

Dasturlash uslublari va paradigmalari fanidan yakuniy nazorat savollari.

№	Nazorat savollari
1.	Semantika dasturlash paradigmasini aniqlashga qanday ta'sir qiladi? Dasturlash tillarini ishlab chiqish kontekstida semantika va pragmatika o'rtasidagi farq nimada?
2.	“Dasturlash uslublari va paradigmalar” fanining asosiy vazifalari qaysilar? Batafsil yozing.
3.	Deklarativ dasturlashning asosiy prinsiplari qanday? Bu yondashuvning imperativ dasturlashga nisbatan afzalligi nimada?
4.	Sinflar tizimini moslashuvchan qilish uchun polimorfizm mexanizmidan foydalanadigan dastur arxitekturasini ishlab chiqing. Bu kodni qo'llab-quvvatlash va kengaytirishni qanday soddalashtiradi?
5.	Soft funksiyalar va rekursiyadan foydalanadigan dastur modelini loyihalang. Ushbu konsepsiyalardan foydalanishning afzalliklari nimada?
6.	Imperativ dasturlashning asosiy tamoyillari qaysilar? Batafsil yozing.
7.	Imperativ va deklarativ dasturlashni solishtiring.
8.	Protседurali dasturlashning o'ziga xos xususiyatlari nimada?
9.	Dasturlash kontekstida semantika va sintaksis o'rtasidagi farqni tushuntiring. Kod yozishda ular qanday o'zaro ta'sir qiladi?
10.	Strukturaviy dasturlash nima? Strukturaviy dasturlashni qo'llashda qanday tamoyillar muhim?
11.	Standart imperativ dasturlash tillariga misol keltiring. Imperativ dasturlash qanday afzalliklarga ega? Algoritmalar imperativ dasturlashda qanday rol o'ynaydi?
12.	Obyektga yo'naltirilgan dasturlashda ko'p merosdan qanday foydalaniladi? Bu qanday afzalliklar va murakkabliklarni keltirib chiqaradi?
13.	Moslashuvchan arxitektura yaratish uchun OOP funksional modellardan foydalanadigan dastur modelini loyihalang. Ushbu model qanday afzalliklar beradi?
14.	Funksional dasturlash paradigmasining asosiy xususiyatlari qaysilar? Yuqori tartibli funksiyalar nima? Misol keltiring.
15.	Hisoblash vazifalarini bajarish uchun parallel algoritmalaridan foydalanadigan dastur konsepsiyasini yarating. Oqimlar o'rtasidagi raqobatni qanday boshqarish mumkin?
16.	Rekursiyaning funksional dasturlashdagi ahamiyati nimada? Soft funksiyalar deganda nimani tushunasiz?
17.	Operatsion semantika dasturlash tillarini formal tavsiflashda qanday yordam beradi? Uning qo'llanilishiga misollar keltiring.
18.	Funksional dasturlashning amaliy qo'llanilish sohalari qanday? Funksional va imperativ dasturlashni solishtiring. Lambda funksiyalar funksional dasturlashda qanday ishlatiladi?

19.	azifalarni bajarishni boshqarish uchun operatsion semantikadan foydalanadigan dastur konsepsiyasini yarating. Ushbu yondashuv qanday afzalliklar beradi?
20.	Obyektlarni boshqarish vazifalarini yechish uchun meros va polimorfizmdan foydalanadigan dastur arxitekturasini loyihalang. Katta tizimlar kontekstida bu qanday afzalliklar beradi?
21.	Mantiqiy dasturlash paradigmasi qanday ishlaydi? Prolog dasturlash tili qaysi paradigмага kiradi?
22.	Obyektga yo'naltirilgan dasturlashning funksional modeli nima? U an'anaviy OOP modeli bilan qanday uyg'unlashadi?
23.	Mantiqiy dasturlashning amaliy qo'llanilish sohalarini ayting. Predikatlar nima va ular mantiqiy dasturlashda qanday ishlatiladi?
24.	Skript dasturlash paradigmasi tez prototip yaratishda qanday qo'llaniladi? Dasturlash tillari va qo'llash stsenariylariga misollar keltiring.
25.	Deklarativ dasturlashning afzalliklari qanday? Deklarativ va mantiqiy dasturlashni solishtiring. Deklarativ dasturlashning asosiy tamoyillari qaysilar?
26.	Katta hajmdagi ma'lumotlarni qayta ishlash uchun iteratorlardan foydalanadigan dastur konsepsiyasini yarating. Iteratorlardan foydalanish an'anaviy sikllarga nisbatan qanday afzalliklar beradi?
27.	Deklarativ dasturlashning qanday turlari mavjud? Deklarativ dasturlashni imperativ dasturlashdan qanday ajratasiz? SQL deklarativ dasturlash tili hisoblanadimi? Nega?
28.	Umumiy maqsadli dasturlash tillarida parallel dasturlash modellari maxsus parallel hisoblash tillaridagi modellardan qanday farq qiladi? Bunday tillarga misollar keltiring.
29.	Dasturlash paradigmalarining multiparadigmali tillarning rivojlanishidagi roli qanday? Multiparadigmali tillar qanday vazifalarni yechadi?
30.	Veb dasturlarni bajarish jarayonida virtual mashinalar va refleksiya qanday qo'llaniladi? Bu ishlab chiquvchilarga qanday imkoniyatlarni beradi?
31.	OOP dasturlashning asosiy tamoyillari qaysilar? Sinf va ob'yekt nima? Misol keltiring.
32.	Obyektga yo'naltirilgan dasturlash va struktural dasturlashning asosiy farqlari qanday? Paradigma tanlovi dastur loyihalashga qanday ta'sir qiladi?
33.	Merosxorlik deganda nimani tushunasiz. Inkapsulyatsiya nima va uning afzalliklari qanday?
34.	Multiparadigmali dasturlash tillarining rivojlanishi dastur loyihalashga qanday o'zgarishlar kiritdi? Zamonaviy multiparadigmali tillarga misollar keltiring.
35.	Polimorfizm tushunchasini tushuntiring. OOP ning funksional modeli qanday ishlaydi?

36.	Ma'lumotlar oqimlarini qayta ishlash uchun iteratorlar va konveyerlardan foydalanadigan dastur konsepsiyasini loyihalang. Bunday yondashuv qanday afzalliklar beradi?
37.	OOP dasturlash tillariga misollar keltiring. Mavhumlik (abstraction) OOPda qanday ishlatiladi? OOP va strukturaviy dasturlashni solishtiring.
38.	Murakkab hisoblash vazifalarini bajarish uchun rekursiv funksiyalardan foydalanadigan dastur konsepsiyasini yarating. Qaysi holatlarda rekursiya iteratsiyadan afzal?
39.	Obyektlar iyerarxiyasini yaratish uchun sinflar va merosdan foydalanadigan dastur konsepsiyasini loyihalang. Bunday holatda merosdan foydalanishning afzalligi nimada?
40.	Modul dasturlash konsepsiyasining mohiyati nimada? Bu murakkab dasturlarni ishlab chiqish jarayonini qanday soddalashtiradi?
41.	Parallel dasturlashning maqsadi nima. Parallel algoritmlar qanday ishlaydi. Parallel dasturlash tillariga misollar keltiring.
42.	Obyektga yo'naltirilgan dasturlashda ko'p meros nima? Bu qanday muammolarni keltirib chiqarishi mumkin va qanday hal qilinadi?
43.	Metadasturlash o'zini o'zgartiruvchi kod yaratish uchun qanday qo'llanilishi mumkin? Bunday yondashuv qaysi stsenariylarda samarali?
44.	Parallel dasturlashda xotira modeli qanday rol o'ynaydi. Parallel dasturlashda transformatsion semantika nima?
45.	Yuqori unumdor hisoblashlar uchun asosiy dasturlash paradigmalarini sanab o'ting. Ular dastur samaradorligini qanday oshiradi?
46.	Parallel dasturlarni yaratishning qaysi usullari mavjud. Amaliy dasturlash tizimlarida parallellik qanday qo'llaniladi. Parallel dasturlashning afzalliklari va kamchiliklari nimada?
47.	Mantiqiy dasturlashda ma'lumotlarni qayta ishlash uchun predikatlardan foydalanadigan dastur konseptual modelini yarating. Predikatlar bilan ishlashning asosiy xususiyatlari qanday?
48.	Dasturlash paradigmalarining mohiyati nima va ularning asosiy turlari qanday? Ular kod yozish uslubi va yondashuviga qanday ta'sir qiladi?
49.	Skriptli dasturlash paradigmasining asosiy xususiyatlari qaysilar. World Wide Web skriptlariga misol keltiring. Domen muammolari va ular uchun skript tillarining ahamiyati qanday?
50.	Parallel hisob-kitoblarni qayta ishlash uchun ko'p oqimli dastur yarating. Musobaqa holatidan qochish uchun oqimlarni sinxronlashni qanday amalga oshirish mumkin?
51.	Funksional dasturlashda qanday xotira boshqaruv modellari qo'llaniladi? Ularning ishlatilishi dastur unumdorligiga qanday ta'sir qiladi?
52.	Skript dasturlash qachon afzaldir. Skript tillari va an'anaviy dasturlash tillarini solishtiring. Skriptli dasturlash tillariga misollar keltiring?
53.	Ishlash jarayonida jarayonlarni boshqarish uchun refleksiya dan foydalanadigan dastur konsepsiyasini loyihalang. Ushbu texnika yordamida qanday vazifalar soddalashtiriladi?

54.	Deklarativ dasturlashning asosiy prinsiplari qanday? U imperativ dasturlashdan nimasi bilan farq qiladi?
55.	Metadasturlash nima. Metadasturlashning qanday turlari mavjud. Meta funksiyalar deganda nimani tushunasiz?
56.	Obyektga yo'naltirilgan dasturlashning asosiy prinsiplari qanday va ular turli dasturlash tillarida qanday amalga oshiriladi?
57.	Mantiqiy dasturlashning asosiy prinsiplari qanday? Qaysi vazifalar mantiqiy dasturlash yordamida osonroq hal qilinadi?
58.	Metadasturlashning afzalliklari nimada. O'z-o'zini o'zgartiruvchi kod qanday ishlaydi. Metadasturlashning xavfsizlikka qanday ta'siri bor?
59.	Abstrakt mashinaning funksiyasi dasturlash tillarida nimadan iborat? U tillar va kompilyatorlarni ishlab chiqishda qanday yordam beradi?
60.	Ma'lumotlar abstraksiyasi uchun subdasturlardan foydalanadigan dastur tuzilmasini loyihalang. Bu murakkab hisoblash jarayonlarini qanday soddalashtiradi?
61.	Protsedurali dasturlashda funksiya va protsedura o'rtasidagi farq nimada?
62.	Hisoblash unumdorligini oshirish uchun parallel algoritmlar qo'llanadigan dastur konsepsiyasini yarating. Oqimlarni sinxronlash muammosini qanday hal qilish mumkin?
63.	Dasturlash paradigmalari kontekstida ma'lumotlar tuzilmalari bilan ishlash tamoyillari qanday? Turli paradigmalarda doirasida qaysi ma'lumotlar tuzilmalari qo'llaniladi?
64.	Dastur kodini yaxshilash paradigmasidan foydalanadigan dastur konsepsiyasini yarating. Yaxshilash samaradorlik va kod o'qilishiga qanday ta'sir qiladi?
65.	Protsessual va obyektga yo'naltirilgan dasturlash o'rtasidagi asosiy farqlar qanday? Ushbu yondashuvlar dasturiy ta'minot arxitekturasiga qanday ta'sir qiladi?
66.	Metadasturlashdan foydalanib, ish vaqtida kod generatsiya qiladigan dastur konsepsiyasini yarating. Metadasturlash yordamida qanday vazifalar hal qilinadi?
67.	Dasturlashning quyi darajali tillari nima? Misol keltiring. Yuqori darajali tillar qanday afzalliklarga ega?
68.	Ma'lumotlarni qayta ishlash uchun yuqori darajali funksiyalardan foydalanadigan dastur konsepsiyasini loyihalang. Ushbu yondashuv murakkab algoritmlarni boshqarishda qanday afzalliklar beradi?
69.	Skript dasturlash paradigmasining mohiyati nimada va u dinamik veb-ilovalar yaratishda qanday qo'llaniladi?
70.	Imperativ va deklarativ dasturlash o'rtasidagi farq nimada? Deklarativ yondashuv yordamida qaysi vazifalarni yaxshiroq hal qilish mumkin?
71.	Parallel algoritmlarni amalga oshirishda paydo bo'ladigan asosiy muammolar nimada? Sinxronizatsiya ularni hal qilishda qanday yordam beradi?
72.	Mantiqiy dasturlashning asosiy tamoyillari va vazifalari qanday? Mantiqiy dasturlash amaliyotda qanday qo'llaniladi?

73.	Yuqori darajali funksiyalardan foydalanish funksional dasturlarni qanday yaxshilaydi? Bu "funksiyalar tozaligi" tushunchasi bilan qanday bog'liq?
74.	Struktural dasturlashning asosiy prinsiplari dastur kodining sifatini oshirishga qanday yordam beradi? Ushbu yondashuvning qanday afzalliklari bor?
75.	Multiparadigmali dasturlash tillari turli yondashuvlarni bitta dasturda qanday birlashtirishga imkon beradi? Bunday tillar va ularning imkoniyatlariga misollar keltiring.
76.	Amalga oshirish pragmatikasi aniq vazifalar uchun dasturlash paradigmasini tanlashga qanday ta'sir qiladi? Paradigma tanlashda qaysi omillarni hisobga olish kerak?
77.	Yuqori samarador hisoblashlarda parallel algoritmlar qanday yordam beradi? Ularning qo'llanilishiga misollar keltiring.
78.	Dasturlashda xavfsizlikni boshqarishning asosiy tamoyillari nimada? Zaifliklarning oldini olish uchun qanday usullar qo'llaniladi?
79.	Deklarativ dasturlash tillarining asosiy xususiyatlari qanday? Ular imperativ tillardan qanday farq qiladi?
80.	Veb vazifalarni avtomatlashtirish uchun skript paradigmasidan foydalanadigan dastur konseptual sxemasini yarating. Skript tillari bu vazifani qanday soddalashtiradi?
81.	Dasturlashda boshqaruv abstraksiyasining roli nimada? Subdasturlar kodni strukturaviy qilish va soddalashtirishga qanday yordam beradi?
82.	Dasturlash paradigmatlari asosidagi asosiy g'oyalar evolyutsiyasi zamonaviy dasturlash tillarining rivojlanishiga qanday ta'sir qildi? Misollar keltiring.
83.	Semantika nima va u dasturlash tillarida dasturlar xatti-harakatini tavsiflashga qanday ta'sir qiladi?
84.	Dasturlash paradigmatlari rivojlanishining asosiy bosqichlari qanday? Ushbu bosqichlar kompyuter texnikasining evolyutsiyasi bilan qanday bog'liq?
85.	Kod generatsiyasi va optimallashtirishning farqi nimada?
86.	Parallel algoritmlarda "Deadlock" muammosi nima?
87.	Funksional dasturlashda iteratorlar va konveyerlarning ahamiyati nimada?
88.	Past darajali va yuqori darajali dasturlash tillarining asosiy farqlari nimada? Ularning har biri qaysi holatlarda qo'llaniladi?
89.	Funksional dasturlashda yuqori darajali funksiyalardan foydalanish kod yozishni qanday soddalashtiradi? Bunday funksiyalar qo'llanilishiga misollar keltiring.
90.	Yuqori darajali dasturlash tillari multiparadigmali yondashuvni qanday qo'llab-quvvatlaydi? Bunday qo'llab-quvvatlash ishlab chiquvchiga qanday afzalliklar beradi?
91.	Skript dasturlash paradigmasidan foydalanishning afzalliklari nimada? Skript tillari yordamida qaysi vazifalarni oson bajarish mumkin?

92.	Obyektga yo‘naltirilgan dasturlash inkapsulyatsiya va meros prinsiplarini qanday amalga oshiradi? Ularning dastur murakkabligini boshqarishdagi ahamiyati nimada?
93.	Dasturlarni bajarish jarayonida abstrakt mashinaning roli nimada? U dastur komponentlarining o‘zaro ta‘sirini qanday modellashtiradi?
94.	Amalga oshirish pragmatikasi dasturiy ta‘minotni ishlab chiqish va qo‘llab-quvvatlashga qanday ta‘sir qiladi? Dasturlash paradigmasini tanlashda uning roli nimada?
95.	Parallel vazifalarni bajarish uchun ko‘p oqimli dastur arxitekturasini loyihalang. Bunday dasturda musobaqa holatidan qanday qochish mumkin?
96.	Operatsion semantikaning asosiy xususiyatlari qanday? U dasturlarni xulq-atvorini formalizatsiya qilish uchun qanday qo‘llaniladi?
97.	Funksional dasturlashda iteratorlar va konveyerlarning ahamiyati nimada?
98.	Ob‘yektga yo‘naltirilgan dasturlashda polimorfizm qanday afzalliklar beradi?
99.	Dasturlash xavfsizligini ta‘minlash uchun qanday kodlash standartlari mavjud?
100.	Struktural dasturlash tamoyillari kodni o‘qilishini va qo‘llab-quvvatlashni qanday yaxshilaydi? Ushbu paradigmaga amal qiluvchi qoidalarni misol qilib keltiring.

ATDT kaferasi muiri

N.O.Raximov

Tuzuvchi:

B.I.Otaxonova

ATDT kafedrasining ____-sonli _____2024 dagi bayyonnomasi bilan tasdiqlangan.