M. R. Zakirova

Informatika va axborot texnologiyalari

4-sinf

Umumiy o‘rta ta’lim maktablari uchun darslik



Yangi nashr

Ta’limni rivojlantirish markazi huzuridagi
Ilmiy-metodik kengash tomonidan tavsiya etilgan

TEKSHIRILGAN
PLAGIATGA QARSHI
TEKSHIRILGAN

novda
edutainment

Toshkent – 2026

UO‘K: 004(075.3)
KBK 32.81ya72
Z 18

Zakirova M. R.
Informatika va axborot texnologiyalari. Umumiy o‘rta ta’lim maktablarining 4-sinfi uchun darslik / M. R. Zakirova. – Toshkent: “Novda Edutainment”, 2026. – 128 b.

Taqrizchilar:

N. M. Usmanova – Toshkent shahridagi 190-maktabning oliy toifali boshlang‘ich sinf o‘qituvchisi;

O. Sh. Yevstafeva – Toshkent shahridagi 190-maktabning oliy toifali boshlang‘ich sinf o‘qituvchisi.

Shartli belgilar



Savollarga javob bering



Mantiqiy topshiriq



Bu qiziq!



Kompyuterda ishlang



Guruh bo‘lib ishlang



Uyga vazifa



Dars yakuni

ISBN 978-9943-9683-4-9

© M. R. Zakirova
© “Novda Edutainment”, 2026

Salom, aziz bolajonlar!

Mening ismim **Quvnoq robot**. Biz siz bilan informatika olamiga allaqachon ilk qadamlarni qo‘yib bo‘ldik, axborot turlari va uni qayta ishlash usullari haqida bilib oldik. Endi esa bizni yangi qiziqarli sayohat kutmoqda!

Ushbu darslikda biz texnologiyalar va kompyuterlarning hayratlanarli olami bilan tanishamiz, zamonaviy atamalarni o‘rganamiz va algoritmlar tuzishni davom ettiramiz. Bu bilan siz mantiqiy fikrlashni rivojlantirasiz, mulohaza yuritishni, yechim topishni va bilimlarni tartibga solishni o‘rganasiz.

Darslik sahifalarida sizni qiziqarli hikoyalar, maroqli topshiriqlar va ko‘plab kashfiyotlar kutmoqda. Informatika olamiga sayohatga tayyormisiz?

Unda olg‘a!



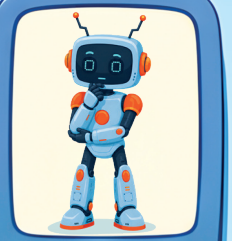
Muhim belgilar



Yodda tut!

Ushbu belgi bilan men sizga matnda yodda tutishingiz kerak bo‘lgan yangi atamalar va tushunchalar borligini ma’lum qilaman.

Ushbu blokda men sizga izohlashingiz yoki yechimini topishingiz kerak bo‘lgan vaziyatlar va vazifalarni taklif qilaman.



O‘ylab ko‘r!



Ogoh bo‘l!

Men sizni xavfsizlik qoidalari haqida shunday ogohlantiraman!

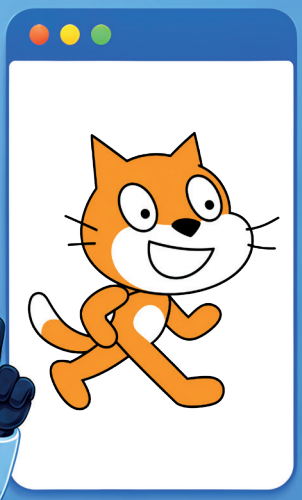
Ushbu blokda men sizga qiziqarli faktlarni bayon qilaman.



Bu qiziq!

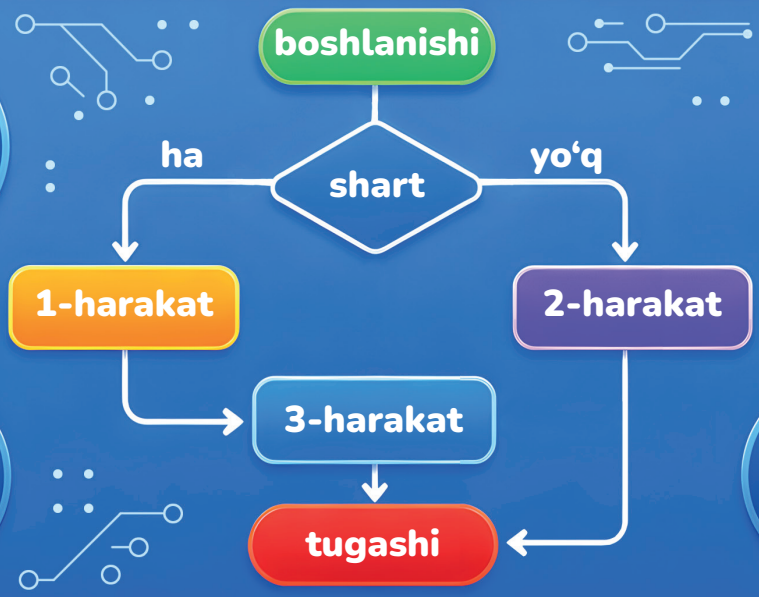






I bob

Algoritmlar va dasturlash asoslari



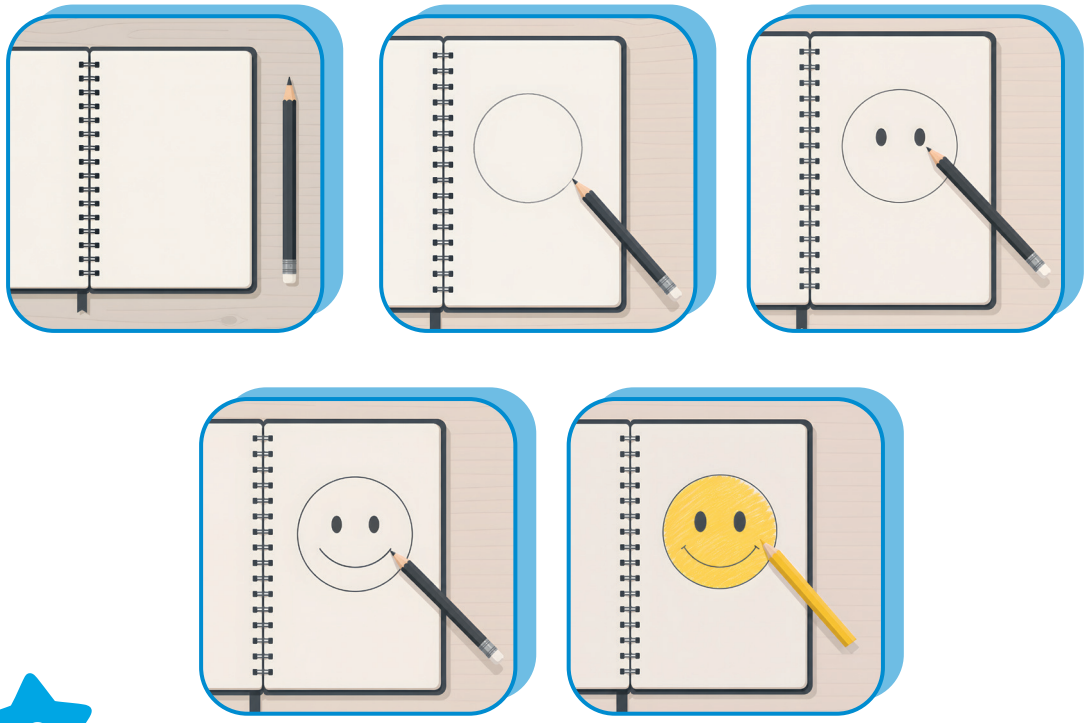


Algoritmlar

- Algoritmlar haqida nimalarni bilasiz?
- Har qanday harakatlar ketma-ketligini algoritm deb atash mumkinmi?
- Algoritmlar nima uchun kerak?

O‘quvchi har kuni turli ishlar bilan shug‘ullanadi: maktabga borish uchun tayyorlanadi, do‘sti bilan telefon orqali gaplashadi, vaqtida ovqatlanadi yoki uyga berilgan vazifalarni bajaradi. Buning uchun u ishlarni to‘g‘ri ketma-ketlikda bajarishi kerak.

1. Rasmlarga qarang. Smaylik chizish bosqichlarini ketma-ket aytib bering.





Bir-biri bilan bog‘liq bo‘lgan, ma’lum tartibda bajariladigan va muayyan natijaga olib keladigan harakatlar **harakatlar ketma-ketligi** deb ataladi.



2. Umid ukasiga oyoq kiyimining bog‘ichlarini bog‘lashni o‘rgatmoqchi bo‘ldi. U harakatlarni qanday bajarish kerakligini bosqichma-bosqich tushuntirdi:

1) Bog‘ichlarni ikki qo‘lingizga oling.

2) Bog‘ichlarni chalishtiring.

3) Halqa hosil qiling.

4) Ikkinchi bog‘ichni halqa atrofidan aylantiring.

5) Bog‘ichni teshikdan o‘tkazing.

6) Bog‘ichlarni tortib, bant hosil qiling.



Umidning ukasi ayrim bosqichlarning o‘rnini almashtirib yubordi va akasi ko‘rsatganidek natijaga erisha olmadi.

Sizningcha:

- Nima uchun natija noto‘g‘ri chiqdi?
- Barcha harakatlarni tartib bilan bajarish kerakmi?
- Harakatlar ketma-ketligidagi ayrim bosqichlarni o‘tkazib yuborish mumkinmi?

Ko‘pgina ishlarda harakatlarni to‘g‘ri ketma-ketlikda bajarish juda muhim. Harakatlarning o‘rnini almashtirish yoki biror bosqichni o‘tkazib yuborish vazifaning noto‘g‘ri bajarilishiga olib kelishi mumkin.

Harakatlar ketma-ketligi vazifani bajarish va kerakli natijaga erishishga yordam bersa, u **algoritm** deb ataladi.

Algoritmdagi har bir alohida harakat **algoritmning bosqichi** deb ataladi.





Demak, har qanday harakatlar ketma-ketligi algoritm hisoblanadimi?



Albatta, yo‘q! Harakatlar ketma-ketligi algoritm bo‘lishi uchun bir qator shartlarga amal qilish kerak.

Harakatlar ketma-ketligi algoritm hisoblanishi uchun quyidagi qoidalarga mos kelishi kerak:

- har bir harakat aniq va tushunarli bayon etilishi kerak;
- algoritm ma’lum sondagi bosqichlardan iborat bo‘lishi kerak;
- algoritmni bajarish belgilangan natijaga olib kelishi kerak.

Algoritmlar turli usullarda ifodalanishi mumkin.

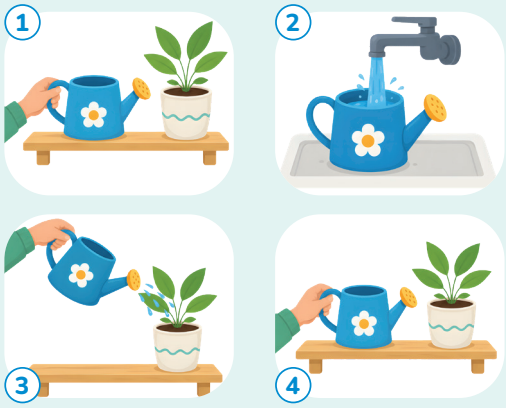
Algoritmni ifodalash usullari

- so‘zlar yordamida
- rasmlar yordamida
- blok-sxema yordamida

So‘zlar yordamida ifodalangan algoritm

1. Telefonni olish.
2. Kamerani ochish.
3. Old kamerani yoqish.
4. Tabassum qilish.
5. Suratga olish tugmasini bosish.
6. Fotosuratni ko‘rish.

Rasmlar yordamida ifodalangan algoritm

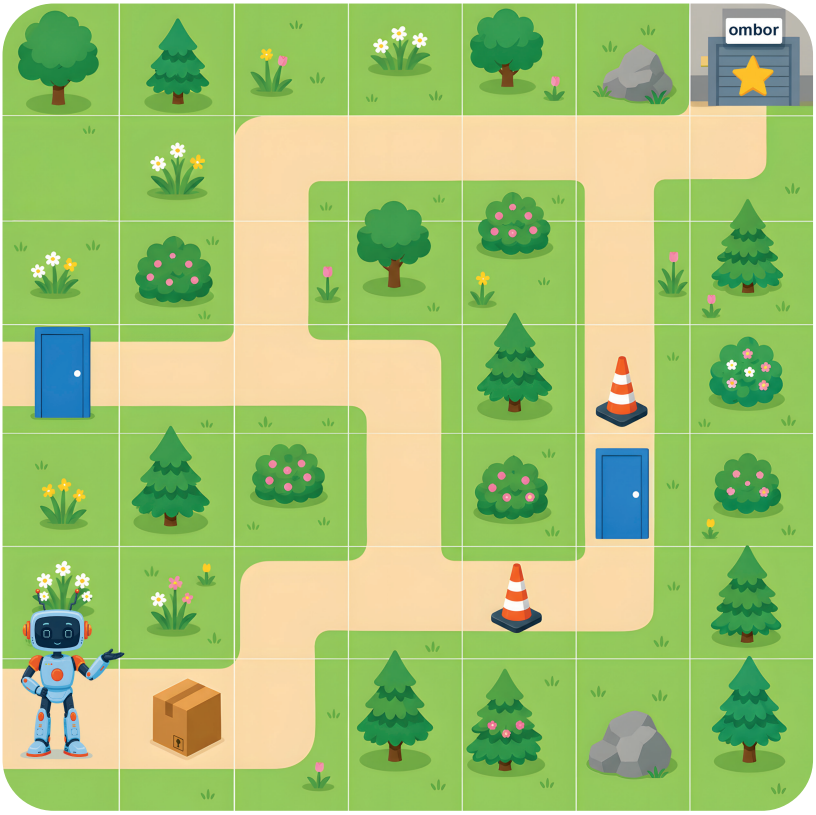


Blok-sxema yordamida ifodalangan algoritm



3. Algoritmni qaysi hollarda so‘zlar yordamida, qaysi hollarda esa rasmlar yoki blok-sxema yordamida ifodalash qulayroq ekaniga misollar keltiring.

4. Qutini omborga yetkazishda robotga yordam bering. Uning harakatlanish algoritmini tuzing.



Robot quyidagi buyruqlarni bajara oladi:

- oldinga bir qadam
- orqaga bir qadam
- chapga bir qadam
- o‘ngga bir qadam
- qutini oling
- eshikni oching
- sakrab o‘ting
- qutini qo‘ying

5. Masalani yechish algoritmini tuzing. Faqat 3 litrli va 5 litrli idishlardan foydalanib, 1 litr suvni qanday o‘lchab olish mumkin?



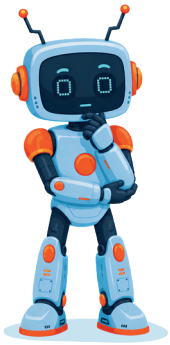
“Algoritm” so‘zi mashhur olim Al-Xorazmiy nomi bilan bog‘liq. U IX asrda yashagan va matematika hamda hisoblash usullarini o‘rgangan. Uning ismi lotin tilida “Algoritmi” shaklida aytilgan. Keyinchalik bu ismdan “algoritm” so‘zi paydo bo‘lgan. Bugungi kunda algoritmlardan kompyuterlar, telefonlar va robotlarda foydalaniladi.



6. Savollarga javob bering.

- Qanday harakatlar ketma-ketligi **algoritm** hisoblanadi?
- Algoritmdagi bosqich nima?
- Nima uchun algoritmda bosqichlarning tartibi muhim?
- Algoritmlarni ifodalashning qanday usullari mavjud?
- O‘z hayotingizdan algoritmga misol keltiring.

Masalani bosqichlarga ajratish



Murakkab masalalarni qanday yechganingizni eslang. Masalani yechish rejasini tuzganmisiz? Masalani amallar bo‘yicha yechganmisiz yoki yechimni darhol bitta ifoda ko‘rinishida yozganmisiz?

Akmal bilan Aziza katta konstruktor yig‘ishyapti. Yo‘riqnomada modelni birdaniga emas, balki qismlarga bo‘lib yig‘ish ko‘rsatilgan.



1. Rasmga qarang. Akmal bilan Aziza avval nimani, keyin nimani yig‘ishganini aytib bering.

Odamlarning murakkab vazifani bajarishda uni kichik harakatlar yoki qismlarga ajratishlari qulay.

Bu quyidagilarga yordam beradi:

- vazifani tushunarliroq qilishga;
- turli qismlarni navbat bilan yoki bir vaqtda bajarishga;
- xatolarni osonroq topish va tuzatishga;
- o‘z harakatlarini yaxshiroq rejalashtirishga.

Informatikada murakkab vazifani oddiy qismlarga ajratish **dekompozitsiya** deb ataladi.

Dekompozitsiya harakatlarning to‘g‘ri ketma-ketligini tuzish va aniq algoritm yaratishga yordam beradi.



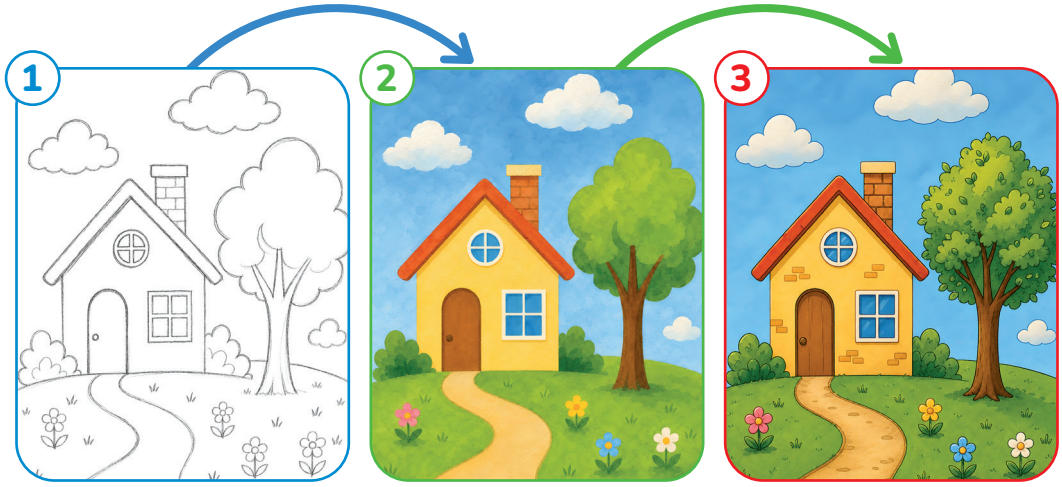


2. Sinfingizda bayram o‘tkazish kerakligini tasavvur qiling. Bunday katta vazifani birdaniga bajarish qiyin, shuning uchun uni kichik harakatlarga ajratish kerak.

- Sinfdoshlaringiz bilan bayramni tashkil qilish uchun nimalar qilish kerakligini muhokama qiling.
- Katta vazifani kichik bosqichlarga ajrating.
- Bosqichlarni to‘g‘ri tartibda yozing.

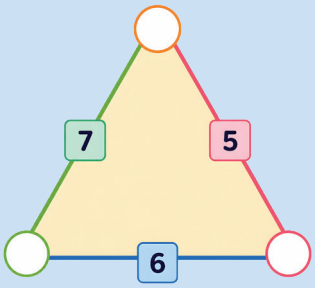


3. Murakkab vazifani oddiy qismlarga ajratadigan holatlarga misollar keltiring. Yordamchi rasmdan foydalaning.



4. Bu vazifani alohida, sodda bosqichlarga qanday ajratish mumkinligi haqida o‘ylab ko‘ring.

Uchburchakning har bir tomonidagi son o‘ziga qo‘shni ikkita doiradagi sonlar yig‘indisiga teng bo‘lishi uchun doiralar ichiga qanday sonlar yozilishi kerak?



Muhokama qiling:

- Nima uchun vazifani bosqichma-bosqich bajarish qulayroq?
- Uni birdaniga bajarish mumkinmi?
- Agar biror bosqichda xato qilinsa, nima bo‘ladi?

“Xanoy minoralari” deb nomlangan mashhur mantiqiy masala bor. Uning nomi Vetnamdagi Xanoy shahri bilan bog‘liq. Bu masala haqida bir afsona mavjud: rohiblar oltin disklarni qat’iy qoidalarga amal qilgan holda bir minoradan boshqa minoraga o‘tkazishar ekan. Ular ishni tugatganida, oxirzamon kelar ekan.

Aslida, bu masalani XIX asrda fransuz matematigi Eduard Lyuka o‘ylab topgan. Bu boshqotirma murakkab vazifani algoritm va dekompozitsiya yordamida asta-sekin, bosqichma-bosqich hal qilish mumkinligini ko‘rsatadi.



5. Savollarga javob bering.

- **Dekompozitsiya** nima?
- Nima uchun murakkab vazifani kichik harakatlarga ajratish qulay?
- Agar murakkab vazifa bosqichlarga ajratilmasa, nima bo‘lishi mumkin?
- Dekompozitsiyaga uydagi yoki maktabdagi ishlaringizdan misol keltiring.

Algoritmni bajarish



- Bir xil buyruqni odamga, robotga va itga berib, bir xil natijaga erishish mumkinmi?
- Nima uchun batafsil tushuntirilganidan keyin ham odam topshiriqni noto‘g‘ri bajarishi mumkin?



1. “Algoritm asosida rasm chizamiz” o‘yinini juft bo‘lib o‘ynang.

A o‘quvchi – algoritm muallifi

- Oddiy rasm o‘ylab toping.
- Uni sherigingizga ko‘rsatmang.
- Rasm chizish uchun aniq buyruqlarni yozing.





B o‘quvchi – ijrochi

- Sherigingizning buyruqlarini diqqat bilan bajaring.
- Qo‘shimcha savollar bermang.

Muhokama qiling:

- Rasm muallif o‘ylaganidek chiqdimi?
- Qaysi buyruqlar yaxshi, qaysilari unchalik yaxshi bo‘lmadi?
- Nima uchun ijrochiga aniq buyruqlar juda muhim?
- Ijrochi harakatni o‘zicha bajarishi mumkinmi?



Ijrochi – buyruqlarni bajara oladigan odam, hayvon, kompyuter, mashina yoki robot. Har bir ijrochi faqat ma'lum buyruqlarni tushunadi.

Algoritm – muayyan ijrochi uchun mo'ljallangan aniq harakatlar ketma-ketligi. Bir xil algoritm bir ijrochiga tushunarli bo'lishi, boshqa ijrochiga esa tushunarsiz bo'lishi mumkin.



It o'tir, oldimga kel buyruqlarini tushunadi.



Pult bilan boshqariladigan mashina faqat oldinga, orqaga, chapga, o'ngga buyruqlarini tushunadi.

2. Rasmga qarang. Rasmdagilardan qaysilari buyruqlarni bajaradi? Ularning har biri qanday buyruqlarni tushunadi?



Ijrochi tushunadigan buyruqlar to‘plami **ijrochining buyruqlar tizimi** deb ataladi. Odam so‘zlar va imo-ishoralarni, kompyuter esa dastur buyruqlarini tushunadi.

Ijrochi algoritmni to‘g‘ri bajarishi uchun buyruqlar quyidagicha berilgan bo‘lishi kerak:

- ijrochi uchun tushunarli;
- aniq;
- to‘g‘ri tartibda.

Agar buyruq noaniq yoki tushunarsiz bo‘lsa, ijrochi topshiriqni to‘g‘ri bajara olmaydi.

Demak, har bir ijrochi uchun o‘zi bajara oladigan alohida algoritm tuzish kerakmi?





Aynan shunday! Ijrochi faqat o‘zining buyruqlar tizimiga kiradigan buyruqlarni tushunadi.



3. Kvadrat chizish algoritmini quyidagi ijrochilar uchun tuzing:
a) har qanday so‘z va imo-ishoralarni tushunadigan odam uchun;
b) “chizishni boshla”, “yuqoriga”, “pastga”, “chapga”, “o‘ngga”, “chizishni tugat” buyruqlarini tushunadigan kompyuter uchun.

Bugungi kunda odamlarga robotlar va kompyuterlar yordam beradi. Robot – insonning buyruqlarini bajaradigan mashina. Robot uchun ham, kompyuter uchun ham algoritm va dasturni to‘g‘ri tuzish juda muhim.

4. Robot uchun tuzilgan algoritmni ko‘rib chiqing. Bu algoritm bajarilganda, tarvuz yashik ichida bo‘ladimi?





5. Savollarga javob bering.

- Kim yoki nima **algoritm ijrochisi** bo‘lishi mumkin? Misollar keltiring.
- Ijrochining buyruqlar tizimi nima deb ataladi?
- Nima uchun ijrochiga beriladigan buyruqlar aniq va tushunarli bo‘lishi kerak? Noaniq buyruq noto‘g‘ri natijaga olib kelishi mumkin bo‘lgan holatga misol keltiring.



Takrorlanuvchi algoritmlar

Hayotimizda ko‘plab harakatlar qayta-qayta takrorlanadi. Shunday harakatlarga misollar keltiring.

Kun va tun qanday almashinadi?

Har kuni ertalab Quyosh chiqadi – kun boshlanadi. Kechqurun Quyosh botadi – tun boshlanadi. Bu harakatlar har kuni takrorlanadi.



Har kuni **ertalab** Quyosh chiqadi – **kun** boshlanadi.

Kechqurun Quyosh botadi – **tun** boshlanadi.

Takrorlanishlarni nafaqat tabiatda, balki insonning harakatlarida ham uchratish mumkin.

Masalan, odam yurganda takrorlanadigan harakatlarni bajaradi: o‘ng oyoq bilan, keyin chap oyoq bilan, so‘ng yana o‘ng oyoq bilan qadam tashlaydi. Bunday takrorlanishlar kundalik hayotda ham uchraydi: yuz yuvishda, ovqatlanishda, ishlashda.

Ba’zan algoritmlarda ayrim harakatlar bir necha marta takrorlanadi. Bunday takrorlanadigan harakatlar ketma-ketligi **sikl** deb ataladi. Siklni o‘z ichiga olgan algoritmlar **siklik algoritmlar** deb ataladi.

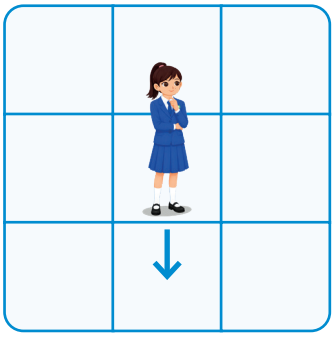


Harakatlar rejasini tuzishda ayrim harakatlarni bir necha marta bajarish kerakligini ko‘rsatishga to‘g‘ri keladi. Bir xil buyruqlarni ko‘p marta yozish o‘rniga, **takrorlashdan** foydalanish mumkin.

1. Algoritmni bajaring:

1) Tik turing.
2) 4 marta takrorlang:
- oldinga bir qadam tashlang;
- o‘ngga buriling.

3) To‘xtang.



- Algoritm oxirida qayerda bo‘lasiz?
- Agar sikldagi qadamlarning o‘rnini almashtirsangiz, ya’ni avval “o‘ngga buriling”, keyin “oldinga bir qadam tashlang” deb buyruq berilsa, qayerda bo‘lasiz?



Algoritmda takrorlashdan foydalanish quyidagilarga yordam beradi:

- algoritm yozuvini qisqartirish;
- algoritmni tushunarliroq qilish;
- bir xil buyruqlarni ko‘p marta takrorlamaslik.

2. 3–4 ta mashqni tanlang, har bir mashqni necha marta bajarish kerakligini belgilang va ertalabki badantarbiya mashqlarini bajarish uchun o‘z algoritmingizni tuzing.

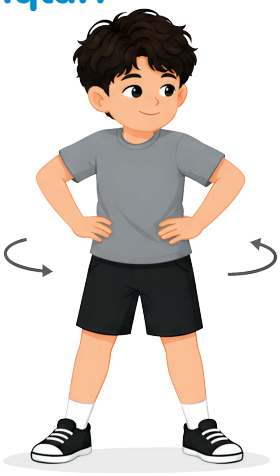
Ertalabki badantarbiya mashqlari



oldinga egilish



o‘tirib-turish



gavdani burish



oldinga tashlanish



qo‘llarga tayanib, gavdani ko‘tarib-tushirish

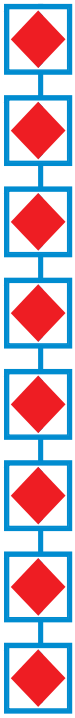


yon tomonga egilish

Siklik algoritmlar kompyuter dasturlarida bir xil harakatni ko‘p marta takrorlash uchun ishlatiladi.

Naqsh chizish algoritmi

1) Boshlanishi.
2) Mo‘yqalamni ko‘k rangga o‘rnating.
3) Ekranning yuqori chetiga o‘ting.
4) 8 marta takrorlang:
   - tomoni 1 cm bo‘lgan kvadrat chizing;
   - mo‘yqalamni qizil rangga o‘rnating;
   - kvadrat ichiga qizil romb chizing;
   - mo‘yqalamni ko‘k rangga o‘rnating;
   - ko‘k kvadratdan pastga uzunligi 5 mm bo‘lgan chiziq chizing;
   - ko‘k chiziqning oxiriga o‘ting.
5) Tugashi.





3. Naqsh chizish uchun o‘z siklik algoritmingizni o‘ylab toping.

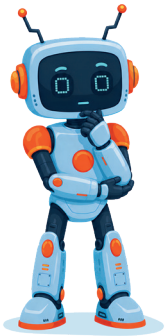


4. Savollarga javob bering.

- **Sikl** deb nimaga aytiladi?
- Qanday algoritmlar siklik algoritmlar deb ataladi?
- Nima uchun **takrorlash** algoritmni soddalashtirishga yordam beradi?
- Hayotda takrorlanadigan harakatlarni qayerlarda uchratish mumkin?



Algoritmdagi xatolarni aniqlash va tuzatish



- Siz ko‘rsatmaga amal qilsangiz ham, natija kutganingizdek bo‘lmay qolishi mumkinmi?
- To‘g‘ri natijaga erishishga ko‘pincha nima xalaqit beradi?

Algoritm tuzishda uning to‘g‘ri ishlashiga ishonch hosil qilish muhim. Buning uchun algoritm tuzuvchilar algoritmdagi barcha harakatlarni o‘zlari ketma-ket bajarib chiqadilar.

Muhokama qiling.

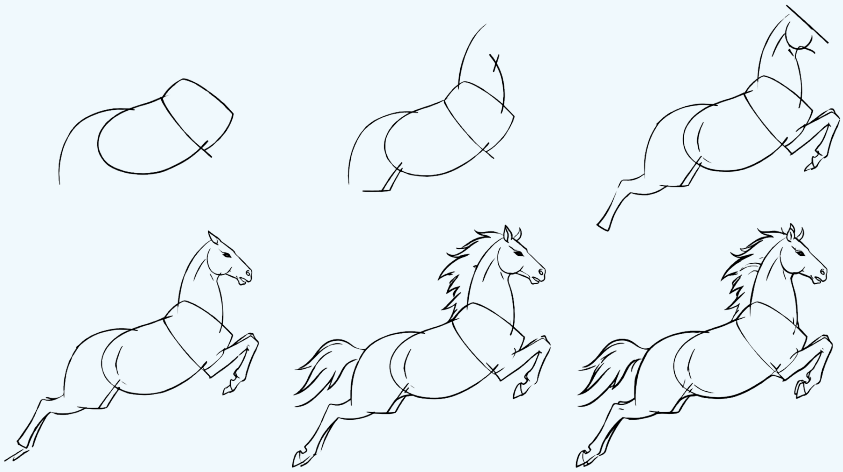
Bu ko‘rsatma bo‘yicha ovozli xabar yuborish mumkinmi? Nima uchun?



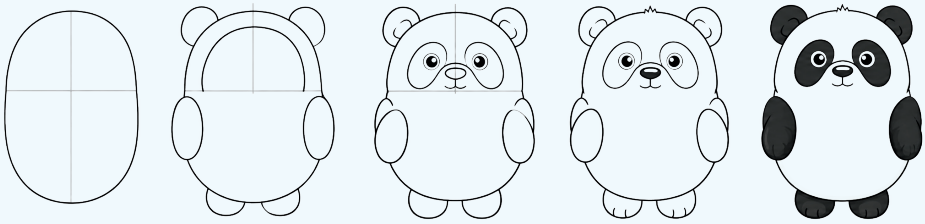


1. Algoritmlarni berilgan qadamlar asosida bajaring. Sizda ham xuddi shunday natija hosil bo‘ldimi?

1



2



Ba’zan algoritm noto‘g‘ri ishlaydi. Bunga harakatlar tartibi buzilgani, buyruqlar chalkashib ketgani yoki muhim qadam tushirib qoldirilgani sabab bo‘lishi mumkin.

Agar harakatlar noaniq yozilgan bo‘lsa ham, algoritm noto‘g‘ri ishlashi mumkin.



Algoritm to‘g‘ri bajarilishi uchun uni tekshirish va xatolarni tuzatish kerak. Bunday jarayon **algoritmdagi xatolarni aniqlash va tuzatish** deb ataladi.

Hatto kichik xato ham noto‘g‘ri natijaga olib kelishi mumkin. Barcha ishni qaytadan bajarmaslik uchun xatoni oldindan topgan ma’qul.

2. Barcha xatolarni toping va bodroq tayyorlashning to‘g‘ri algoritmini tuzing.

1



2



3



4



5



6



7



8



9





3. Tadqiqot o‘tkazing.

- Kichik algoritm o‘ylab toping va yozing.
- Unga ataylab xatolar kiriting.
- Xatolar kiritilgan algoritmingizni sinfdoshingiz bilan o‘zaro almashing.
- Sinfdoshingizning algoritmini bajarishga harakat qilib ko‘ring.
- Algoritmdagi barcha xatolarni toping, ularni tuzating va algoritmning to‘g‘ri variantini yozing.
- Natijalarni muhokama qiling: qaysi xatolarni topish osonroq, qaysilarini topish qiyinroq bo‘lganini ayting.



Hatto tajribali dasturchilar ham ba’zan xatoga yo‘l qo‘yadilar. Shuning uchun dasturlar ishlatilishidan oldin ko‘p marta tekshiriladi va sinovdan o‘tkaziladi.

Dasturlardagi xatolar “bag” deb ataladi. “Bag” so‘zi ingliz tilidan tarjima qilinganda “qo‘ng‘iz” degan ma’noni anglatadi. Tarixiy ma’lumotlarga ko‘ra, eski kompyuter ichiga bir kuni rostdan ham kichkina kuya kirib qoladi va uning ishlashiga xalaqit beradi.

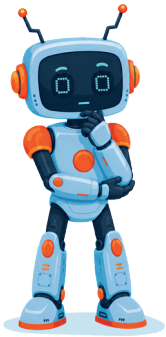


4. Savollarga javob bering.

- Algoritmdagi **xatolarni aniqlash va tuzatish** deganda nimani tushunasiz?
- Algoritmlarda qanday xatolar uchrashi mumkin?
- Nima uchun harakatlar tartibiga rioya qilish muhim?
- Algoritmni qanday tekshirish mumkin?



“Scratch” dasturlash muhiti



Insonni kompyuterlar, robot-changyutkichlar, kir yuvish mashinalari, mobil telefonlar va boshqa ko‘plab avtomatlashtirilgan qurilmalar o‘rab turadi. Bu qurilmalar buyruq va ko‘rsatmalarni qanday tushunadi?

Har bir avtomatlashtirilgan qurilmaning ichida algoritmlarni juda aniq va tez bajara oladigan kichik kompyuter joylashgan.

Kompyuter tushunadigan tilda yozilgan algoritm **dastur** deb ataladi.

Kompyuter tushunadigan maxsus til **dasturlash tili** deb ataladi.



Kompyuter dasturlarini dasturlash tillarini biladigan mutaxassislar yaratadi. Bunday mutaxassislar **dasturchilar** deb ataladi.

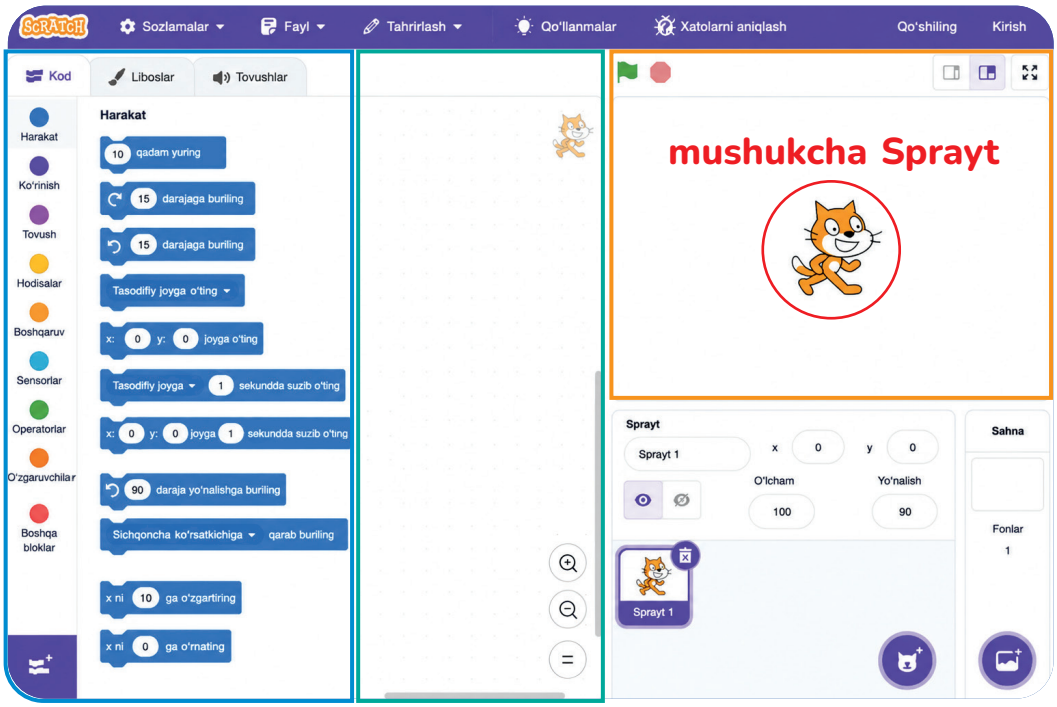
Scratch dasturlash tillaridan biridir. **Scratch** dasturida buyruqlarni qo‘lda yozish shart emas. Dastur pazl yoki konstruktor kabi turli rangdagi bloklardan yig‘iladi.

Scratch tilida dastur yozish uchun **Scratch** dasturlash muhitini ochish kerak.

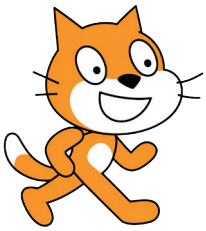
Scratch dasturining oynasi uchta sohaga bo‘lingan:

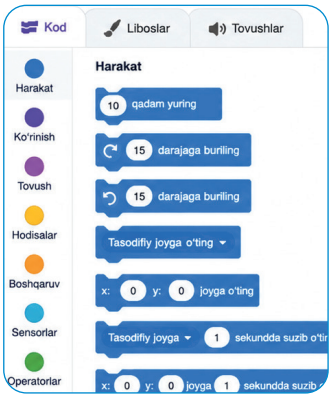
- Buyruqlar;
- Ishchi soha;
- Sahna.

Buyruqlar **Ishchi soha** **Sahna**



Scratch muhitidagi qahramonlardan biri – **Sprayt** ismli mushukcha. Qahramonlar o‘z-o‘zidan harakat qila olmaydi. Ular faqat biz beradigan buyruqlarni bajaradi.





Buyruqlar bloki

Qahramonlar tushunadigan barcha buyruqlar chap tomondagi **Buyruqlar blokida** joylashgan.

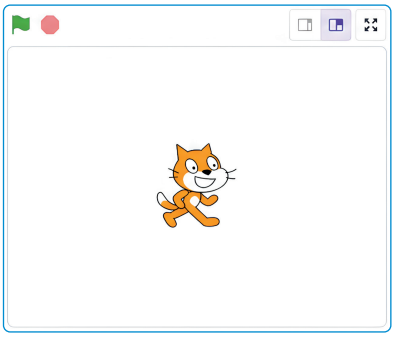
Buyruqlar ranglariga ko‘ra guruhlangan:
ko‘k rangdagilar – harakat buyruqlari;
binafsharangdagilar – ko‘rinish buyruqlari;
pushti rangdagilar – tovush buyruqlari;
sariq rangdagilar – hodisa buyruqlari.

Bu buyruqlardan qahramonlar uchun algoritmlar tuzamiz. Buning uchun buyruq bloklarini **Ishchi sohaga** olib o‘tib, ularni to‘g‘ri tartibda joylashtirish kerak.

Ishchi sohada yozilgan algoritmlar **skriptlar** deb ataladi.



Skript



Sahna

Sahna – qahramoningiz siz bergan buyruqlarni bajaradigan joy. Dastlab Sahnada Sprayt ismli mushukcha bo‘ladi.

Dasturni ishga tushirish uchun **Sahna** oynasida maxsus tugmalar mavjud:

– skriptni ishga tushirish

– skriptni to‘xtatish

Sprayt – buyruqlarni bajaradigan qahramon.
Skript – bitta Sprayt uchun bloklardan yig‘ilgan dastur.
Sahna – barcha jarayonlar sodir bo‘ladigan fon.



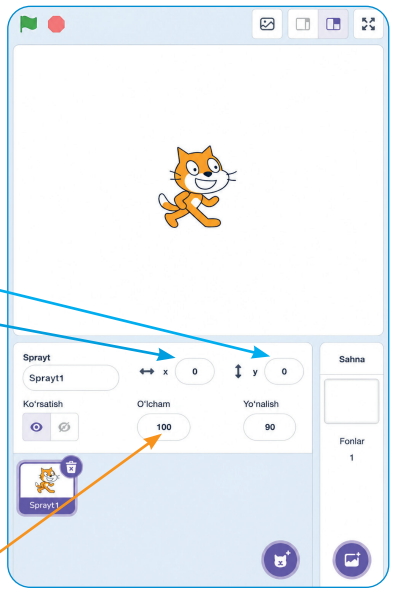
1. **Scratch** dasturini oching.

- Qahramonning nomini o‘zgartiring. Buning uchun Sahna ostidagi sozlamalar panelini toping, uning nomi ustiga sichqoncha bilan bosing va **Mushuk** deb yozing.
- Qahramonning joylashuvini (x = 0; y = 0) va o‘lchamini (200) belgilang.

Qahramon qayerda paydo bo‘ldi? U qanday o‘zgardi?

Qahramonning joylashuvi ikkita koordinata bilan belgilanadi: gorizontal yo‘nalishdagi qiymat (*x* o‘qi) va vertikal yo‘nalishdagi qiymat (*y* o‘qi). Sahna markazining koordinatalari (0; 0).

Qahramonning o‘lchami ekrandagi nuqtalar – **piksellarda** belgilanadi.



Obyektni harakatlantirish uchun u harakatlanishi kerak bo‘lgan hodisani ko‘rsatish kerak.

Birinchi hodisa skript ishga tushirilganda sodir bo‘ladi. Shuning uchun obyekt boshidanoq harakatlana boshlashi uchun  **bosilganda** buyrug‘ini **Buyruqlar blokidan Ishchi sohaga** olib o‘tish kerak. Buyruqni olib o‘tish uchun buyruq ustiga sichqonchaning chap tugmasini bosib, tugmani qo‘yib yubormagan holda uni **Ishchi sohaga** olib o‘tish kerak.

Kerakli buyruq qaysi ro‘yxatda joylashganini qanday bilaman?





Bunda buyruqning rangi yordam beradi. Agar buyruq biror hodisa bilan bog‘liq bo‘lsa, u **sariq** rangda, agar harakat bilan bog‘liq bo‘lsa, **ko‘k** rangda bo‘ladi.



2. Ishchi sohaga quyidagi buyruqlarni olib o‘ting:

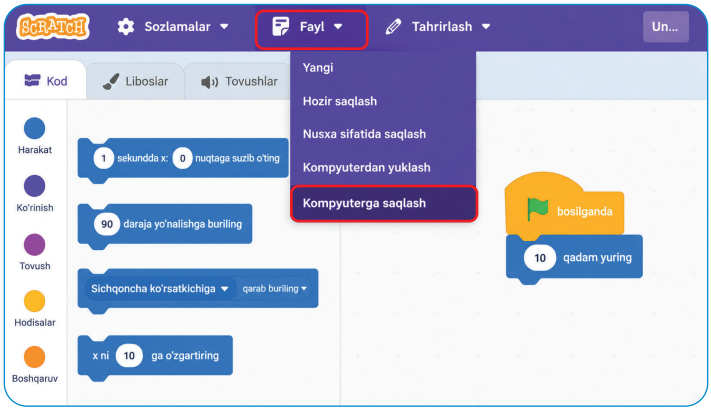
-  bosilganda;
- 10 qadam yuring.



Skriptni ishga tushiring. Nima sodir bo‘ldi?

Ikkinchi buyruqdagi qadamlar sonini o‘zgartirish mumkin. Buning uchun sichqoncha ko‘rsatkichini 10 soni ustiga olib boring. Bunda ko‘rsatkich qo‘lcha belgisi o‘rniga tik chiziq shakliga o‘zgaradi. So‘ng sichqonchaning chap tugmasini bosib, klaviatura yordamida qadamlar sonini o‘zgartiring.

Skript kompyuter xotirasida saqlanib qolishi uchun uni saqlash kerak. Menyudan **Fayl** tugmasini bosing va paydo bo‘lgan ro‘yxatdan **Kompyuterga saqlash** bandini tanlang.



3. **Scratch** dasturida rasmdagi kabi skript yarating.



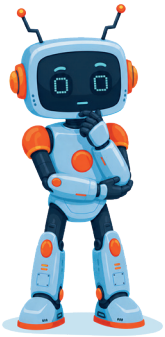
- Skript ishga tushirilgandan so‘ng nima sodir bo‘ladi?
- Qahramonga yangi nom o‘ylab toping va uning nomini o‘zgartiring.
- Qahramon boshqa burchakka burilishi uchun algoritmni o‘zgartiring.

4. Qahramon kvadrat bo‘ylab harakatlanadigan skript tuzing. Skriptingizni “Burilish” nomi bilan saqlang.

5. Savollarga javob bering.

- **Scratch** nima?
- **Sprayt** qanday qahramon?
- **Skript** deb nimaga aytiladi?
- **Sahna** nima uchun kerak?
- **Scratch** dasturida qanday buyruqlar guruhi mavjud?

Spraytlar va obyektlarni boshqarish



- Ko‘pincha o‘yinlar va multfilmlarda bitta qahramon emas, balki bir nechta qahramon ishtirok etadi.
- Qahramonlar qanday qilib birgalikda harakat qilishi mumkinligini o‘ylab ko‘ring.

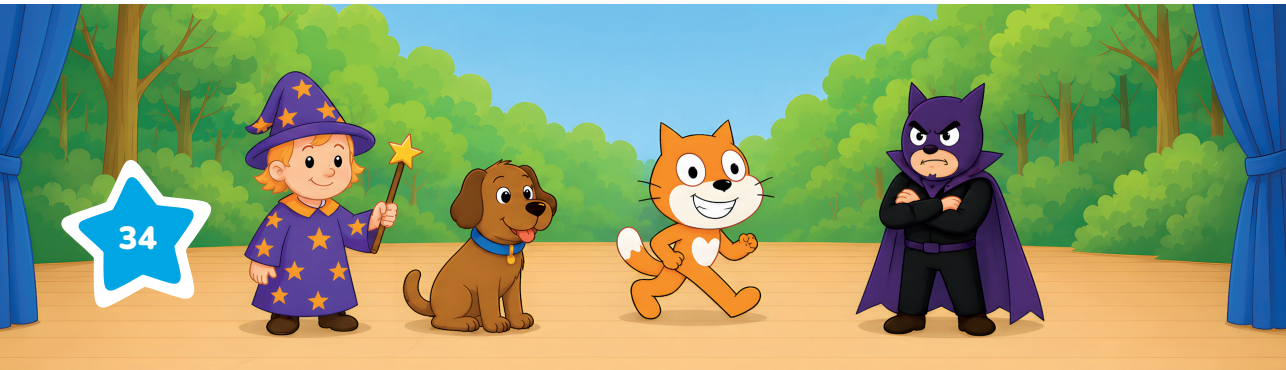
Haqiqiy teatrni tasavvur qiling. Sahnada turli aktyorlar: bosh qahramon, uning do‘stlari va yordamchilari, yovuz qahramonlar, hayvonlar, sehrgarlar chiqish qiladi. Har bir aktyor o‘z rolini ijro etadi va boshqa qahramonlar bilan birgalikda harakat qiladi.

Scratch dasturida bunday teatr tomoshasi “loyiha” deb ataladi: unda barcha qahramonlar, ularning skriptlari va Sahnasi jamlangan bo‘ladi. Loyihani saqlash va keyin yana ochib, ishni davom ettirish mumkin.

Sprayt – harakatlanishi, gapirishi, raqsga tushishi va buyruqlarni bajarishi mumkin bo‘lgan istalgan qahramon. Scratch dasturida mushukcha Spraytidan tashqari boshqa ko‘plab qahramonlar ham bor. Bu qahramonlarning barchasi bir vaqtning o‘zida bitta Sahnada bo‘lishi mumkin.

Har bir Sprayt:

- o‘z hayoti bilan yashaydi;
- o‘z skriptlar to‘plamiga ega;
- boshqa Spraytlar bilan birgalikda harakat qila oladi.

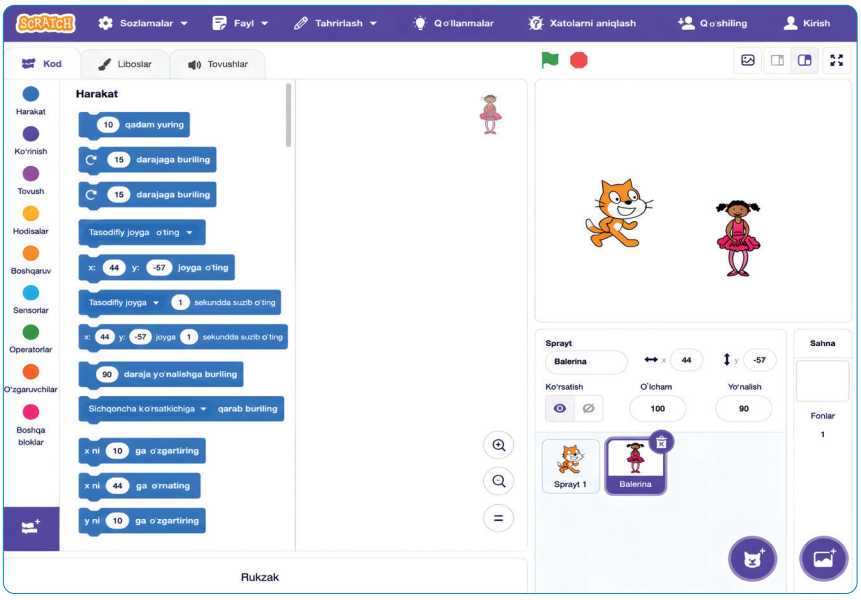


Yangi Sprayt qo‘shish

Sahnaga yangi **Sprayt** qo‘shish uchun ekranning pastki qismidagi **“Spraytni tanlash”** tugmasi bosiladi. Shundan so‘ng **Spraytlar** kutubxonasi ochiladi. Undan yangi **Sprayt** tanlanadi.



Shu tariqa **Sahnaga** yangi qahramon qo‘shiladi.



Pastdagi ro‘yxatda yangi **Sprayt** paydo bo‘ladi.

Har bir Sprayt uchun skript alohida yoziladi.

Kerakli qahramonning skriptini yozish uchun avval ekranning pastki qismidagi ro‘yxatda uning ustiga bosing.

Agar **Spraytlar** aktyorlar bo‘lsa, **Sahna** barcha voqealar sodir bo‘ladigan joydir. Biz qahramonlarni Sahnaga qo‘shishni va ularga buyruqlar to‘plamini berishni o‘rgandik.

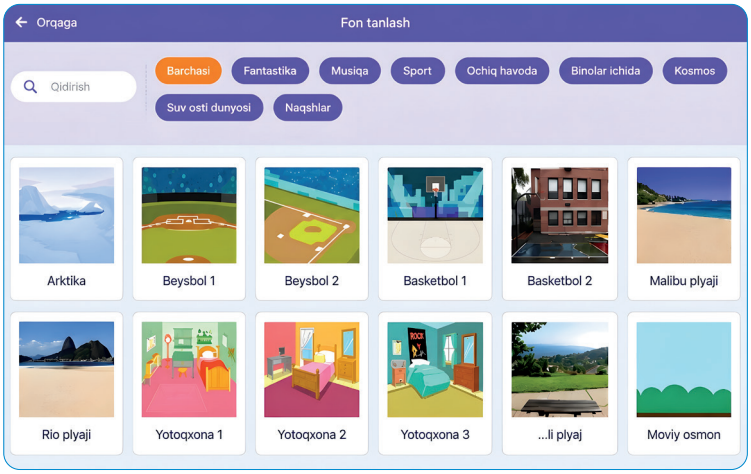
Sahnani teatr dekoratsiyalariga qiyoslash mumkin. U turli fonlar yordamida o‘z ko‘rinishini o‘zgartira oladi.

Scratch dasturida **Sahnani** quyidagicha o‘zgartirish mumkin:

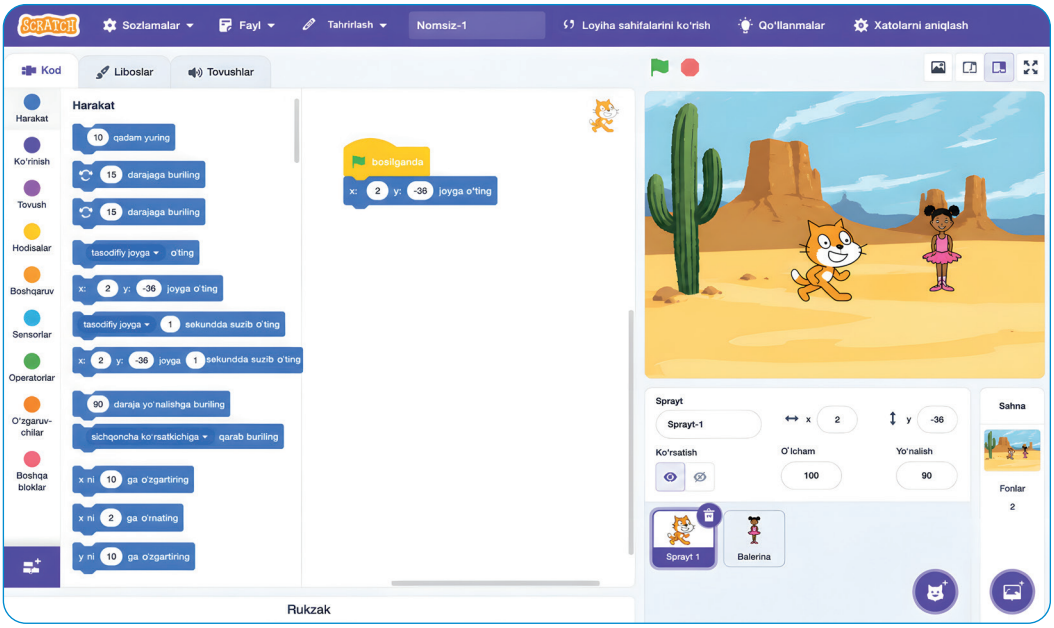
- fonni o‘zgartirish (xona, o‘rmon, kosmos, maktab, dengiz va hokazo);
- **Sahnada** bir nechta fonni saqlash va dastur davomida ularni almashtirish;
- **Sahnaning** o‘zi ham buyruqlarni bajarishi mumkin (fonni o‘zgartirish, musiqa ijro etish).

Sahna bilan ishlash

Sahna fonini o‘zgartirish uchun ekranning pastki o‘ng qismidagi **“Sahna”** belgisini bosing. Shundan so‘ng fonlar kutubxonasi joylashgan oyna ochiladi. Undan kerakli fonni tanlang.



Shunda **Sahna** ko‘rinishi o‘zgaradi.





Bitta Scratch loyihasiga 1000 tagacha Sprayt qo‘shish mumkin, ammo dastur tez ishlashi uchun 10–15 tadan ko‘p Sprayt ishlatmagan ma’qul.



Savollarga javob bering.

- Bitta **Sahnada** nechta **Sprayt** bo‘lishi mumkin?
- Turli Spraytlarning **skriptlari** qayerda joylashadi?
- Yangi Spraytni qanday qo‘shish mumkin?



8-dars. 1-nazorat ishi

9-dars. “Scratch dasturida multfilm yaratish” loyihasi

Biz bilib oldik:

algoritm kerakli natijaga olib keladigan aniq qadamlar ketma-ketligi ekanini;

algoritm qadami algoritmdagi har bir alohida harakat ekanini;

dekompozitsiya murakkab vazifani sodda qismlarga ajratish ekanini, bu to‘g‘ri ketma-ketlikni tuzish va algoritm yaratishga yordam berishini;

inson, robot, kompyuter va Sprayt kabilar algoritmni bajaradigan **ijrochi** ekanini, har bir ijrochining o‘z buyruqlar tizimi mavjudligini va shuning uchun buyruqlar aniq va tushunarli bo‘lishi kerakligini;

bir xil harakatlarning bir necha marta takrorlanishi **sikl** deb atalishini;

algoritmdagi xatolarni aniqlash va tuzatish mumkinligini;

Scratch – dasturlar rangli bloklardan tuziladigan blokli dasturlash muhiti ekanini;

Scratch dasturidagi buyruqlarni bajaradigan qahramon **Sprayt** deb atalishini.