni O'rganish Kerak?

Dasturlash nafaqat kasb, balki ko'nikma. Uning afzalliklari:

- Karyera imkoniyatlari: IT sohasi tez o'smoqda. Dasturchilar, veb-ishlab chiquvchilar, data scientistlar yuqori maosh oladi. Masalan, O'zbekistonda IT kompaniyalari (EPAM, Uzum, yoki xalqaro firmalar) doimiy ravishda mutaxassislar izlaydi.

- Muammo yechish qobiliyati: Dasturlash mantiqiy fikrlashni rivojlantiradi, bu biznes, ta'lim yoki hatto kundalik hayotda foydali.

- Ijodkorlik: O'z ilovangizni yaratish – bu rassomlikka o'xshaydi. Masalan, mobil o'yin yoki shaxsiy blog yaratish orqali.



- Avtomatizatsiya: Oddiy vazifalarni avtomatlashtirish orqali vaqtni tejash.
Misol: Excel fayllarini avtomatik tahlil qilish.

Agar siz O'zbekistonda yashasangiz, dasturlashni o'rganish uchun bepul resurslar mavjud: Coursera, Udemu, yoki mahalliy platformalar kabi Uzinfocom yoki IT-Park kurslari.

Mashhur Dasturlash Tillari

Dasturlash tillari ko'p, ammo boshlovchilar uchun quyidagilar tavsiya etiladi:

1. Python: Eng oddiy va ko'p qo'llaniladigan til. Mashinaviy o'rganish (AI), data science va veb-dasturlash uchun ideal. Misol kod:

```
print("Salom, Dunyo!") # Bu oddiy salomlashuv
```

Python bilan siz Telegram botlar yoki Instagram skriptlarini yaratishingiz mumkin.

2. JavaScript: Veb-saytlar uchun asosiy til. React yoki Node.js bilan ishlaydi. Interaktiv saytlar (masalan, online do'konlar) yaratishda foydali.

3. Java: Mobil ilovalar (Android) va korporativ dasturlar uchun. Kuchli va xavfsiz.

4. C++: Tezlik talab qiladigan o'yinlar yoki tizim dasturlari uchun. Biroz murakkab, lekin qudratli.

5. Swift yoki Kotlin: iOS va Android ilovalari uchun.

Tilni tanlash loyihangizga bog'liq. Boshlovchilar Python bilan boshlasin – u oson va ko'p resurslarga ega.

Dasturlashning Ilovalari va Kelajagi

Dasturlash bugun:



- Sun'iy Intellect (AI): ChatGPT kabi modellar Python va TensorFlow orqali yaratiladi.

- Blockchain va Kriptoalyutalar: Ethereum dasturlari Solidity tilida yoziladi.

- IoT (Internet of Things): Aqlli uylar va qurilmalar.

- Big Data: Ma'lumotlarni tahlil qilish uchun Hadoop yoki Spark.

Xulosaviy Maslahatlar

- Boshlash uchun: Codecademy yoki freeCodeCamp saytlaridan foydalaning. Har kuni 30 daqiqa mashq qiling.

- Amaliyot: GitHub'da loyihalar yuklang va open-source ga hissa qo'shing.

- Muammolar: Debugging (xatolarni topish) – bu normal. Stack Overflow kabi forumlar yordam beradi.

Dasturlash – bu cheksiz o'rganish jarayoni

Veb dasturlash (web development) — bu veb-saytlar va veb-ilovalarni yaratish, ularni brauzerda ishlatish va doimiy ravishda yaxshilash jarayonidir. Bu soha frontend (foydalanuvchi ko'radigan qism), backend (server tarafi) va fullstack (ikkalasini ham qamrab olgan) yo'nalishlarga bo'linadi. <grok:render card_id="a59b9b" card_type="image_card" type="render_searched_image">

<argument name="image_id">3</argument>

<argument name="size">"LARGE"</argument>

</grok:render><grok:render card_id="134a4d" card_type="image_card" type="render_searched_image">

<argument name="image_id">4</argument>



```
<argument name="size">"LARGE"</argument>
```

```
</grok:render>
```

Asosiy texnologiyalar

Veb dasturlashning asosi quyidagi "stack"dan iborat:<grok:render
card_id="98b710" card_type="image_card" type="render_searched_image">

```
<argument name="image_id">0</argument>
```

```
<argument name="size">"LARGE"</argument>
```

```
</grok:render><grok:render card_id="03a355" card_type="image_card"  
type="render_searched_image">
```

```
<argument name="image_id">1</argument>
```

```
<argument name="size">"LARGE"</argument>
```

```
</grok:render><grok:render card_id="8a7b39" card_type="image_card"  
type="render_searched_image">
```

```
<argument name="image_id">2</argument>
```

```
<argument name="size">"LARGE"</argument>
```

```
</grok:render>
```

- Frontend (Old tomom): Foydalanuvchi interfeysi.

- HTML → Sahifa tuzilishi.

- CSS → Dizayn va ko'rinish (Bootstrap, Tailwind kabi frameworklar).

- JavaScript → Interaktivlik (React, Vue, Angular frameworklari).

- Backend (Orqa tomom): Server, ma'lumotlar bazasi va logika.



- Tillar: Node.js (JavaScript), PHP, Python (Django/Flask), Ruby on Rails, Java va boshqalar.

- Ma'lumotlar bazasi: MySQL, PostgreSQL, MongoDB.

- Fullstack: Ikkalasini ham biladigan dasturchi (masalan, MERN stack: MongoDB, Express, React, Node.js).

Qanday boshlash kerak?

1. Asoslarni o'rganing: HTML, CSS, JavaScriptdan boshlang.

- Misol: Oddiy "Hello World" sahifa:

```
<!DOCTYPE html>

<html>

<head>

  <title>Mening birinchi sahifam</title>

</head>

<body>

  <h1>Salom, Dunyo!</h1>

  <p>Bu mening veb sahifam.</p>

</body>

</html>
```

2. O'zbek tilidagi resurslar (2025 yil holatiga ko'ra eng yaxshilari):

- Udemere.uz — O'zbek tilida to'liq dokumentatsiya (HTML, CSS, JS, Python va boshqalar). Onlayn editor ham bor.



- Mohirdev.uz — Frontend asoslari, Django bilan fullstack kurslar (amaliy loyihalar bilan).

- Webiston.uz — Bepul o'zbekcha kitoblar (JavaScript, React va boshqalar).

- JavaScript.info ning o'zbekcha tarjimasi (GitHubda mavjud).

- YouTube kanallari: Sammi.ac, Programmer UZ va boshqa o'zbek dasturchilarning video darslari.

3. Xalqaro bepul resurslar (ingliz tilida, lekin eng sifatli):

- freeCodeCamp.org — To'liq kurslar, loyihalar va sertifikatlar.

- MDN Web Docs (developer.mozilla.org) — Eng ishonchli hujjatlar.

- The Odin Project — Loyihaga asoslangan o'quv dasturi.

- W3Schools — Oddiy tushuntirishlar va misollar.

Maslahatlar

- Har kuni amaliyot qiling: VS Code editorini o'rnatib, oddiy saytlar yarating.

- GitHubda loyihalaringizni joylashtiring (portfolio uchun).

- Keyin React yoki Node.js ga o'ting.

FOYDALANILGAN ADABIYOTLAR RO'YXATI:

1. Абдуллаева Н.М. Педагогик технологиялар асосида замонавий таълим жараёнларини ташкил этиш. – Т.: Ўқитувчи, 2018. – 145 б.

2. Бешимов, Ж.Ф. ва Каримова, Ш.Б. Компетентлик ёндашуви: назарий асослар ва амалиёт. – Самарқанд: СамДУ нашриёти, 2020. – 200 б.



3. Dewey, J. Experience and Education. – New York: Macmillan, 1938. – 91 p.
4. Kolb, D. A. Experiential Learning: Experience as the Source of Learning and Development. – Englewood Cliffs, NJ: Prentice-Hall, 1984. – 256 p.
5. Piaget, J. The Psychology of Intelligence. – London: Routledge, 1950. – 182 p.
6. Vygotsky, L.S. Mind in Society: The Development of Higher Psychological Processes. – Cambridge, MA: Harvard University Press, 1978. – 159 p.
7. Ҳасанов, Ж.М. Таълимда интеграция ва инновациялар: назария ва амалиёт. – Тошкент: Фан, 2019. – 145 б.
8. Савельева, Т.Н. Психолого-педагогические аспекты формирования компетенций. – Москва: Изд-во МГППУ, 2016. – 208 с.
9. Shulman, L.S. Those Who Understand: Knowledge Growth in Teaching. – Educational Researcher, 1986. – №15(2). – p. 4–14.
10. Siemens, G. Connectivism: A Learning Theory for the Digital Age. – International Journal of Instructional Technology and Distance Learning, 2005. – №2(1). – p. 3-10.
11. Taber, K. S. Constructivism as Educational Theory: Contingency in Learning, and Optimally Guided Instruction. – Nova Science Publishers, 2006. – p. 39–61.
12. Тошов, Ж.Т. Компьютер тизимлари ва дастурлаш асослари. – Тошкент: ТДТУ нашриёти, 2021. – 245 б.
13. Trilling, B., & Fadel, C. 21st Century Skills: Learning for Life in Our Times. – San Francisco: Jossey-Bass, 2009. – 258 p.
14. Умаров, А.Б. Компетентлик ёндашуви: назарий асослар ва инновацион ёндашувлар. – Тошкент: Ношир, 2019. – 136 б.
15. Yilmaz, H. & Tüzün, H. Factors Affecting Students' Programming Competence in University Programming Courses.