|  |
| --- |
| **Dasturlash tushunchalari va dasturlashni rivojlantirish** |
| **Muxtorov Shaxzodbek** |

**4-qism: Programming**

**O’qituvchi:** Tursunboev Otabek

**Guruh IDsi:** 23-228

**Talaba IDsi: 230933**

**Topshirish sanasi:** 17.01.2024

**BTEC Learner Assessment Submission and Declaration**

Har bir o’quvchi vazifani baholash uchun taqdim etganda, yozilgan deklaratsiya o’ziniki ekanligini tasdiqlovchi imzo qoldirishi kerak.

|  |  |
| --- | --- |
| **O’quvchi (Talaba) IDsi:** | 230933 |
| **Baholovchining ismi:** | Tursunboev Otabek |
| **BTEC Dasturining mavzusi:** | Pearson BTEC 4-darajali Programming |
| **Qism yoki Komponent raqami va Mavzu:** | 4-qism : Programming |
| **Vazifa mavzusi:** | Dasturlash tushunchalari va dasturlashni rivojlantirish |
| **Vazifani topshirish sanasi:** | 17.01.2024 |

Iltimos, har bir topshiriq uchun taqdim etilgan dalillarni sanab o'ting. Dalillarni topish mumkin bo'lgan sahifa raqamlarini ko'rsating yoki dalillarning mohiyatini tavsiflang (masalan, video, rasm).

|  |  |
| --- | --- |
| **Topshiriq manbalari** | **Taqdim etilgan dalillar** |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| **O’quvchi deklaratsiyasi**  Ushbu vazifa uchun topshirilgan ish meniki ekanligini tasdiqlayman. Ishda foydalanilgan manbalarga aniqlik kiritdim va ko’rsatib o’tdim. Noto’g’ri deklaratsiya noto’g’ri ishlashning bir shakli ekanligini tushunaman.  **Talabaning imzosi:** Screenshot 2023-12-03 at 21.56.14 **Sana:** 17.01.2024 |

Mundarija

[1.0. Kirish 3](#_Toc129600062)

[2. Topshiriq 1 3](#_Toc129600063)

[3. Topshiriq 2 1](#_Toc129600064)2

[4. Xulosa 2](#_Toc129600067)1

[Manbalar ro’yxati 2](#_Toc129600068)2

1.0. Kirish

**Men Muxtorov Shaxzodbek va men kichik dasturiy ta’minot ishlab chiqaruvchi kompaniya hodimiman. B**ilim bazasini yaratish uchun qo'llanilgan tushunchalarni o'rganish va tashrifga tayyorgarlik ko'raman va ballarni hisoblovchi kichik dastur tuzaman. Bunda men dasturdagi Hisoblash fikrlash (Computational thinking) ko'nikmalari va kompyuter dasturlash tamoyillarini yordam beradi va men mijoz talablariga mos keladigan dasturiy yechimni loyihalash, mijoz talablariga mos keladigan dasturiy yechimni ishlab chiqaman.

2. TOPSHIRIQ 1

Topshiriq 1 **Learning Aim A** ni qamrab oladi.

****A.P1**** Hisoblash fikrlash ko'nikmalari qanday qilib dasturiy ilovalarga aylantirilishi mumkin bo'lgan yechimlarni topishda qo'llanilishi haqida tushuntirish.

Computational thinking (CT) - bu dasturiy injineringda ishlatiladigan muammolarni hal qilish va usullarini ishlatishga yordam beradigan ko'nikmalar to'plamidir. CT muammolarni tushunish va mumkin bo'lgan yechimlarni ishlab chiqishga yordam beradi. So'ngra, ushbu yechimlarni kompyuter, odam yoki ikkalasi tushunishi mumkin bo'lgan shaklda taqdim etish mumkin.

Computational thinkingning to'rtta asosiy usullari bor:

1. Decomposition (Decomposition): Muammolarni kichikroq qismlarga bo'lish.

2. Pattern recognition (Pattern recognition): Ushbu kichikroq qismlar ichida namunalarni qidirish.

3. Abstraction (Abstraction): Muhim ma'lumotlarga e'tibor berish va noo'rin ma'lumotlarni e'tibor tashlash.

4. Algorithms (Algorithms): Kichikroq muammolarni hal qilish uchun qadam-qadam yechimni yoki qoidalarini topish.

Bu usullar muammolarni hal qilishda juda samarali. Chunki ular murakkab muammolarni boshqariladigan qismlarga bo'lishga yordam beradi.

Misol uchun, siz dastur tuzdingiz va dastur ishga tushmaydi. Bu murakkab muammo. Lekin computational thinking usullarini qo'llagan holda, siz bu muammoga quyidagicha yondashasiz:

1. Decomposition: Muammolarni kichikroq qismlarga bo'lish. Masalan, funksiyalarnini tekshirish, o’zgaruvchi va shart operatorlarni tekshirish, xatoliklarni tekshirish, va hokazo.

2. Pattern recognition: O'xshash muammolarni hal qilishda ishlatilgan yechimlarni qidirish. Masalan, dasturlar odatda baglar yoki noto’gri struktura sababli ishga tushmaydi.

3. Abstraction: Faqatgina muhim ma'lumotlarga e'tibor berish. Masalan, sizga uni kim ishlatishi yoki sizning dasturingizni nechta kishi foydalanishi mumkinligi haqida ma'lumot kerak emas.

4. Algorithms: Muammolarni hal qilish uchun qadam-qadam yechimni yoki qoidalarini topish. Masalan, Logikani tekshirish, keyin strukturani tekshirish, keyin tartibini va baglarni tekshirish, va hokazo.

1. Hisoblash fikrlash - bu kompyuterlar tomonidan amalga oshirilishi mumkin bo'lgan usullar bilan muammolarni qayta formulatsiya qilish va hal qilish qobiliyatidir. Hisoblash fikrlash ko'nikmalari, masalan, dekompozitsiya, abstraksiya va algoritm ishlab chiqish kabi tushunchalar, murakkab muammolarni sodda qilish, muhim qismlarga e'tibor berish va dasturlash uchun aniq yo'riqlarni belgilash orqali yuqori sifatli dasturiy ta'minot yaratishda muhim rol o'ynaydi.

2. Amaliy hisoblash fikrlash jarayonlari dasturiy ta'minotning sifatini aniq rag'batlantiradi (umumiy hisoblash fikrlash qobiliyati va noaniq aql o'ylash rag'batlantiruvchi emas). Bu natijalar talabalar uchun hisoblash fikrlashning amaliy jarayonli qobiliyatlarini rivojlantirishning umumiy hisoblash fikrlash qobiliyatlaridan ko'proq ahamiyatga ega ekanligini ko'rsatadi.

3. Bu tadqiqot, hisoblash fikrlash ko'nikmalarini egallashning ko'plab hayot sohalarida foydalanish mumkin bo'lgan usul bo'lishi mumkinligini ko'rsatadi.

Bu usullar dasturiy ta'minotning sifatini oshirishga yordam beradi, chunki ular muhandislarga muammolarni yaratuvchi va ijodiy usullar bilan hal qilishga imkon beradi. Bu, dasturiy ta'minotning samaradorligini oshiradi va foydalanuvchilar uchun yanada yaxshi tajriba yaratadi.

****A.P2**** Turli dasturlash tillarida dasturlash tamoyillari qanday qo'llanilishi va dasturiy ilovalarni ishlab chiqarish haqida tushuntirish.

Dasturlash tillarining xususiyatlari va xarakteristikalarini o'rganish juda muhimdir. Har bir dasturlash tili o'ziga xos xususiyatlarga ega bo'lib, ularning har biri muayyan turlardagi muammolarni hal qilishga yordam beradi.

Misol uchun, **Python** va **C++** dasturlash tillarini taqqoslash mumkin:

**Python**:

Python dasturlash tili yordamida foydalanuvchidan kiritishni olish mumkin(3)(4). Python 3.6-da input() metodi ishlatiladi, Python 2.7-da esa raw\_input() metodi ishlatiladi.

Python dasturlash tili yordamida for va while sikllarini ishlatib, foydalanuvchidan kiritishni takrorlash mumkin(3).

Python dasturlash tili yordamida, foydalanuvchidan kiritishni olish va uni tekshirish mumkin.

**C++**:

C++ dasturlash tili yordamida, cin o'zgaruvchisini ishlatib foydalanuvchidan kiritishni olish mumkin.

Har bir dasturlash tili o'ziga xos xususiyatlarga ega bo'lib, ularning har biri muayyan turlardagi muammolarni hal qilishga yordam beradi. Bu xususiyatlar dasturlash tili tanlashda muhim ahamiyatga ega(5).

Dasturlash tamoyillari har bir dasturlash tilida o’zgarishi mumkin, lekin asosiy g’oya barcha tillarda ham xuddi shu. Bu koddagi LibraryManagementSystem klassi Python dasturlash tilida yozilgan va bu tilning tamoyillarini aks ettiradi.Python dasturlash tilida, klasslar obyektga yo’naltirilgan dasturlashning asosiy qismi hisoblanadi. Klasslar ma’lumotlarni va funktsiyalarni birlashtiradi. Bu yerda LibraryManagementSystem klassi kutubxona boshqarish tizimini boshqarish uchun metodlarni taqdim etadi.Har bir dasturlash tilida, dasturlash tamoyillari va dasturiy ilovalarni ishlab chiqarishning o’ziga xos usullari mavjud. Masalan, JavaScript dasturlash tilida, siz odatda asinxron operatsiyalar bilan ishlash uchun “callback” funksiyalardan yoki "promise" lardan foydalanasiz. Java dasturlash tilida, siz odatda klasslarni va interfeyslarni ishlatasiz, chunki bu til to’liq obyektga yo’naltirilgan.Dasturlash tamoyillari va dasturiy ilovalarni ishlab chiqarishning eng yaxshi usullari haqida ko’proq bilish uchun, sizga biror dasturlash tilida yuqori darajadagi kurslarni o’tkazishni tavsiya etaman. Bu kurslar sizga dasturlash tamoyillari, dasturiy ilovalarni ishlab chiqarishning eng yaxshi usullari, kodning o’qiladiganligi, kodning qayta ishlatiladiganligi va kodning kengaytiriladiganligi kabi muhim mavzular haqida ko’proq tushuncha beradi.  
****A.P3**** Hisoblash fikrlash ko'nikmalari dasturiy dizayn va ishlab chiqarilgan dasturiy ilovalarning sifatiga qanday ta'sir qilishi mumkinligini tahlil qilish.

1. Dastur dizayniga mantiqni qo'llash: Dasturiy ta'minot modulini ishlab chiqishda birinchi qadam, asosan, pastdan dastur tuzilmasini qo'shni nazorat qilishdan iborat: modul spetsifikatsiyasini o'rganish orqali ishlab chiquvchi uning o'zi uchun tushunarli ekanligiga va ushbu modulni ishlab chiqish uchun etarli ekanligiga ishonch hosil qilishi kerak.

2. Matematik va taklif mantiqining tamoyillari: Matematik mantiq, diskret matematika va algoritm tushunchalarini o'rganish juda muhimdir. Bu, dasturlashda muhim ahamiyatga ega bo'lgan logikani, shartlarni va qarorlar qabul qilishni tushunishga yordam beradi.

3. To'plamlar va iteratsiya foydalanish: To'plamlar va ular ustida amallar bajarish, dasturlashda muhim ahamiyatga ega. Bu, ma'lumotlarni tuzish, qidiruv va tahlil qilish uchun kerak bo'lgan asosiy asboblar. Iteratsiya esa, biror amalni takrorlash uchun ishlatiladi, bu dasturlashda juda odatiydir.

**A.M1** Hisoblash fikrlash ko'nikmalari dasturiy dizayn va ishlab chiqarilgan dasturiy ilovalarning sifatiga qanday ta'sir qilishi mumkinligini tahlil qilish.

lib\_management.py faylida yozilgan kodda “Library Management System” nomli dastur yaratilgan, uning maqsadi kutubxona intervalini boshqarish. Dasturda quyidagi asosiy funksiyalar mavjud:

1. Kutubxonaga kitob qo’shish (“add\_book” method).
2. Kutubxonadan kitob olib tashlash(“remove\_book” method).
3. Kitob ma’lumotlarini yangilsh(“update\_book” methot).
4. Kitobni vaqtincha olib turish (“loan\_book” methot).
5. Kitobni qaytarish (“return\_book” methot).
6. Barcha kitoblar ro’yxatini ko’rish(“view\_all\_book” methot).

Dasturda LibraryManagementSystem nomli klass yaratilgan, uncha barcha kitoblar ma’lumotlarini saqlash uchun books nomli lug’at (dictionary) ishlatilgan. Har bir kitob uchun har xil ID berilgan va kitob ma’lumotlari yani nomi, muallifi, qaytarilganligi yoki qo’shilganligi saqlanadi. Barcha asosiy funksiyalar “main” metodda chaqiriladi va foydalanuvchi menyudan tanlash orqali dastur bajariladi. Dasturda “LibraryManagementSystem” nomli klass tuzilgan. Bunda obyektlar foydalanuvchi tomonidan qo’shiladi hamda ular “books” lug’atida saqlanadi. Bu dasturni tuzish va boshqarishda qulaylik yaratiladi. Barcha funksiyalar klassning metodlari sifatida yaratilgan. Bu dastur logikasini tuzish va har bir bir boshqa funksiyaga o’zgartirish/qo’shish imkonini beradi. Foydalanuvchiga tanlash imkoniyatini beruvchi menyu “display\_menu” nmatodida yig’ilgan. Foydalanuvchi tanlagan har bir buyruq uchun mos funksiya chairiladi. Ular sizga yangi istiqbollar bilan muammolarni yengishda yordam beradi, bu esa yanada samarali va innovatsion qarorlarga olib keladi.

Ular ilm-fan, texnologiya, san'at yoki biznes sohalarida innovatsiyalarni boshqaradi. Bu yangi ixtirolar va yangi g‘oyalar ortida turgan vosita.

Bizning tez sur'atlar bilan rivojlanayotgan dunyomizda moslashish juda muhimdir. Ijodiy fikrlash qobiliyatlari sizga ko'proq moslashishga yordam beradi, chunki ular o'zgarishlarga duch kelganda yangi g'oyalar va strategiyalarni ishlab chiqishi mumkin.

Professional dunyoda ijodiy fikrlash yuqori baholanadi. Bu martaba o'sishiga olib kelishi mumkin, chunki u odamlarga ajralib turishga, murakkab muammolarni hal qilishga va tashkilotlarda innovatsiyalarni rivojlantirishga imkon beradi. Ijodkorlik tajribani boyitish, o'zini namoyon qilishni rivojlantirish va dunyoni ko'rish uchun noyob ob'ektivni taklif qilish orqali hayot sifatini oshiradi.Ochiq fikrlash: Ijodiy mutafakkirlar yangi g'oyalar va qarashlarni hukm qilmasdan qabul qilib, innovatsiyalar rivojlanishi mumkin bo'lgan muhitni yaratadilar.

Qiziquvchanlik: Ularda yangi bilim va tajribalarni o'rganishga, savol berishga va izlashga undaydigan tabiiy qiziqish bor.

Tasavvur: Ijodiy mutafakkirlar jonli tasavvurga ega bo'lib, ularga odatdagidan tashqari imkoniyatlarni tasavvur qilish imkonini beradi.

Tavakkalchilik: Muvaffaqiyatsizlik muvaffaqiyat sari pog'ona bo'lishi mumkinligini bilib, ular hisoblangan tavakkal qilishdan qo'rqmaydilar.

Moslashuvchanligi: Ijodiy shaxslar o'zgaruvchan sharoitlarga moslashadi va kerak bo'lganda o'z strategiyalari yoki g'oyalarini o'zgartirishga tayyor.

Qat'iylik: Ular o'zlarining ijodiy maqsadlariga erishish uchun qiyinchiliklar va muvaffaqiyatsizliklarni engib, qat'iyat va qat'iyatni namoyish etadilar.

Hamkorlik: Ijodiy mutafakkirlar ko'pincha boshqalar bilan hamkorlik qiladilar va turli nuqtai nazarlar yanada innovatsion echimlarga olib kelishi mumkinligini tan oladilar.

Muammoni hal qilish: ishlab chiqarish jarayonida samaradorlikni oshirish uchun yangi yondashuvni topish.

Innovatsiya: kengroq auditoriyani jalb qilish uchun yangi marketing kampaniyasini ishlab chiqish.

Moslashuvchanlik: noyob mahsulot qatorini yaratish orqali mijozlarning afzalliklaridagi o'zgarishlarga moslashish(1).

Hisoblash fikrlash ko'nikmalari, dasturiy dizayn va ishlab chiqarilgan dasturiy ilovalarning sifatiga katta ta'sir qiladi.

****A.D1**** Hisoblash fikrlash ko'nikmalari dasturiy dizayn va ishlab chiqarilgan dasturiy ilovalarning sifatiga qanday ta'sir qilishi mumkinligini baholash.

Dastur dizayniga mantiqni qo'llash, matematik va taklif mantiqining tamoyillarini, to'plamlar va iteratsiya foydalanish juda muhimdir. Bu borada bir nechta muhim nuqtalar bor:

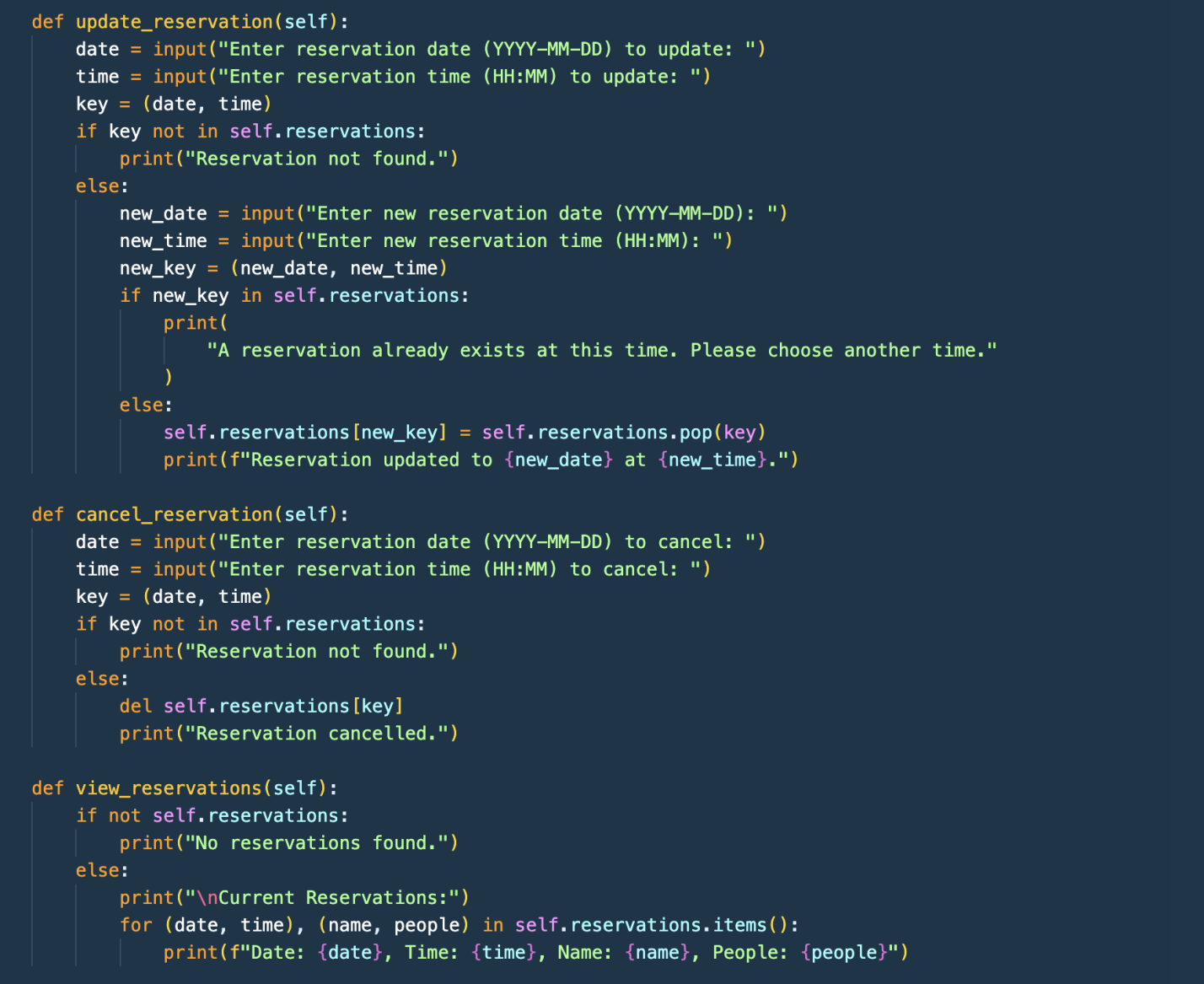
Bu borada, sizning so'rovingizga javob berish uchun qo'shimcha ma'lumotlarni topish uchun qanday qilib qidirishni bajarishim kerakligini tushunishim kerak. Bu, sizning so'rovingizni eng yaxshi qanday qilib tahlil qilishim kerakligini aniqlashga yordam beradi. Bu borada, sizning so'rovingizga javob berish uchun qanday qilib qidirishni bajarishim kerakligini tushunishim kerak. Bu, sizning so'rovingizni eng yaxshi qanday qilib tahlil qilishim kerakligini aniqlashga yordam beradi (6).



A.1-rasm.

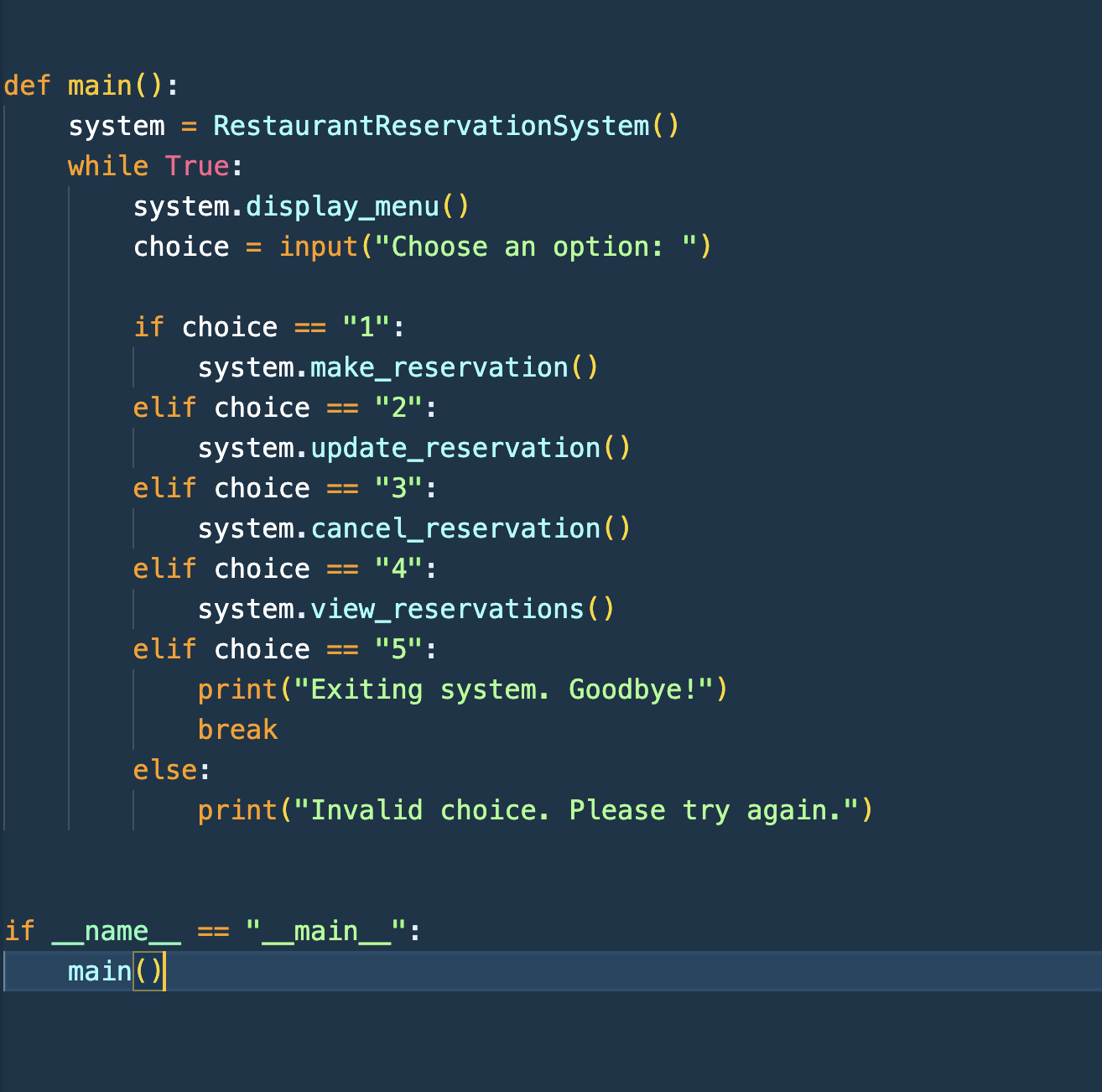
Yuqoridagi A.1-rasmda RestaurantRaservationSystem classi ichida 2 ta funksiye (def) display\_menu hamda make\_resertion funksiyalarini qamrab olgan. Birinchi funksiya foydalanuvchiga menuni ko’rsatadi, shu funksiya ichida oltita print ishlatmasdan bitta print va \n orqali yozib ketilsa bo’lardi, yani kod ixchamroq va kamroq ko’rinar edi. Ikkinchifunksiyada esa o’zgaruvchilar to’g’ri tanlanga va ular nimalarni saqlab turishini nomidan bilib olsangiz bo’ladi va key date va time ni o’ziga o’zlashtirvoladi qachonki foydalanuvchi yuqoridagi ma’lumotlarni kiritsa. Bu klassda faqat bitta metod bor: init. \_\_init\_\_ metodi Python dasturlash tilida maxsus ma’noga ega. U klassdan obyekt yaratilganda avtomatik ravishda chaqiriladi. Bunday metodlarga “constructor” deyiladi. \_\_init\_\_ metodi self.reservations = {} kod qatorini bajaradi. Bu kod qatori reservations nomli lug’atni yaratadi va uni klassning xususiyatiga aylantiradi. Lug’at bo’sh, ya’ni hech qanday bron qilish ma’lumotlari yo’q. Keyinchalik, bronlarni saqlash uchun bu lug’atdan foydalanish mumkin.

Har bir bron, lug’atga kalit (misol uchun, bron qiluvchi ismi yoki telefon raqami) va qiymat (masalan, bron qilingan vaqt yoki stol raqami) ko’rinishida qo’shilishi mumkin. Bu tizim orqali restoran bronlarini samarali ravishda boshqarish mumkin. Bu, restoranning bronlarni to’g’ri boshqarishini ta’minlaydi va mijozlarga qulaylik yaratadi.



A.2-rasm.

A.2-rasmda ham bir nechta metodlarni ko’rishingiz mumkin. update\_reservation metodi bronni yangilash uchun ishlatiladi. Avvaliga, foydalanuvchi bronni yangilash uchun sana va vaqtni kiritadi. Agar bu bron mavjud bo’lsa, foydalanuvchi yangi sana va vaqtni kiritadi. Agar yangi sana va vaqt bo’yicha bron mavjud bo’lsa, foydalanuvchiga xabar beriladi. Aks holda, eski bron o’chiriladi va yangi bron qo’shiladi. calcel\_reservation, bu metod bronni bekor qilish uchun ishlatiladi. Foydalanuvchi bekor qilish uchun sana va vaqtni kiritadi. Agar bu bron mavjud bo’lsa, u o’chiriladi va foydalanuvchiga xabar beriladi. view\_reservations bu metod hozirgi bronlarni ko’rish uchun ishlatiladi. Agar bronlar mavjud bo’lsa, ular ekranga chiqariladi. Aks holda, foydalanuvchiga xabar beriladi. Bu metod restoranning bronlarini tez va samarali ravishda ko’rishga yordam beradi. Bu, restoranning bronlarni to’g’ri boshqarishini ta’minlaydi va mijozlarga qulaylik yaratadi. for sikli Python dasturlash tilida biror amalni bir nechta marta takrorlash uchun ishlatiladi. Kiritilgan kodda for sikli self.reservations.items() metodi yordamida bronlarni chiqarish uchun ishlatilgan. Bu kod qismi har bir bronni ekranga chiqaradi. self.reservations.items() metodi lug’atning har bir elementini ikki qismga ajratadi: kalit va qiymat. Bu yerda kalit bu (date, time) va qiymat bu (name, people). For sikli har bir bron uchun takrorlanadi va har bir takrorlashda, bronning sanasi, vaqti, ismi va odamlar soni ekranga chiqariladi. Bu kod qismi restoranning bronlarini tez va samarali ravishda ko’rishga yordam beradi.



A.3-rasm.

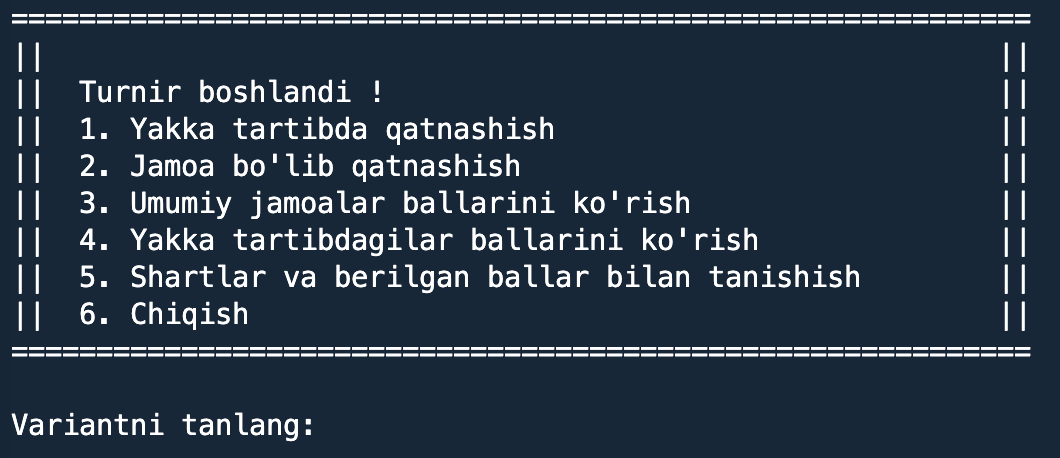
A.3-rasmdagi while True: - bu tsikl dastur tugaguncha takrorlanadi. Bu, foydalanuvchiga bir nechta amallarni bajarish imkoniyatini beradi, masalan, bir nechta bronirovaniye qilish yoki bronirovaniyelarni ko’rish. Foydalanuvchi 5-sonli tanlovni kiritganda, break operatori ishga tushadi va tsikl to’xtaydi, bu esa dasturni tugatadi. Bu usul, foydalanuvchiga nafaqat bir marta, balki bir nechta marta amal bajarish imkoniyatini beradi. Bu, restoran bronirovaniye tizimida juda foydali bo’lishi mumkin, chunki foydalanuvchilar odatda bir nechta buyurntma qilishni xohlaydilar.

3. TOPSHIRIQ 2

**Topshiriq 2 **Learning Aim B va C** ni qamrab oladi.**

**B.P4** Mijoz talablariga mos keladigan kompyuter dasturi uchun dizayn yaratish.

Mijoz talablariga mos keladiga dizanni yaratish uchun foydalanuchiga ko’rinadiga qismi tartibli va ma’lumot kiritish qulay bo’lishi kerak. Misol uchun menu qismida tartiblangan, yani alohida ajrarib turishi uchun (K.1-rasm).

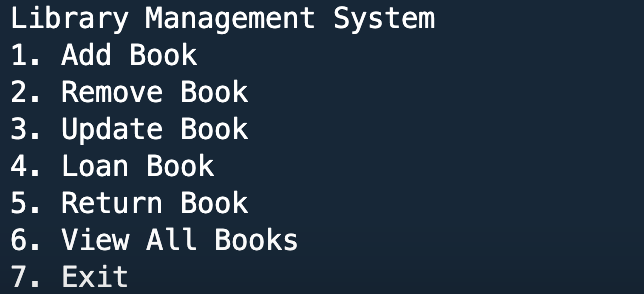


K.1-rasm.

Keyingi jarayonlarda har bitta so’rov ajralib turadi va ushbu dizayn foydalanuchi nima borligini va nima jarayon sodir bo’layotganini yaqqol ko’rishi/tushunishi mumkin.

****B.P5**** Taklif qilingan yechimlarga o'zgartirishlar kiritish uchun boshqalar bilan dizaynni ko'rib chiqish.

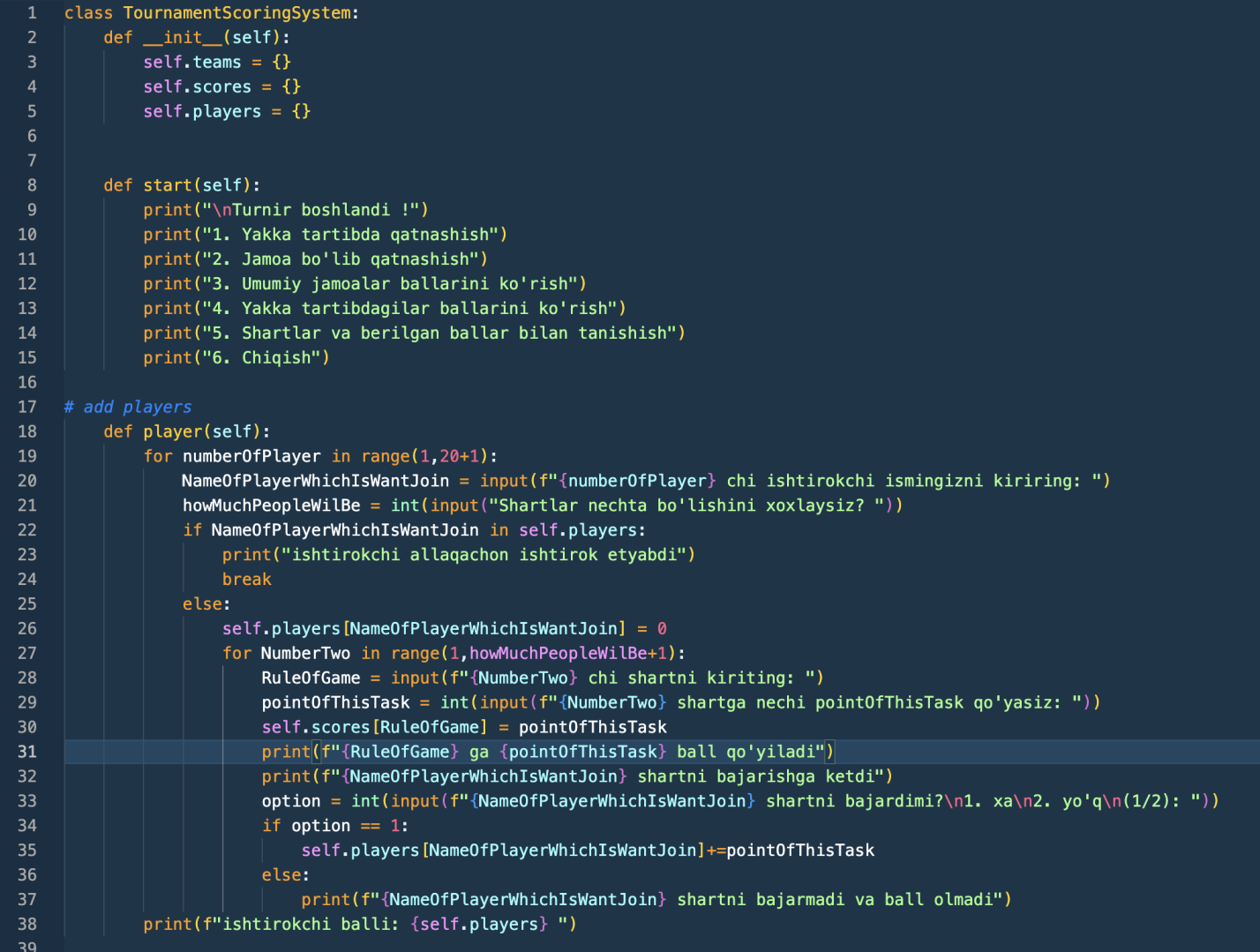
Boshqalar ha xuddi shunga o’xshash dizayndan foydalangan va shu dizan foydalanuvchi interfeysiga qulay tarazda qo’llanganligini ko’rishingiz mumkin.



K.2-rasm.

K.2-rasmda ko’rib turganigizdek, albatta bu sodda qilib dizayn berilgan va bu dizayn ham foydalanuvchi tushunishiga oson va qulay. Bundan tashqari kamchiliklari bor, yani buyruq dasturga berayotganda yangi qatorga tushmayapti, bunda foydalanuvhi adashib qolishi mumkin.

**C.P6** Mijoz talablariga javob beradigan kompyuter dasturini yaratish.

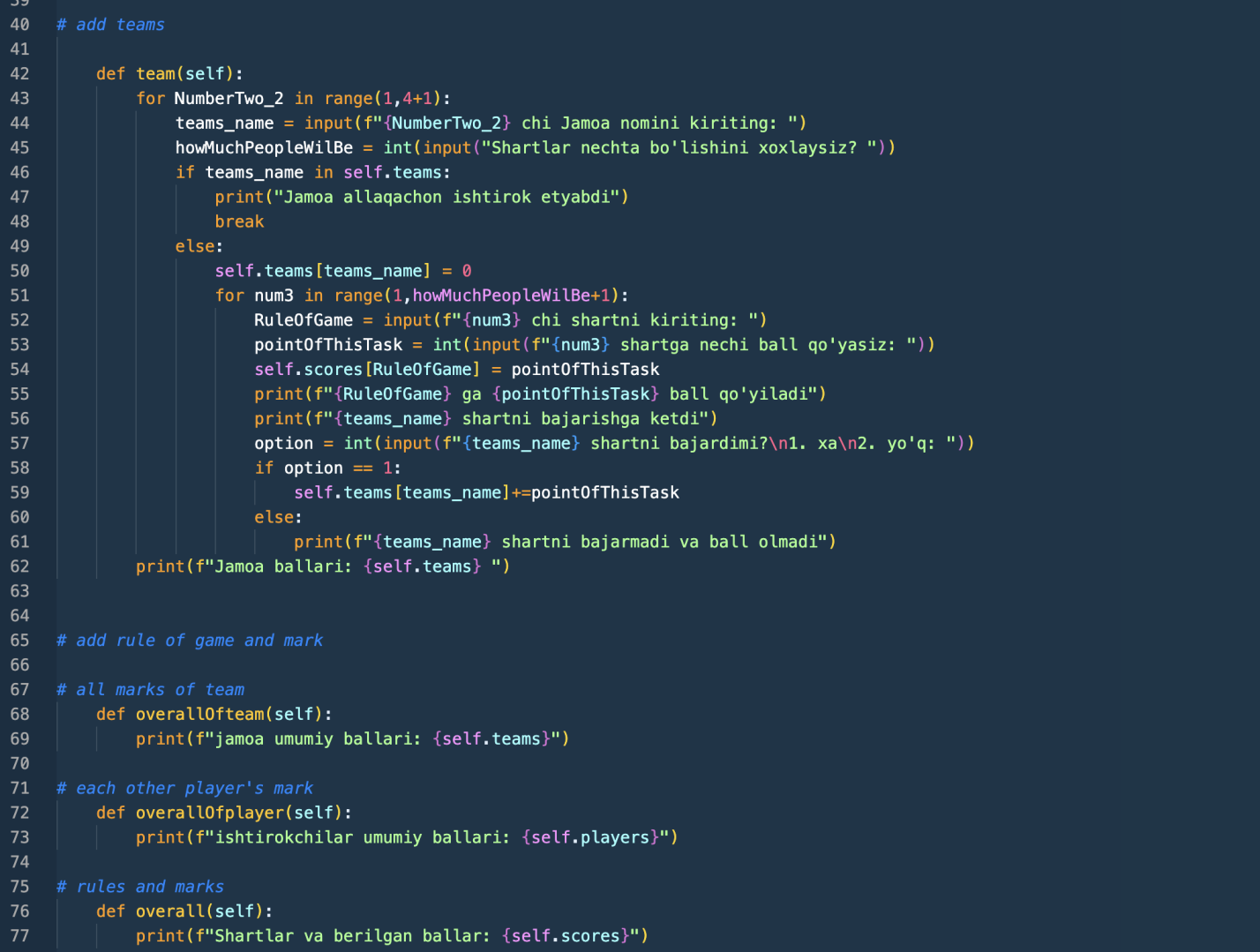


C.1-rasm.

\_\_init\_\_ metodi: Bu klassning konstruktori. U klassning obyektlarini yaratishda ishlatiladi. Bu yerda, teams, scores va players nomli 3 ta lug’at yaratiladi.

start metodi: Bu metod turnirni boshlash uchun ishlatiladi. Foydalanuvchiga turli xil tanlovlar beriladi, masalan, yakka tartibda qatnashish, jamoa bo’lib qatnashish, jamoalar yoki yakka tartibdagilar ballarini ko’rish, shartlar va berilgan ballar bilan tanishish yoki dasturdan chiqish.

player metodi: Bu metod ishtirokchilarni qo’shish uchun ishlatiladi. Har bir ishtirokchi uchun, ishtirokchi ismi va shartlar soni so’raladi. Agar ishtirokchi allaqachon ro’yxatdan o’tgan bo’lsa, xabar chiqariladi va qo’shish jarayoni to’xtatiladi. Aks holda, ishtirokchi ro’yxatga olinadi va unga 0 ball beriladi. Keyin, har bir shart uchun, shart matni va unga beriladigan ballar soni so’raladi. Shart bajarilgandan so’ng, ishtirokchiga ballar qo’shiladi. Agar ishtirokchi shartni bajarmagan bo’lsa, u ball olmaydi. Oxirgi qatorda, ishtirokchilar va ularning jami ballari chiqariladiC1-rasm).



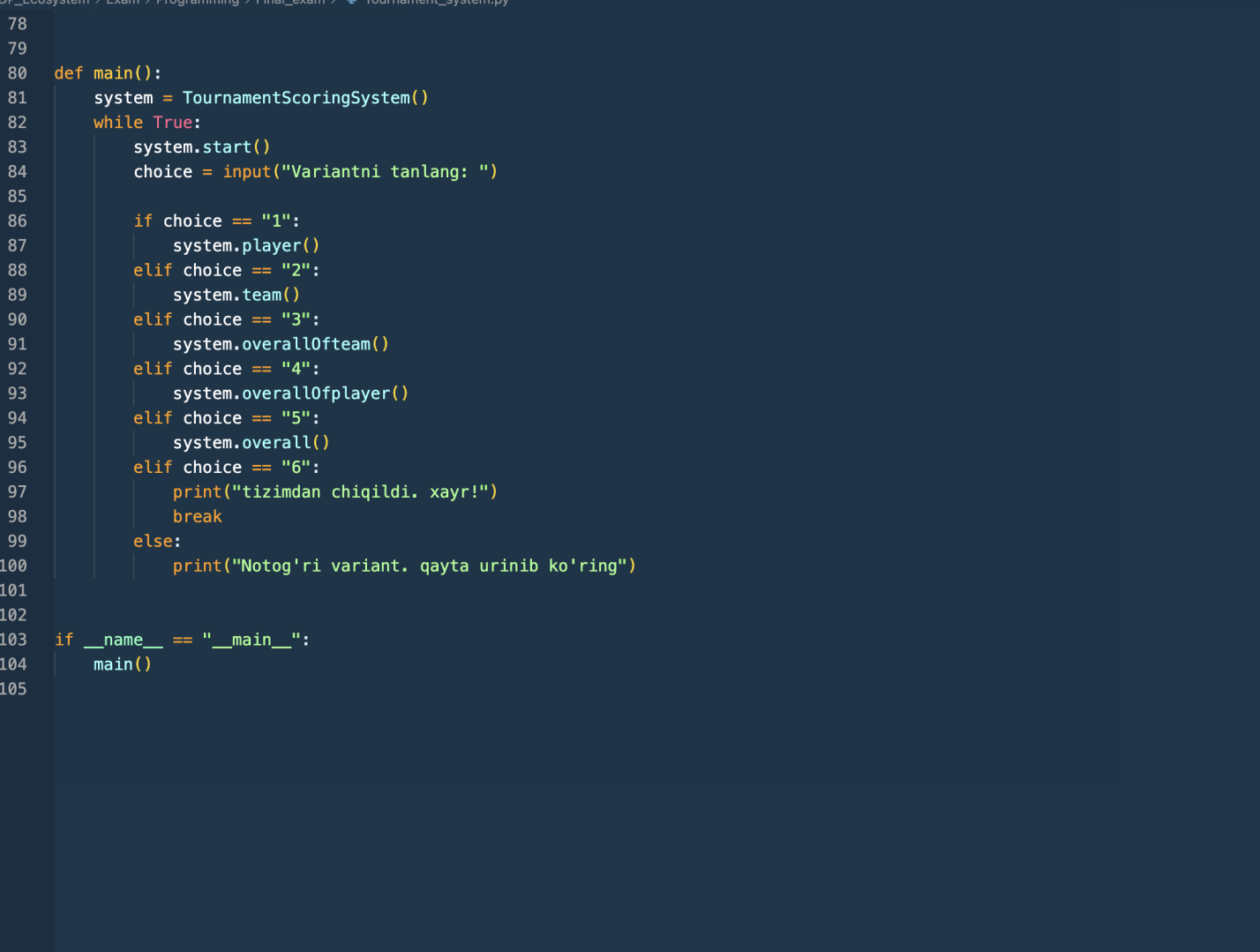
C.2-rasm.

team metodi: Bu metod jamoalarni qo’shish uchun ishlatiladi. Har bir jamoa uchun, jamoa nomi va shartlar soni so’raladi. Agar jamoa allaqachon ro’yxatdan o’tgan bo’lsa, xabar chiqariladi va qo’shish jarayoni to’xtatiladi. Aks holda, jamoa ro’yxatga olinadi va unga 0 ball beriladi. Keyin, har bir shart uchun, shart matni va unga beriladigan ballar soni so’raladi. Shart bajarilgandan so’ng, jamoaga ballar qo’shiladi. Agar jamoa shartni bajarmagan bo’lsa, u ball olmaydi. Oxirgi qatorda, jamoalar va ularning jami ballari chiqariladi.

overallOfteam metodi: Bu metod jamoalar umumiy ballarini chiqarish uchun ishlatiladi.

overallOfplayer metodi: Bu metod ishtirokchilar umumiy ballarini chiqarish uchun ishlatiladi.

overall metodi: Bu metod shartlar va berilgan ballarni chiqarish uchun ishlatiladi.



C.3-rasm.

system.player(): O’yinchilar haqida ma’lumotlarni ko’rsatadi.

system.team(): Jamoalar haqida ma’lumotlarni ko’rsatadi.

system.overallOfteam(): Jamoalar umumiy natijalarini ko’rsatadi.

system.overallOfplayer(): O’yinchilar umumiy natijalarini ko’rsatadi.

system.overall(): Umumiy natijalarni ko’rsatadi.

Foydalanuvchi konsolda raqamlar orqali kerakli variantni tanlaydi. Agar foydalanuvchi “6” raqamini tanlasa, dastur tugaydi va “tizimdan chiqildi. xayr!” xabari chiqadi. Agar foydalanuvchi noto’g’ri raqamni kiritsa, dastur “Noto’g’ri variant. qayta urinib ko’ring” xabarini chiqaradi va foydalanuvchidan qayta variantni tanlashni so’raydi.

**C.P7** Yakuniy kompyuter dasturi mijoz talablariga qanchalik mos kelishini ko'rib chiqish.

Albatta buni mijozga topshirishdan oldin test qilish kerak. Buni test.py orqali ko’rishingiz mumkin. Bu kod Python tilida yozilgan va unittest kutubxonasini ishlatib TournamentScoringSystem klassini sinovdan o’tkazish uchun ishlatiladi. Import unittest: Bu satr Pythonning unittest kutubxonasini import qiladi. Bu kutubxona sinovlarni yozish va bajarganish uchun ishlatiladi.

from Tournament\_system import TournamentScoringSystem: Bu satr Tournament\_system modulidan TournamentScoringSystem klassini import qiladi.

class TestTournamentScoringSystem(unittest.TestCase): Bu satr yangi sinov klassini e’lon qiladi. Bu klass unittest.TestCase klassidan vorislik oladi, shuning uchun uni sinovlar uchun ishlatish mumkin.

def setUp(self): Bu metod har bir sinovdan oldin chaqiriladi. Buyerda, TournamentScoringSystem klassining yangi nusxasi yaratiladi va self.system ga saqlanadi.

def test\_initialization(self): Bu metod TournamentScoringSystem klassining boshlang’ich holatini tekshiradi. U self.system.teams, self.system.scores, va self.system.players ning boshlang’ich holatini tekshiradi.

if \_\_name\_\_ == '\_\_main\_\_': unittest.main(): Bu satr faqatgina fayl to’g’ridan-to’g’ri ishga tushirilganda sinovlarni ishga tushiradi.

Bu tast hamda yozib ko’rigan natijalar mijozga mos kelishini ko’rsatadi.

**B.M2** Dizayn qarorlarini asoslab, dizaynning samarali yechim olib kelishini ko'rsatish.

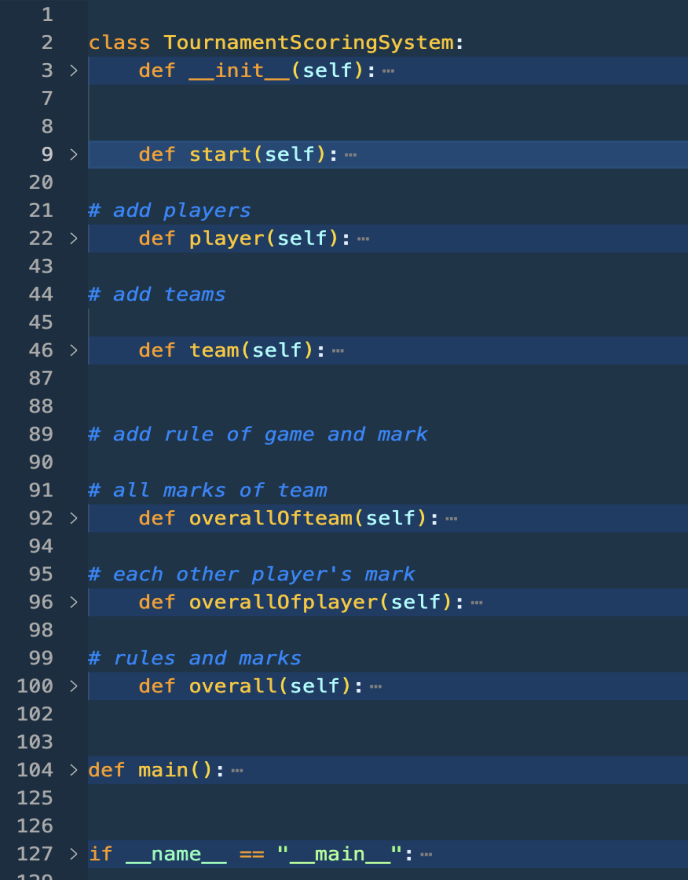
Bu dizayn qarorlari samarali hisoblanadi chunki:

Kodni tashkil etish: Har bir klassning vazifasi aniq, shuning uchun kodni o’qish va tushunish oson.

Kengaytirishga qulaylik: Agar siz turnir tizimiga yangi xususiyatlar qo’shishni xohlasangiz, masalan, jamoalar sonini hisoblash yoki o’yin natijalarini saqlash, siz shunchaki yangi metodlarni kerakli klassga qo’shishingiz mumkin.

Qayta ishlash: Bu dizayn qarorlari sizga kodni qayta ishlatish imkonini beradi. Masalan, siz Match klassini turli xil turnirlar uchun qayta ishlata olasiz.

Bu dizayn qarorlari Python dasturlash tilida turnir tizimini yaratish uchun samarali yechim bo’lib xizmat qiladi. Bu yechimlar sizning loyiha talablaringizga, ma’lumotlar tuzilishingizga va kodning oson o’qilishiga qarab turli bo’lishi mumkin. Bu kodda, Team, Match va Tournament klasslari yaratilgan. Bu dizayn qarorlari Pythonning obyektga yo’naltirilgan dasturlash (OOP) imkoniyatlarini ishlatadi. Har bir Match obyekti ikkita Team obyektini o’z ichiga oladi va g’alaba qozonuvchi jamoani saqlaydi. Tournament obyekti esa barcha Team va Match obyektlarini saqlaydi (K.3-rasm).



K.3-rasm.

**B.M3** Dasturni mijoz talablariga moslashtirish.

Mijoz ishtirokchilarni jamoa yoki yakka tartibda qatnashishini hohlagan va bunga system.player() va system.choose() javob beradi, yani foydalanuvchi uchun bu ikkita qism kerak bo’lagi uchun dasturga qo’shilgan va ularni self.players={} hamda self.teams={} lug’tiga qo’shib ketadi.

Bu dastur asosan sportga moslashgani uchun har bir ishtirokchini balini ko’rsatiladi yoki guruh bo’lib qatnashsa har bir guruh ballari ko’rsatiladi. Ballarni o’zlari bergani uchun har bir jamoa yoki guruhga qo’lda ballar kiritiladi va ular saqlanib boriladi.

Dasturni mijoz talablariga moslashtirish uchun quyidagi qadamlarni amalga oshirishingiz mumkin:

1.Talablarni tushunish: Mijozning nima xohlaganini aniq tushunish uchun, mijoz bilan muloqot qilish va uning talablari va maqsadlarini aniq tushunish juda muhim. Bu, sizning dasturingizning mijozning muammolarini hal qilishga qanday yordam berishi kerakligini tushunishingizga yordam beradi.

2.Dizayn qarorlarini qabul qilish: Mijozning talablari asosida, siz dastur dizaynini belgilash uchun qanday dizayn qarorlarini qabul qilishingiz kerakligini aniqlashingiz kerak. Bu, dastur arxitekturasi, ma'lumotlar tuzilmasi, interfeys va boshqa muhim elementlarni o'z ichiga oladi.

3.Prototip yaratish: Mijoz talablari asosida prototip yaratish va uni mijozga ko'rsatish. Bu, sizning dizayn qarorlaringizning mijozning talablari bilan mos kelishini tekshirishga yordam beradi.

4.Iteratsiya: Mijozdan qaytarishni olganingizdan so'ng, siz dasturni takomillashtirishingiz va mijoz talablari bilan yanada moslashtirishingiz kerak. Bu jarayon bir necha marta takrorlanishi mumkin.

5.Testlash va tekshirish: Dasturni mijoz talablari bilan mos kelishini ta'minlash uchun, siz uni to'liq testlash va tekshirishingiz kerak. Bu, dasturda xatoliklarni aniqlash va ularni tuzatishga yordam beradi.

Bu qadamlar menga dasturni mijoz talablari bilan moslashtirishga yordam bergan va yuqorida ko’rishingzi mumkin.

**BC.D2** Yakuniy dizayn va optimallashtirilgan dasturiy ilovani mijoz talablariga qarshi baholash.

 Yakuniy dizayn va optimallashtirilgan dasturiy ilovani mijoz talablariga qarshi baholash uchun quyidagi qadamlarni amalga oshirdim:

1. Funksional talablarni tekshirish: Dastur mijozning funksional talablari bilan mos kelishini tekshirish uchun, men dasturni barcha funksiyalarini tekshirdim. Bu dastur sizning loyiha talablaringizga qarab turli bo'lishi mumkin.

2. Istifodachi interfeysi: Istifodachi interfeysi mijozning talablari bilan mos kelishini tekshirish uchun, men istifodachi interfeysini tekshirishim kerak. Bu interfeysning foydalanuvchiga qulay, tushunarli va samarali bo'lishi kerak.

3. Ishonchli va xavfsizlik: Dastur ishonchli va xavfsiz bo'lishi kerak. Bu, dasturda xatoliklar yoki xavfsizlik nuqsonlari bo'lmaganligini ta'minlash uchun men dasturni to'liq testladim va tekshirdim.

4. Optimallashtirish: Dastur samarali ishlash uchun, siz dasturni optimallashtirishingiz kerak.

5.Mijozdan feedback olish: Mijozdan feedbackni olganingizdan so'ng, siz dasturni takomillashtirishingiz va mijoz talablari bilan yanada moslashtirishingiz kerak. Bu jarayon bir necha marta takrorlanishi mumkin.

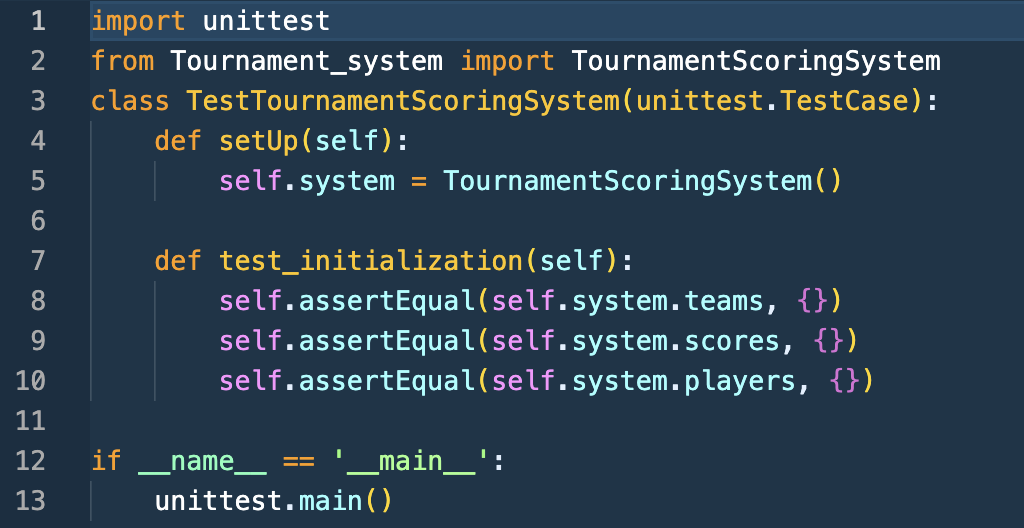
**BC.D3** Shaxsiy mas'uliyat, ijodkorlik va samarali o'z-o'zini boshqarishni dastur dizayni, ishlab chiqilishi va ko'rib chiqilishida namoyish etish.  
Shaxsiy mas'uliyat, ijodkorlik va samarali o'z-o'zini boshqarish dastur dizayni, ishlab chiqilishi va ko'rib chiqilishida juda muhimdir. Bu prinsiplar quyidagi usullar orqali namoyon bo'ladi:

1.Tahlil va dizayn: Dasturlash jarayonining boshlanishida, muammoni tushunish va uni hal qilish uchun kerak bo'lgan funksiyalarni aniqlash uchun shaxsiy mas'uliyat talab qilinadi. Bu jarayon ijodkorlikni talab qiladi, chunki siz muammoni yechimlari uchun yangi va samarali yondashuvlarni topishingiz kerak.

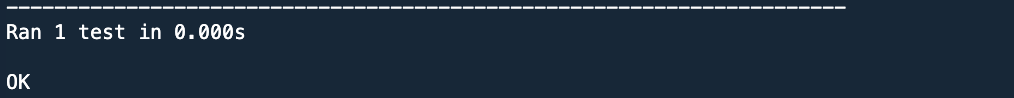
2.Kod yozish: Dastur yaratish jarayonida, dasturchi o'z-o'zini boshqarishni ko'rsatishi kerak. Bu, vazifalarni bajarish, kodni toza va tushunarli yozish, va dasturning samaradorligini ta'minlash uchun kerak bo'lgan optimizatsiyalarni amalga oshirishni o'z ichiga oladi.

3.Testlash va tekshirish: Dastur ko'rib chiqilishi jarayonida, dasturchi o'z-o'zini boshqarishni namoyon qilishi kerak. Bu, kodni tekshirish, xatoliklarni topish va ularni tuzatishni o'z ichiga oladi. Bu jarayon shuningdek, dastur samarali ishlashini ta'minlash uchun kerak bo'lgan optimizatsiyalarni amalga oshirishni talab qiladi.Albatta test natijasi ham OK bo’ldi va buni foydalanuvchiga bemalol topshirsa bo’ladi.

Bu prinsiplar dasturchining muvaffaqiyatli bo'lishi uchun muhimdir va ularning ishlarining sifatini ta'minlashga yordam beradi. Bu prinsiplar har bir dasturchi uchun muhimdir, chunki ular dastur yaratish jarayonining har qanday bosqichida ishlatilishi mumkin (T.1-rasm va T.2-rasm).



T.1-rasm.



T.2-rasm.

4. XULOSA

Har bir dastur tuzishda hisoblash fikrlash va logika muhim ahamiyatga ega va siz uni qancha barqaror va sifatli ishlashi uchun kerakli dasturlash tilini tanlay olishingiz va test qilishingiz muhimdir. Toki foydalanuvchiga topshirgunga qadar siz u dasturning qanday ishlashi uchun va har qanday logik xatolarni bartaraf etishingiz kerak. Bundan tashqari dizaynnga ham e’tibor berishingiz kerak va foydalanuvchi interfeysi uchun ko’proq ishashingiz kerak.

**So’zlar soni: 3590**

Manbalar ro’yXati

1. Anon, (2023). Ijodiy fikrlash ko’nikmalarini egallash: 2024 yilda keng qamrovli qo’llanma - AhaSlides. [online] Available at: <https://ahaslides.com/uz/blog/creative-thinking-skills/.>
2. Boom, K.-D., Bower, M., Siemon, J. and Arguel, A. (2022). Relationships between computational thinking and the quality of computer programs. Education and Information Technologies. doi:https://doi.org/10.1007/s10639-022-10921-z.
3. www.w3schools.com. (n.d.). Python User Input. [online] Available at: https://www.w3schools.com/python/python\_user\_input.asp.‌
4. PythonForBeginners.com. (2021). How to Take User Input in Python. [online] Available at: https://www.pythonforbeginners.com/basics/how-to-take-user-input-in-python.‌
5. Stack Abuse. (n.d.). Using For and While Loops for User Input in Python. [online] Available at: https://stackabuse.com/bytes/using-for-and-while-loops-for-user-input-in-python/.
6. Bing. (n.d.). matematik va taklif mantiqining tamoyillari. [online] Available at: https://www.bing.com/search?q=matematik+va+taklif+mantiqining+tamoyillari&toWww=1&redig=C17303DAA67446ABA49B61D839CDF590 [Accessed 9 Jan. 2024].
7. www.learning-mind.com. (2020). How to Use Computational Thinking to Solve Problems Like a Pro - Learning Mind. [online] Available at: https://www.learning-mind.com/computational-thinking-problem-solving/#google\_vignette [Accessed 18 Jan. 2024].‌
8. brainly.in. (n.d.). 2. How are Computational Thinking skills applied in finding solutions that can be interpreted into - Brainly.in. [online] Available at: https://brainly.in/question/28381946.
9. Google.com. (2019). Google for Education: Computational Thinking. [online] Available at: https://edu.google.com/resources/programs/exploring-computational-thinking/.‌
10. PythonForBeginners.com. (2021). How to Take User Input in Python. [online] Available at: https://www.pythonforbeginners.com/basics/how-to-take-user-input-in-python.‌
11. Stack Overflow. (n.d.). Tournament Program. [online] Available at: https://stackoverflow.com/questions/19611749/tournament-program [Accessed 18 Jan. 2024].

‌