

No	Savol va javoblar
1	<b>Repository Design Pattern-dan foydalanishning afzalliklaridan biri shundaki, u quyidagilarga yordam beradi?</b>
	<i>Tahlil masalalarini aniq ajratish</i>
	<i>Bir nechta dasturlash tillarini birlashtirish</i>
	<i>Murakkab algoritmlarni loyihalash</i>
	<i>Xotirani taqsimlashni boshqarish</i>

2	<b>Biznes sifatining atributlari qaysilar?</b>
	<i>Narx va jadval, bozor qobiliyati</i>
	<i>To'g'rilik</i>
	<i>Portativlik, integrallik, o'zgartirish imkoniyatlari</i>
	<i>Foydalanish imkoniyati</i>

3	<b>Repository Design Pattern dasturiy ta'minotni sinovdan o'tkazishga qanday ta'sir qiladi?</b>
	<i>Dasturiy ta'minotni sinab ko'rishni osonlashtiradi</i>
	<i>Dasturiy ta'minotni sinab ko'rishni qiyinlashtiradi</i>
	<i>Dasturiy ta'minotni sinab ko'rishga ta'sir qilmaydi</i>
	<i>Dasturiy ta'minotni sinovdan o'tkazishni oldini oladi</i>

4	<b>Repository Design Pattern asosan nimani boshqaradi</b>
	<i>Ma'lumotlar</i>
	<i>Tarmoq trafigi</i>
	<i>Foydalanuvchi seanslari</i>
	<i>Texnikaviy uzilishlar</i>

5	<b>Repository Design Pattern qanday turdagi muammolarni ajratishga yordam beradi?</b>
	<i>Tahlil masalalari</i>
	<i>Tarmoq muammolari</i>
	<i>Uskuna muammolari</i>
	<i>Foydalanuvchi interfeysi muammolari</i>

6	<b>3. Dizayn sifatlari atributlari qaysilar: 1 kontsepsual yaxlitlik; 2 saqlash qobiliyati; 3 to'g'rilik 4 qayta foydalanish qobiliyati; 5 masshtablilik; 6 xavfsizlik.</b>
	<i>1 2 4</i>
	<i>2 3 4</i>
	<i>1 4 6</i>
	<i>3 5 6</i>

7	<b>Dasturiy ta'minotni modellashtirish nima?</b>
	<i>kodlashdan oldin dasturiy ta'minotni loyihalash.</i>
	<i>Dasturiy ta'minot modellarini ishlab chiqish.</i>
	<i>dasturiy ta'minot diagrammalarini ishlab chiqish.</i>
	<i>dasturiy ta'minot prototiplarini ishlab chiqish.</i>

8	<b>Birlashtirilgan modellashtirish tili nima?</b>
	<i>ob'ektga yo'naltirilgan modellarni tavsiflash uchun standartlashtirilgan grafik til va yozuv.</i>
	<i>ob'ektga yo'naltirilgan modellarni tavsiflash uchun dasturlash tili.</i>
	<i>ob'ektga yo'naltirilgan modellarni chizish uchun diagramma vositasi.</i>
	<i>ob'ektga yo'naltirilgan modellarni tavsiflash uchun grafik til.</i>

<b>9</b>	<b>Dastur arxitekturasi nima?</b>
	<i>dasturiy ta'minot tizimining umumiy tuzilishi.</i>
	<i>bino ichidagi dasturiy ta'minot.</i>
	<i>mijoz / server tizimining tuzilishi.</i>
	<i>dasturiy ta'minot sinflari va ularning o'zaro aloqalari</i>

<b>10</b>	<b>Dasturiy ta'minot loyihasi kontseptsiyasi nima?</b>
	<i>tizimni loyihalashda qo'llanilishi mumkin bo'lgan asosiy g'oya.</i>
	<i>dasturiy ta'minotning grafik yoki matnli tavsifi.</i>
	<i>dasturiy ta'minotni hujjatlashtirish.</i>
	<i>dizayn ishlab chiqarish uchun tizimli yondashuv.</i>

<b>11</b>	<b>UML bilan ishlashning ikki xil diagrammasi qanday?</b>
	<i>ketma-ketlik diagrammasi va aloqa diagrammasi</i>
	<i>Sinf diagrammasi va ketma-ketlik diagrammasi</i>
	<i>sinf diagrammasi va aloqa diagrammasi</i>
	<i>Statechart va aloqa diagrammasi</i>

<b>12</b>	<b>O'zaro ta'sir diagrammasi nimani aks ettiradi?</b>
	<i>ob'ektlar va xabarlar</i>
	<i>Ob'ektlar va havolalar</i>
	<i>sinflar va munosabatlar</i>
	<i>Holatlar va hodisalar</i>

<b>13</b>	<b>UML to'plami nima?</b>
	<i>model elementlarining guruhlanishi</i>
	<i>quti</i>
	<i>sinflarni guruhlash</i>
	<i>foydalanish holatlarini guruhlash</i>

<b>14</b>	<b>Dasturiy ta'minotning hayotiy aylanishi nima?</b>
	<i>dasturiy ta'minotni ishlab chiqishga bosqichma-bosqich yondashish</i>
	<i>dasturiy ta'minotning ishlash muddati</i>
	<i>dasturiy ta'minotni ishlab chiqishda davriy yondashuv</i>
	<i>sikllarda ishlab chiqilgan dasturiy ta'minotning ishlash muddati</i>

<b>15</b>	<b>Sharshara hayotiy siklining modeli qanday?</b>
	<i>har bir bosqich keyingi bosqich boshlanishidan oldin yakunlanadigan jarayon modeli</i>
	<i>sharshara ostida ishlab chiqilgan dasturiy ta'minot</i>
	<i>fazalar ustma-ust tushadigan jarayon modeli</i>
	<i>fazalar davriy bo'lgan jarayon modeli</i>

<b>16</b>	<b>Quyidagilardan qaysi biri ob'ektga yo'naltirilgan tushunchalar?</b>
	<i>sinflar, ma'lumotni yashirish va meros</i>
	<i>modullar va interfeyslar</i>
	<i>modullar va ma'lumotlarni yashirish</i>
	<i>bir xillik va ma'lumotni yashirish</i>

<b>17</b>	<b>Quyidagilardan qaysi biri ob'ektga xos xususiyatdir?</b>
	<i>ma'lumotlar asosida ishlaydigan ma'lumotlar va protseduralarni guruhlariga ajratish</i>
	<i>funktsiya yoki pastki dastur</i>

	<i>modul</i>
	<i>funktsiya va algoritmi guruhlariga ajratadi</i>

<b>18</b>	<b>Sinf nima?</b>
	<i>bir xil xususiyatlarga ega bo'lgan ob'ektlar to'plami</i>
	<i>ob'ekt nusxasi</i>
	<i>ob'ektni amalga oshirish</i>
	<i>turli xil xususiyatlarga ega bo'lgan ob'ektlar to'plami</i>

<b>19</b>	<b>Use case nima?</b>
	<i>foydalanuvchi va tizimdagi ob'ektlar o'rtasidagi o'zaro aloqalar ketma-ketligi</i>
	<i>foydalanuvchilar bilan bog'liq bo'lgan amaliy ish</i>
	<i>foydalanuvchi va tizim o'rtasidagi o'zaro ta'sirlar ketma-ketligi</i>
	<i>foydalanuvchini tizimga kiritadigan ketma-ketligi</i>

<b>20</b>	<b>Use casedagi aktyor nima?</b>
	<i>tizim bilan o'zaro aloqada bo'lgan tashqi shaxs turi</i>
	<i>tizim ichidagi ob'ekt</i>
	<i>sahnada ijro etadigan shaxs</i>
	<i>tizim etkazib beriladigan mijoz</i>

<b>21</b>	<b>UMLda asosiy aktyor nima?</b>
	<i>foydalanish holatini boshlaydigan aktyor</i>
	<i>Avval sahnaga chiqqan aktyor</i>
	<i>foydalanish holatida qatnashadigan aktyor</i>
	<i>tizim ichidagi ob'ekt</i>

<b>22</b>	<b>UMLda ikkilamchi aktyor nima?</b>
	<i>foydalanish holatida qatnashadigan aktyor</i>
	<i>Ikkinchi sahnaga chiqqan aktyor</i>
	<i>foydalanish holatini boshlaydigan aktyor</i>
	<i>tizim ichidagi ob'ekt</i>

<b>23</b>	<b>Dasturiy ta'minot hayot siklining nechta modeli mavjud?</b>
	<i>5</i>
	<i>3</i>
	<i>4</i>
	<i>6</i>

<b>24</b>	<b>Eng ko'p ishlatiladigan modellar qaysilar?</b>
	<i>Kaskad, ortib boruvchi, spiral</i>
	<i>Spiral, v shaklidagi, iterative</i>
	<i>Iterative, kaska, spiral</i>
	<i>Kaskad, V shaklidagi model, ortib boruvchi</i>

<b>25</b>	<b>Quyidagi ta'rif qaysi modelga tegishli? O'zgartirib bo'lmaydigan ma'lum bir ketma-ketlikda sodir bo'ladigan bosqichlar darajalarining ketma-ket o'zgaruvchan tsiklidir.</b>
	<i>Kaskad</i>
	<i>Spiral</i>
	<i>V shakldagi model</i>
	<i>Iterative</i>

26	Quyidagi ta'rif qaysi modelga tegishli? ... model harakatlarning chiziqli ketma-ketligini, bosqichma-bosqich fikr-mulohazani va natijalarni nazorat qilishni o'z ichiga oladi.
	<i>Ortib boruvchi</i>
	<i>Kaskad</i>
	<i>Spiral</i>
	<i>V shakldagi model</i>

27	Repository Design Pattern kodni saqlashni qanday soddalashtiradi?
	<i>Barcha ma'lumotlar operatsiyalarini bir joyda boshqarish orqali</i>
	<i>Bir nechta dasturlash tillarini birlashtirish orqali</i>
	<i>Uskunaning ishlashini optimallashtirish orqali</i>
	<i>Foydalanuvchi kiritishlarini avtomatlashtirish orqali</i>

28	Repository Design Pattern toza kodga qanday hissa qo'shadi?
	<i>Ma'lumotlarga kirish mantiqini izolyatsiya qilish orqali</i>
	<i>Barcha mantiqni bir modulga birlashtirish orqali</i>
	<i>Kod qatorlarini ko'paytirish orqali</i>
	<i>Ma'lumotlarga kirishni biznes mantig'i bilan aralashtirish orqali</i>

29	Repository Design Pattern izchil interfeysni qanday ta'minlay oladi?
	<i>Ma'lumotlarga kirish</i>
	<i>Foydalanuvchi kiritishi</i>
	<i>Tarmoq trafigi</i>
	<i>Tizim xotirasi</i>

30	Ma'lumotlar bazalariga ma'lumotlarga kirishni boshqarish uchun "Repository Design Pattern" odatda qaysi turdagi ilovalarda qo'llaniladi?
	<i>Veb-ilovalar</i>
	<i>Ish stoli o'yinlari</i>
	<i>Mobil operatsion tizimlar</i>
	<i>O'rnatilgan tizimlar</i>

31	Repository Design Pattern veb-ilovalarda migratsiyani qanday osonlashtiradi?
	<i>Ma'lumotlar bazasi bilan o'zaro aloqani mavhumlash orqali</i>
	<i>Foydalanuvchi interfeysi bilan bevosita integratsiyalashgan holda</i>
	<i>Apparat resurslarini boshqarish orqali</i>
	<i>Tarmoq protokolini soddalashtirish orqali</i>

32	Repository Design Pattern API va mikrosvislarni yaratishda qanday rol o'ynaydi?
	<i>Bu ma'lumotlarga kirishni toza va tartibli boshqarishga yordam beradi</i>
	<i>U bevosita foydalanuvchi autentifikatsiyasini boshqaradi</i>
	<i>Foydalanuvchining grafik interfeyslarini boshqaradi</i>
	<i>Tarmoq sozlamalarini sozlaydi</i>

33	MVC dasturiy ta'minot arxitekturasida nimani anglatadi?
	<i>Model-View-Controller</i>
	<i>Main-View-Component</i>
	<i>Module-View-Connection</i>
	<i>Manage-Visualize-Control</i>

<b>34</b>	<b>MVC qanday dizayn namunasi hisoblanadi?</b>
	<i>Arxitektura loyihalash namunasi</i>
	<i>Yaratuvchi dizayn namunasi</i>
	<i>Konstruktiv loyihalash namunasi</i>
	<i>Xulq-atvorni loyihalash namunasi</i>

<b>35</b>	<b>MVC dastur strukturasi qanday soddalashtiradi?</b>
	<i>Loyihani uchta asosiy qatlamga bo'lish orqali</i>
	<i>Barcha komponentlarni bir qatlamga birlashtirish orqali</i>
	<i>Ma'lumotlar bazalaridan foydalanishni bekor qilish orqali</i>
	<i>Barcha funktsionallikni qattiq kodlash orqali</i>

<b>36</b>	<b>Controller MVC da View bilan qanday o'zaro ta'sir qiladi?</b>
	<i>Ko'rinishni tanlash va unga modelni yuborish orqali</i>
	<i>Ko'rinishni bevosita o'zgartirish orqali</i>
	<i>Ko'rinish holatini saqlash orqali</i>
	<i>Ko'rish hodisalarini boshqarish orqali</i>

<b>37</b>	<b>MVC ning qaysi komponenti foydalanuvchi kiritgan ma'lumotlarni qayta ishlash va uni model yoki View uchun buyruqlarga aylantirish uchun javobgardir?</b>
	<i>Controller</i>
	<i>Model</i>
	<i>Ko'rish</i>
	<i>Ma'lumotlar bazasi</i>

<b>38</b>	<b>MVC dasturiy ta'minotni loyihalashda tashvishlarni ajratishga qanday hissa qo'shadi?</b>
	<i>Ilovani uchta alohida qatlamga bo'lish orqali</i>
	<i>Model va ko'rinishni bir qatlamga birlashtirish orqali</i>
	<i>Faqat foydalanuvchi interfeysiga e'tibor qaratish orqali</i>
	<i>Kontrollerdagi barcha mantiqni markazlashtirish orqali</i>

<b>39</b>	<b>MVC ning qaysi qatlami foydalanuvchi so'rovlarini bajaradi?</b>
	<i>Controller</i>
	<i>Model</i>
	<i>View</i>
	<i>Ma'lumotlar bazasi</i>

<b>40</b>	<b>MVC da boshqaruvchining asosiy javobgarligi nima?</b>
	<i>Model bilan ishlash</i>
	<i>Foydalanuvchiga ma'lumotlarni ko'rsatish</i>
	<i>Ma'lumotlarni doimiy ravishda saqlash</i>
	<i>Foydalanuvchi autentifikatsiyasini boshqarish</i>

<b>41</b>	<b>Controller MVC da Viewni qanday yangilaydi?</b>
	<i>Model ma'lumotlarini ko'rinishga yuborish orqali</i>
	<i>Ko'rinish kodini bevosita o'zgartirish orqali</i>
	<i>Ko'rish ma'lumotlarini saqlash orqali</i>
	<i>Ma'lumotlar bazasi sxemasini manipulyatsiya qilish orqali</i>

<b>42</b>	<b>MVC ning qaysi komponenti bilan foydalanuvchi bevosita o'zaro aloqada bo'ladi?</b>
	<i>View</i>

	<i>Model</i>
	<i>Controller</i>
	<i>Ma'lumotlar bazasi</i>

<b>43</b>	<b>Real vaqt arxitekturasining belgilovchi xususiyati nima?</b>
	<i>Voqealarga zudlik bilan munosabat bildirish</i>
	<i>Ma'lumotlarni qayta ishlashning kechikishi</i>
	<i>Faqat partiyalarni qayta ishlash</i>
	<i>Asinxron aloqa</i>

<b>44</b>	<b>Dasturiy ta'minotni ishlab chiqishda MVC dan foydalanishning asosiy afzalliklaridan biri nimada?</b>
	<i>Konsernlarni ajratish orqali soddalashtirilgan texnik xizmat ko'rsatish</i>
	<i>Ma'lumotlar bazasini shifrlash orqali xavfsizlikni kuchaytirish</i>
	<i>Bir qatlamdan foydalanish orqali unumdorlikni oshirish</i>
	<i>Komponentlarni birlashtirish orqali qisqartirilgan kod</i>

<b>45</b>	<b>Qanday turdagi tizimlar odatda real vaqt arxitekturasidan foydalanadi?</b>
	<i>Hodisalarga zudlik bilan javob berishni talab qiluvchi tizimlar</i>
	<i>Vaqt cheklovlari bo'lmagan tizimlar</i>
	<i>Partiyali qayta ishlashga tayanuvchi tizimlar</i>
	<i>Ma'lumotlarning kelishi vaqtini oldindan aytib bo'lmaydigan tizimlar</i>

<b>46</b>	<b>Real vaqt arxitekturasining qaysi komponenti ma'lumotlarni o'z vaqtida qayta ishlashni ta'minlaydi?</b>
	<i>Voqealarga asoslangan ishlov berish</i>
	<i>Rejali partiyalarni qayta ishlash</i>
	<i>Ma'lumotlarning qo'lda aralashuvi</i>
	<i>Ma'lumotlarni tasodifiy qayta ishlash</i>

<b>47</b>	<b>Real vaqt arxitekturasida past kechikish nimani anglatadi?</b>
	<i>Ma'lumotlarni qayta ishlashda minimal kechikish</i>
	<i>Ma'lumotlarni qayta ishlashning sezilarli kechikishi</i>
	<i>Ma'lumotlarni ma'lum vaqt oralig'ida qayta ishlash</i>
	<i>Asinxron ma'lumotlarni qayta ishlash</i>

<b>48</b>	<b>Real vaqt arxitekturasida qaysi aloqa turi ko'p qo'llaniladi?</b>
	<i>Sinxron aloqa</i>
	<i>Asinxron aloqa</i>
	<i>Paketli aloqa</i>
	<i>Qo'l bilan aloqa qilish</i>

<b>49</b>	<b>Hodisaga asoslangan ishlov berish real vaqt arxitekturasida qanday rol o'ynaydi?</b>
	<i>Voqea sodir bo'lganda darhol qayta ishlashni ishga tushiradi</i>
	<i>Partiyalarni qayta ishlashni ma'lum vaqt oralig'ida rejalashtiradi</i>
	<i>Ma'lumotlarni qayta ishlashni keyingi vaqtgacha kechiktiradi</i>
	<i>U tasodifiy ma'lumotlarni qayta ishlaydi</i>

<b>50</b>	<b>Real vaqt arxitekturasini paketli ishlov berish tizimlaridan nimasi bilan farq qiladi?</b>
	<i>Voqealarga zudlik bilan munosabat bildirish</i>
	<i>Ma'lumotlarni qayta ishlashning kechikishi</i>
	<i>Faqat ketma-ket ishlov berish</i>
	<i>Ma'lumotlarni qayta ishlashga qo'l aralashuvi</i>

<b>51</b>	<b>Ma'lumotlarni qayta ishlashda real vaqt arxitekturasining asosiy afzalligi nimada?</b>
	<i>Hodisalarga oniy munosabat</i>
	<i>Hodisalarga kechikish munosabati</i>
	<i>Ma'lumotlarni ketma-ket qayta ishlash</i>
	<i>Ma'lumotlarning qo'lda aralashuvi</i>

<b>52</b>	<b>Qaysi komponent real vaqtda tizimlar ma'lumotlar kelishi bilan oq qayta ishlanishini ta'minlaydi?</b>
	<i>Voqealarga asoslangan ishlov berish</i>
	<i>Partiyalarni qayta ishlash</i>
	<i>Qo'lda ishlov berish</i>
	<i>Asinxron ishlov berish</i>

<b>53</b>	<b>MVCda nechta turdagi natijalar mavjud?</b>
	<i>12</i>
	<i>10</i>
	<i>11</i>
	<i>13</i>

<b>54</b>	<b>Model-view-controller-da (MVArchitecture, model _____ ni belgilaydi.</b>
	<i>Biznes-mantiqiy qatlam</i>
	<i>Ma'lumotlarga kirish qatlami</i>
	<i>Taqdimot qatlami</i>
	<i>Interfeys qatlami</i>

<b>55</b>	<b>MVC uchta komponentdan iborat. Bular?</b>
	<i>Model View Controller</i>
	<i>Member Vertical Controller</i>
	<i>Model Variable Centered</i>
	<i>Member Vertical Controller</i>

<b>56</b>	<b>Qat'iylik qatlami nima qilish kerak</b>
	<i>Qat'iylik qatlami texnik ishlarni bajarish uchun mas'ul bo'lgan barcha sinflarni o'z ichiga oladi.</i>
	<i>Doimiylik qatlami oxirgi foydalanuvchiga UI taqdim etish uchun mas'ul bo'lgan barcha sinflarni o'z ichiga oladi.</i>
	<i>Qat'iylik qatlami ilovaning funktsional talablariga javob berish uchun talab qiladigan barcha mantiqni o'z ichiga oladi</i>
	<i>Qat'iylik qatlami asosiy domenni ifodalaydi</i>

<b>57</b>	<b>Qatlamli arxitektura odatda bir necha qatlamlardan iborat. Ular nechta?</b>
	<i>4</i>
	<i>1</i>
	<i>2</i>
	<i>15</i>

<b>58</b>	<b>Qatlamlar orasidagi bog'lanish qanday?</b>
	<i>Yuqoridan pastga</i>
	<i>Pastdan tepaga</i>
	<i>Har qanday yo'nalishda</i>
	<i>Chapdan o'ngga</i>

<b>59</b>	<b>Qatlamli arxitektura nimadan yasalgan?</b>
	<i>Qatlamlar</i>
	<i>Qatorlar</i>
	<i>Ustunlar</i>
	<i>Kataklar</i>

<b>60</b>	<b>Qaysi arxitektura ishlab chiquvchilar moslashuvchan va qayta foydalanish mumkin bo'lgan ilovalarni yaratishi modelni taqdim etadi?</b>
	<i>N-darajali ilovalar arxitekturasi</i>
	<i>p2p arxitektura</i>
	<i>MVC arxitektura</i>
	<i>Mikroservis arxitekturasi.</i>

<b>61</b>	<b>Ob'ektga yo'naltirilgan dizaynga ega axborot tizimi uchun mantiqiy ko'p qatlamli arxitekturada eng keng tarqalgan turlari nechta?</b>
	<i>4ta</i>
	<i>3ta</i>
	<i>2ta</i>
	<i>5ta</i>

<b>62</b>	<b>Taqsimlangan tarmoqdagi uch yoki undan ortiq alohida kompyuterlar orasida taqsimlangan dastur?</b>
	<i>N -Tier Application</i>
	<i>n-darajali arxitektura</i>
	<i>CurrencyConverter</i>
	<i>BI Application</i>

<b>63</b>	<b>Sanoat tomonidan tasdiqlangan dasturiy ta'minot arxitekturasi modeli?</b>
	<i>n-darajali arxitektura</i>
	<i>p2p arxitektura</i>
	<i>MVC arxitektura</i>
	<i>Mikroservis arxitektura</i>

<b>64</b>	<b>Qaysi komponent real vaqtda tizimlar ma'lumotlar kelishi bilan oq qayta ishlanishini ta'minlaydi?</b>
	<i>Voqealarga asoslangan ishlov berish</i>
	<i>Partiyalarni qayta ishlash</i>
	<i>Qo'lda ishlov berish</i>
	<i>Asinxron ishlov berish</i>

<b>65</b>	<b>Qaysi xususiyat real vaqt tizimlarining javob berish qobiliyatini belgilaydi?</b>
	<i>Past kechikish</i>
	<i>Yuqori kechikish</i>
	<i>Partiyalarni qayta ishlash</i>
	<i>Tasodifiy ishlov berish</i>

<b>66</b>	<b>IT infratuzilmasida peer-to-peer (P2P) arxitektura naqshini nima belgilaydi?</b>
	<i>Ikki yoki undan ortiq kompyuter tizimlarini alohida serverlarni talab qilmasdan resurslarni ulash va almashish imkonini berish</i>
	<i>Yagona serverda resurslarni markazlashtirish</i>
	<i>Resurslarni almashish uchun virtual maydonlarni yaratish</i>
	<i>Muayyan serverlarga resurslardan foydalanishni cheklash</i>



<b>67</b>	<b>P2P arxitekturasi an'anaviy mijoz-server modelidan nimasi bilan farq qiladi?</b>
	<i>Resurs almashish uchun kompyuterlarni bevosita bog'laydi</i>
	<i>U barcha resurslarni almashish uchun bitta serverga tayanadi</i>
	<i>Har bir resurs uchun alohida serverlar talab qilinadi</i>
	<i>U ma'lum serverlar bilan resurslar almashishni cheklaydi</i>

<b>68</b>	<b>Peer-to-peering o'ziga xos xususiyati nimada?</b>
	<i>Markaziy serverlarsiz resurslarni almashish uchun kompyuterlarni ulash</i>
	<i>Faqat server resurslariga tayanish</i>
	<i>Har bir kompyuter uchun virtual maydonlar yaratish</i>
	<i>Serverga asoslangan tarmoqlarga resurslardan foydalanishni cheklash</i>

<b>69</b>	<b>Peer-to-peer arxitekturasi resurslarni almashish uchun kompyuterlar qanday o'zaro ta'sir qiladi?</b>
	<i>Markaziy serverlarsiz resurslarni bevosita ulash va almashish</i>
	<i>Resursga kirish uchun markaziy server bilan muloqot qilish</i>
	<i>Resurslarni almashish uchun virtual maydonlarni yaratish</i>
	<i>Resurslarga faqat serverga asoslangan tarmoqlar orqali kirish</i>

<b>70</b>	<b>Peer-to-peer arxitekturasi mijoz-server modelidan nimasi bilan farq qiladi?</b>
	<i>Markaziy serversiz kompyuterlar o'rtasida to'g'ridan-to'g'ri resurslar almashish</i>
	<i>Server resurslariga eksklyuziv tayanish</i>
	<i>Resurslarni almashish uchun virtual maydonlarni yaratish</i>
	<i>Maxsus serverlar orqali resurslardan foydalanishning cheklanganligi</i>

<b>71</b>	<b>Peer-to-peer arxitekturasi alohida serverlarni talab qilmasdan nimani beradi?</b>
	<i>Kompyuterlar o'rtasida to'g'ridan-to'g'ri ulanish va resurslarni taqsimlash</i>
	<i>Server resurslariga eksklyuziv tayanish</i>
	<i>Resurslarni izolyatsiyalash uchun virtual maydonlarni yaratish</i>
	<i>Resurslardan foydalanishni markazlashtirilgan nazorat qilish</i>

<b>72</b>	<b>Peer-to-peer arxitekturasi kompyuterlar o'rtasida resurslar almashinuvini qanday boshqaradi?</b>
	<i>Markaziy serversiz kompyuterlar o'rtasida to'g'ridan-to'g'ri ulanishni ta'minlash orqali</i>
	<i>Barcha resurs so'rovlarini bitta server orqali yo'naltirish orqali</i>
	<i>Har bir kompyuter uchun virtual maydonlar yaratish orqali</i>
	<i>Muayyan serverlarga resurslardan foydalanishni cheklash orqali</i>

<b>73</b>	<b>Peer-to-peer arxitekturasi asosiy xususiyati nimada?</b>
	<i>Kompyuterlar o'rtasida to'g'ridan-to'g'ri resurslar almashinuvi</i>
	<i>Server resurslariga eksklyuziv tayanish</i>
	<i>Resurslarni ajratish uchun virtual maydonlarni yaratish</i>
	<i>Resurslarni taqsimlash ustidan markazlashgan nazorat</i>

<b>74</b>	<b>Peer-to-peer arxitekturasi an'anaviy tarmoq arxitekturasidan nimasi bilan farq qiladi?</b>
	<i>Markaziy serversiz kompyuterlar o'rtasida to'g'ridan-to'g'ri resurslar almashish</i>
	<i>Faqatgina server resurslariga tayanish</i>
	<i>Resurslarni taqsimlash uchun virtual maydonlarni yaratish</i>
	<i>Maxsus serverlar orqali resurslardan foydalanishning cheklanganligi</i>

<b>75</b>	<b>Kompyuterlar resurs almashish uchun Peer-to-peer arxitekturasi qanday o'zaro ta'sir qiladi?</b>
	<i>Ular markaziy serverlarsiz resurslarni bevosita ulashadi va almashadilar</i>
	<i>Ular barcha resurslarga kirish uchun bitta serverga tayanadilar</i>

	<i>Ular resurslarni izolyatsiyalash uchun virtual bo'shliqlarni yaratadilar</i>
	<i>Ular resurslarga faqat serverga asoslangan tarmoqlar orqali kirishadi</i>
<b>76</b>	<b>Peer-to-peer arxitekturasidagi kompyuterlarga markaziy serverlarsiz resurslarni almashishga nima imkon beradi?</b>
	<i>Kompyuterlar o'rtasida to'g'ridan-to'g'ri ulanish</i>
	<i>Server resurslariga eksklyuziv tayanish</i>
	<i>Har bir kompyuter uchun virtual maydonlar yaratish</i>
	<i>Resurslarni taqsimlash ustidan markazlashgan nazorat</i>
<b>77</b>	<b>Peer-to-peerning asosiy tamoyilini nima belgilaydi?</b>
	<i>Kompyuterlar o'rtasida to'g'ridan-to'g'ri resurslar almashinuvini ta'minlash</i>
	<i>Faqat server resurslariga tayanish</i>
	<i>Resurslarni ajratish uchun virtual maydonlarni yaratish</i>
	<i>Resurs taqsimoti ustidan markazlashgan nazorat</i>
<b>78</b>	<b>Peer-to-peer arxitektura modeli mijoz-server modelidan qanday farq qiladi?</b>
	<i>Kompyuterlar o'rtasida to'g'ridan-to'g'ri resurslar almashish imkonini beradi</i>
	<i>Yagona server orqali resurslarga kirishni markazlashtiradi</i>
	<i>Resurslarni izolyatsiyalash uchun virtual joylarni yaratadi</i>
	<i>Muayyan serverlar uchun resurslarni taqsimlashni cheklaydi</i>
<b>79</b>	<b>Peer-to-peerning o'ziga xos xususiyati nimada?</b>
	<i>Kompyuterlar o'rtasida to'g'ridan-to'g'ri resurslar almashinuvini ta'minlash</i>
	<i>Faqat server resurslariga tayanish</i>
	<i>Har bir kompyuter uchun virtual maydonlar yaratish</i>
	<i>Yagona server orqali resurslarga kirishni markazlashtirish</i>
<b>80</b>	<b>P2P tarmog'ining asosiy xususiyati nima?</b>
	<i>Har bir kompyuter resurslarni ta'minlaydi va iste'mol qiladi</i>
	<i>Faqat bitta kompyuter resurslarni taqdim etadi</i>
	<i>Resurslar hissa qo'shmasdan iste'mol qilinadi</i>
	<i>Resurslar bir kompyuterda markazlashtirilgan</i>
<b>81</b>	<b>P2P tarmog'ida qanday turdagi resurslarni taqsimlash mumkin?</b>
	<i>Fayllar, printerlar, saqlash, tarmoqli kengligi va ishlov berish quvvati</i>
	<i>Faqat fayllar va printerlar</i>
	<i>Faqat saqlash va tarmoqli kengligi</i>
	<i>Faqat ishlov berish quvvati va printerlar</i>
<b>82</b>	<b>P2P tarmog'ida kirish qanday boshqariladi?</b>
	<i>Ruxsatlarni almashish va parollarni belgilash orqali</i>
	<i>Tarmoq tomonidan avtomatik ravishda</i>
	<i>Jismoniy yaqinlik bilan</i>
	<i>Faqat tarmoq administratorlari tomonidan</i>
<b>83</b>	<b>Komponentga asoslangan arxitekturaning (CBA) asosiy yo'nalishi nima?</b>
	<i>Komponentlar deb ataladigan qayta foydalanish mumkin bo'lgan mustaqil dasturiy birliklarni yaratish</i>
	<i>Monolitik dasturiy ilovalarni ishlab chiqish</i>
	<i>Murakkab dasturiy tizimlarni noldan qurish</i>
	<i>Oldindan belgilangan dasturiy ta'minot kutubxonalaridan foydalanish</i>

<b>84</b>	<b>Komponentga asoslangan arxitekturada inkapsulyatsiya qanday rol o'ynaydi?</b>
	<i>Komponentlar ichki amalga oshirish detallarini yashiradi</i>
	<i>Komponentlar amalga oshirishning barcha tafsilotlarini ochib beradi</i>
	<i>Komponentlarda inkapsulyatsiya mavjud emas</i>
	<i>Komponentlar amalga oshirish tafsilotlarini tanlab ochib beradi</i>

<b>85</b>	<b>Komponentga asoslangan arxitekturada yaxshi ishlab chiqilgan komponentlarning belgilovchi xususiyati nimada?</b>
	<i>Bir tizimning bir nechta ilovalari yoki qismlarida qayta foydalanish mumkinligi</i>
	<i>Boshqa komponentlardan izolyatsiya</i>
	<i>Amalga oshirishning barcha tafsilotlarini oshkor qilish</i>
	<i>Faqat aniq ilovalarga bog'liqlik</i>

<b>86</b>	<b>Komponentga asoslangan arxitekturada komponentlar bir-biri bilan qanday o'zaro ta'sir qiladi?</b>
	<i>Aniq belgilangan interfeyslar orqali</i>
	<i>Interfeyslarsiz</i>
	<i>Ichki amalga oshirish tafsilotlarini fosh qilish orqali</i>
	<i>Barcha xossa va usullarni ulashish orqali</i>

<b>87</b>	<b>Komponentga asoslangan arxitekturada interfeysning maqsadi nima?</b>
	<i>Komponentlarning o'zaro ta'sirining usullari va xususiyatlarini aniqlash</i>
	<i>Komponent funksiyalarini yashirish</i>
	<i>Ichki amalga oshirish tafsilotlarini oshkor qilish</i>
	<i>Komponentlarning qayta ishlatilishini cheklash</i>

<b>88</b>	<b>Qanday xususiyat komponentlarni CBAda turli konfiguratsiyalarda yig'ish imkonini beradi?</b>
	<i>Tarkibi</i>
	<i>Inkapsulyatsiya</i>
	<i>Meros</i>
	<i>Abstraktsiya</i>

<b>89</b>	<b>Komponentga asoslangan arxitekturada qayta foydalanishga qanday yordam beradi?</b>
	<i>Yaxshi ishlab chiqilgan komponentlar bir xil tizimning bir nechta ilovalarida yoki qismlarida ishlatilishi mumkin.</i>
	<i>Komponentlarda qayta foydalanish mumkin emas.</i>
	<i>Qayta foydalanish mumkinligi aniq ilovalar bilan chegaralanadi.</i>
	<i>CBAda qayta foydalanish mumkinligi tashvishlanmaydi.</i>

<b>90</b>	<b>Komponentga asoslangan arxitekturada kompozitsiya qanday afzalliklarga ega?</b>
	<i>Komponentlarni turli xil konfiguratsiyalarda yig'ish mumkin.</i>
	<i>Komponentlarni yig'ib bo'lmaydi.</i>
	<i>Komponentlarda kompozitsiya mavjud emas.</i>
	<i>Komponentlar qattiq konfiguratsiyaga ega.</i>

<b>91</b>	<b>Komponentga asoslangan arxitekturaning o'zgaruvchanlik nuqtai nazaridan asosiy xususiyati nimada?</b>
	<i>Alohida komponentlar butun tizimga ta'sir qilmasdan almashtirilishi yoki yangilanishi mumkin.</i>
	<i>Bir komponentni o'zgartirish butun tizimga ta'sir qiladi.</i>
	<i>Komponentlarni almashtirish yoki yangilash mumkin emas.</i>
	<i>Bitta komponentni yangilash barcha komponentlarni yangilashni talab qiladi.</i>

<b>92</b>	<b>Komponentga asoslangan arxitekturada sinovdan o'tish qobiliyatini qanday oshiradi?</b>
-----------	---

	<i>Komponentlarni alohida tekshirish mumkin.</i>
	<i>Komponentlarni sinash mumkin emas.</i>
	<i>Sinov butun tizimning ishlashini talab qiladi.</i>
	<i>Tekshiruv muayyan komponentlar bilan chegaralanadi.</i>

<b>93</b>	<b>Komponentga asoslangan arxitektura (CBA) ning asosiy maqsadi nima?</b>
	<i>Qayta foydalanish mumkin bo'lgan mustaqil dasturiy komponentlarni yaratish.</i>
	<i>Monolitik dasturiy ilovalarni ishlab chiqish.</i>
	<i>Murakkab dasturiy tizimlarni noldan qurish.</i>
	<i>Oldindan belgilangan dasturiy kutubxonalardan foydalanish.</i>

<b>94</b>	<b>Komponentga asoslangan arxitekturada inkapsulyatsiya nimani o'z ichiga oladi?</b>
	<i>Komponentlar ichki amalga oshirish detallarini yashiradi.</i>
	<i>Komponentlar amalga oshirishning barcha tafsilotlarini ochib beradi.</i>
	<i>Komponentlarda inkapsulyatsiya mavjud emas.</i>
	<i>Komponentlar amalga oshirish tafsilotlarini tanlab ochib beradi.</i>

<b>95</b>	<b>Komponentga asoslangan arxitekturada yaxshi ishlab chiqilgan komponentlarga xos xususiyat nimada?</b>
	<i>Bir tizimning bir nechta ilovalari yoki qismlarida qayta foydalanish mumkinligi.</i>
	<i>Boshqa komponentlardan izolyatsiya.</i>
	<i>Amalga oshirishning barcha tafsilotlarini oshkor qilish.</i>
	<i>Faqat aniq ilovalarga bog'liqlik.</i>

<b>96</b>	<b>Komponentlar Komponentga asoslangan arxitekturada bir-biri bilan qanday aloqa qiladi?</b>
	<i>Aniq belgilangan interfeyslar orqali.</i>
	<i>Interfeyslarsiz.</i>
	<i>Ichki amalga oshirish tafsilotlarini fosh qilish orqali.</i>
	<i>Barcha xossa va usullarni ulashish orqali.</i>

<b>97</b>	<b>Komponentlarga asoslangan arxitekturada interfeys qanday vazifani bajaradi?</b>
	<i>Komponentlarning o'zaro ta'siri usullari va xususiyatlarini aniqlash.</i>
	<i>Komponent funksiyalarini yashirish uchun.</i>
	<i>Ichki amalga oshirish tafsilotlarini oshkor qilish.</i>
	<i>Komponentlarning qayta ishlatilishini cheklash.</i>

<b>98</b>	<b>Qanday xususiyat CBAda komponentlarni moslashuvchan yig'ish imkonini beradi?</b>
	<i>Tarkibi.</i>
	<i>Inkapsulyatsiya.</i>
	<i>Meros.</i>
	<i>Abstraktsiya.</i>

<b>99</b>	<b>Komponentga asoslangan arxitekturada qayta foydalanishni qanday osonlashtiradi?</b>
	<i>Yaxshi ishlab chiqilgan komponentlar bir xil tizimning bir nechta ilovalarida yoki qismlarida qayta ishlatilishi mumkin.</i>
	<i>Komponentlarda qayta foydalanish mumkin emas.</i>
	<i>Qayta foydalanish mumkinligi aniq ilovalar bilan chegaralanadi.</i>
	<i>CBAda qayta foydalanish imkoniyati hisobga olinmaydi.</i>

<b>100</b>	<b>Komponentga asoslangan arxitekturada kompozitsiya qanday afzalliklarga ega?</b>
	<i>Komponentlarni turli xil konfiguratsiyalarda yig'ish mumkin.</i>
	<i>Komponentlarni yig'ib bo'lmaydi.</i>

	<i>Komponentlarda kompozitsiya mavjud emas.</i>
	<i>Komponentlar qattiq konfiguratsiyaga ega.</i>
<b>101</b>	<b>Komponentga asoslangan arxitektura (CBA) ishlab chiqish vaqtini va resurslarini tejashga nima yordam beradi?</b>
	<i>Komponentlarning qayta ishlatilishining ortishi.</i>
	<i>Komponentlarning qayta ishlatilishining kamayishi.</i>
	<i>Komponentlar almashinuvining cheklanganligi.</i>
	<i>Murakkab komponentlar kutubxonalari.</i>
<b>102</b>	<b>Komponentga asoslangan arxitekturada ilovalarning texnik xizmat ko'rsatish qobiliyatini yaxshilashga qanday hissa qo'shadi?</b>
	<i>O'z-o'zidan tarkib topgan komponentlarga o'zgartirishlar kiritilishi mumkin.</i>
	<i>O'zgarishlar bir vaqtning o'zida butun tizimga ta'sir qiladi.</i>
	<i>Komponentlar o'zini tuta olmaydi, bu esa texnik xizmat ko'rsatishni qiyinlashtiradi.</i>
	<i>CBAda xizmat ko'rsatish qobiliyati o'zgarishsiz qoladi.</i>
<b>103</b>	<b>Komponentga asoslangan arxitekturada masshtablilik nuqtai nazaridan qanday afzalliklarni taklif qiladi?</b>
	<i>Komponentlar bir nechta serverlarga taqsimlanishi mumkin.</i>
	<i>Komponentlarni taqsimlab bo'lmaydi.</i>
	<i>Masshtablash katta kuch talab qiladi.</i>
	<i>Masshtablash muayyan komponentlar bilan chegaralanadi.</i>
<b>104</b>	<b>Komponentga asoslangan arxitekturada tezroq rivojlanish tsikllariga qanday hissa qo'shadi?</b>
	<i>Parallel rivojlanish imkonini berish orqali.</i>
	<i>Rivojlanish sikllarini sekinlashtirish orqali.</i>
	<i>Rivojlanishni ketma-ket jarayonlar bilan cheklash orqali.</i>
	<i>Jamoa hamkorligini susaytirish orqali.</i>
<b>105</b>	<b>Komponentga asoslangan arxitekturada o'zgaruvchan talablarga moslashish uchun moslashuvchanlik nuqtai nazaridan nimani beradi?</b>
	<i>Komponentlarni minimal buzilishlar bilan qo'shish, olib tashlash yoki almashtirish qobiliyati.</i>
	<i>Moslashuvchan bo'lmagan tizim tuzilishi.</i>
	<i>Tizimni jiddiy ta'mirlash talabi.</i>
	<i>O'zgaruvchan talablarga moslasha olmaslik.</i>
<b>106</b>	<b>Ishlab chiqish vaqti va resurslari nuqtai nazaridan CBAning asosiy afzalligi nimada?</b>
	<i>Komponentlarning qayta ishlatilishining ortishi.</i>
	<i>Komponentlarning qayta ishlatilishining kamayishi.</i>
	<i>Komponentlar almashinuvining cheklanganligi.</i>
	<i>Murakkab komponentlar kutubxonalari.</i>
<b>107</b>	<b>Komponentga asoslangan arxitekturada ilovalarning yaxshilanishiga qanday hissa qo'shadi?</b>
	<i>O'z-o'zidan tarkib topgan komponentlarga o'zgartirishlar kiritilishi mumkin.</i>
	<i>O'zgarishlar bir vaqtning o'zida butun tizimga ta'sir qiladi.</i>
	<i>Komponentlar o'zini tuta olmaydi, bu esa texnik xizmat ko'rsatishni qiyinlashtiradi.</i>
	<i>CBAda xizmat ko'rsatish qobiliyati o'zgarishsiz qoladi.</i>
<b>108</b>	<b>Komponentga asoslangan arxitekturada masshtablilik nuqtai nazaridan qanday afzalliklarga ega?</b>
	<i>Komponentlar bir nechta serverlarga taqsimlanishi mumkin.</i>
	<i>Komponentlarni taqsimlab bo'lmaydi.</i>

	<i>Masshtablash katta kuch talab qiladi.</i>
	<i>Masshtablash muayyan komponentlar bilan chegaralanadi.</i>
<b>109</b>	<b>Qanday qilib Komponentga asoslangan arxitekturada tezroq rivojlanish davrlarini osonlashtiradi?</b>
	<i>Parallel rivojlanish imkonini berish orqali.</i>
	<i>Rivojlanish sikllarini sekinlashtirish orqali.</i>
	<i>Rivojlanishni ketma-ket jarayonlar bilan cheklash orqali.</i>
	<i>Jamoa hamkorligini susaytirish orqali.</i>
<b>110</b>	<b>Komponentga asoslangan arxitekturada o'zgaruvchan talablarga moslashish uchun moslashuvchanlik nuqtai nazaridan nimani beradi?</b>
	<i>Komponentlarni minimal buzilishlar bilan qo'shish, olib tashlash yoki almashtirish qobiliyati.</i>
	<i>Moslashuvchan bo'lmagan tizim tuzilishi.</i>
	<i>Tizimni jiddiy ta'mirlash talabi.</i>
	<i>O'zgaruvchan talablarga moslasha olmaslik.</i>
<b>111</b>	<b>Xizmatga yo'naltirilgan arxitekturada (SOA) xizmatlarning belgilovchi xususiyati nima?</b>
	<i>Xizmatlar mustaqil ishlaydi va o'z iste'molchilariga funktsionallik yoki ma'lumotlar almashinuvini ta'minlaydi.</i>
	<i>Xizmatlar asosan markazlashgan boshqaruvga tayanadi.</i>
	<i>Xizmatlar iste'molchilar bilan chambarchas bog'langan.</i>
	<i>Xizmatlar bir-biridan ajratilgan.</i>
<b>112</b>	<b>Xizmatga yo'naltirilgan arxitekturada ma'lumotlarni uzatish protokollarining o'rnini qanday?</b>
	<i>Xizmatlar o'rtasida ma'lumotlarni uzatishning boshqaruv qoidalari.</i>
	<i>Ma'lumotlarni saqlashning boshqaruv qoidalari.</i>
	<i>Xizmat xatti-harakatlarini nazorat qilish.</i>
	<i>Xizmat interfeyslarini aniqlash.</i>
<b>113</b>	<b>Quyidagilardan qaysi biri Xizmatga yo'naltirilgan arxitekturani amalga oshirish uchun standart protokol emas?</b>
	<i>Masofaviy protsedura chaqiruvi (RPC)</i>
	<i>Ob'ektga oddiy kirish protokoli (SOAP)</i>
	<i>Apache ActiveMQ</i>
	<i>Java xabarlar xizmati (JMS)</i>
<b>114</b>	<b>SOAP SOA kontekstida nimani anglatadi?</b>
	<i>Obyektga kirishning oddiy protokoli</i>
	<i>Xizmatga yo'naltirilgan amaliy dasturlash</i>
	<i>Xizmat ob'ektiga kirish nuqtasi</i>
	<i>Tizim obyektlariga kirish protokoli</i>
<b>115</b>	<b>SOAda veb-xizmatlarni amalga oshirish uchun odatda qaysi protokoldan foydalaniladi?</b>
	<i>RESTful HTTP</i>
	<i>Apache tejamkorligi</i>
	<i>Java xabarlar xizmati (JMS)</i>
	<i>Masofaviy protsedura chaqiruvi (RPC)</i>
<b>116</b>	<b>SOAda Apache ActiveMQ ning asosiy vazifasi nima?</b>
	<i>Xabar navbati</i>
	<i>Ma'lumotlarni saqlash</i>
	<i>Autentifikatsiya</i>

	<i>Xizmatni kashf qilish</i>
<b>117</b>	<b>Qaysi protokol SOAda yengil yondashuvi va soddaligi bilan mashhur?</b>
	<i>RESTful HTTP</i>
	<i>Ob'ektga oddiy kirish protokoli (SOAP)</i>
	<i>Apache tejamkorligi</i>
	<i>Java xabarlar xizmati (JMS)</i>
<b>118</b>	<b>SOAda ma'lumotlarni uzatish protokollarining maqsadi nima?</b>
	<i>Xizmatlar o'rtasida ma'lumotlar almashinuvi qoidalarini belgilash.</i>
	<i>Xizmat funksiyalarini aniqlash uchun.</i>
	<i>Xizmatni joylashtirishni boshqarish uchun.</i>
	<i>Xavfsizlik siyosatini amalga oshirish uchun.</i>
<b>119</b>	<b>Ma'lumotlar almashinuvi bo'yicha SOA xizmatlarining o'ziga xos xususiyati nimada?</b>
	<i>Xizmatlar ma'lumotlarni qayta ishlaydi va iste'molchilar so'rovlariga javob yuboradi.</i>
	<i>Xizmatlar faqat iste'molchilardan ma'lumotlarni oladi.</i>
	<i>Xizmatlar ma'lumotlar almashinuvi uchun umumiy ma'lumotlar bazasini birlashtiradi.</i>
	<i>Xizmatlar ma'lumotlar almashinuvida ishtirok etmaydi.</i>
<b>120</b>	<b>Mikroservislar arxitekturasida mikroservislar nima belgilaydi?</b>
	<i>Juda kichik va butunlay mustaqil dasturiy komponentlar.</i>
	<i>Katta va o'zaro bog'liq dasturiy ta'minot komponentlari.</i>
	<i>O'rtacha kattalikdagi dasturiy komponentlar.</i>
	<i>Markazlashtirilgan dasturiy komponentlar.</i>
<b>121</b>	<b>Mikroservislar arxitekturasida mikroservislar qanday aloqa qiladi?</b>
	<i>API orqali.</i>
	<i>To'g'ridan-to'g'ri usul chaqiruvlari orqali.</i>
	<i>Umumiy ma'lumotlar bazalari orqali.</i>
	<i>Fayl almashish orqali.</i>
<b>122</b>	<b>Mikroservislarining arxitektura uslubi qaysi muhit uchun eng mos keladi?</b>
	<i>Zamonaviy bulutli hisoblash muhitlari.</i>
	<i>An'anaviy monolit muhitlar.</i>
	<i>Mahalliy ma'lumotlar markazlari.</i>
	<i>Mainfreyim muhitlari.</i>
<b>123</b>	<b>Mikroservislar uchun odatiy joylashtirish usuli qanday?</b>
	<i>Containers.</i>
	<i>Virtual mashinalar.</i>
	<i>Bare-metal servers.</i>
	<i>Umumiy xosting.</i>
<b>124</b>	<b>Mikroservislarining masshtablilik xususiyati nimada?</b>
	<i>Mustaqil ravishda kengaytiriladigan.</i>
	<i>Masshtablilik o'zaro bog'liqlikni talab qiladi.</i>
	<i>Masshtablilik chegaralangan.</i>
	<i>Masshtablilik markazlashtirilgan resurslarga bog'liq.</i>
<b>125</b>	<b>Mikroservislarining samaradorlik nuqtai nazaridan asosiy afzalligi nimada?</b>

	<i>Mustaqil ravishda kengaytiriladigan va tez.</i>
	<i>Sekin va o'zaro bog'liq.</i>
	<i>Faoliyatning cheklanganligi.</i>
	<i>Samaradorlik markazlashgan resurslarga bog'liq.</i>

<b>126</b>	<b>Mikroservislar an'anaviy arxitekturalarga nisbatan ma'lumotlarga kirishni qanday boshqaradi?</b>
	<i>Mikroservislar barcha kerakli ma'lumotlarga mahalliy kirish imkoniyatiga ega.</i>
	<i>Mikroservislar markazlashtirilgan ma'lumotlarga masofadan kirishga tayanadi.</i>
	<i>Mikroservislarda ma'lumotlarga kirish imkoni yo'q.</i>
	<i>Mikroservislar ma'lumotlarga umumiy ma'lumotlar bazalari orqali kirishadi.</i>

<b>127</b>	<b>Ma'lumotlarni takrorlash bo'yicha mikroservislarining o'ziga xos xususiyati nimada?</b>
	<i>Ishlash va moslashuvchanlikka erishish uchun ma'lumotlarni takrorlash mavjud.</i>
	<i>Ma'lumotlarning takrorlanishi har qanday holatda ham oldini oladi.</i>
	<i>Ma'lumotlarni takrorlash muayyan stsenariylar bilan chegaralanadi.</i>
	<i>Mikroservislar arxitekturasida ma'lumotlarni takrorlash muhim emas.</i>

<b>128</b>	<b>Mikroservislar o'rtasida bog'liqlik yo'qligi qaysi atama bilan tavsiflanadi?</b>
	<i>Ajratilgan.</i>
	<i>O'zaro bog'langan.</i>
	<i>qaram.</i>
	<i>Integratsiyalashgan.</i>

<b>129</b>	<b>Olti burchakli arxitektura yoki portlar va adapterlar arxitekturasining asosiy maqsadi nima?</b>
	<i>Erkin bog'langan dasturiy komponentlarni yaratish.</i>
	<i>Bir-biriga mahkam bog'langan dasturiy komponentlarni yaratish.</i>
	<i>Dasturiy ta'minot ulanishlarini minimallashtirish.</i>
	<i>Dasturiy ta'minotga bog'liqlikni maksimal darajada oshirish.</i>

<b>130</b>	<b>Hexagon arxitekturasi dasturiy ta'minotni loyihalashda nimaga erishmoqchi?</b>
	<i>Erkindan bog'langan dasturiy komponentlarni yaratish.</i>
	<i>Monolit dasturiy komponentlarni yaratish.</i>
	<i>Qatlamlar orasidagi bog'liqlikni oshirish.</i>
	<i>Foydalanuvchi interfeysi kodini biznes mantig'i bilan aralashtirish</i>

<b>131</b>	<b>Obyektga yo'naltirilgan dasturiy ta'minotni loyihalashda strukturaviy tuzoqlarni bartaraf etish uchun olti burchakli arxitekturani kim ixtiro qilgan?</b>
	<i>Alister Kokbern.</i>
	<i>Martin Fauler.</i>
	<i>Robert C. Martin.</i>
	<i>Kent Bek.</i>

<b>132</b>	<b>Olti burchakli arxitekturada port va adapterlarning o'rni qanday?</b>
	<i>Komponentlarni dasturiy muhitga osongina ulash imkonini berish.</i>
	<i>Komponentlar orasidagi aloqalarni cheklash.</i>
	<i>Sinovlarni avtomatlashtirishning oldini olish.</i>
	<i>Keraksiz qaramlikni rag'batlantirish</i>

<b>133</b>	<b>Hexagon arxitekturasi sinovlarni avtomatlashtirish uchun qanday foyda keltiradi?</b>
	<i>Sinovlarni avtomatlashtirishni osonlashtirish.</i>
	<i>Testlarni avtomatlashtirishga to'sqinlik qilish.</i>



	<i>Test murakkabligini oshirish.</i>
	<i>Avtomatlashtirishni sinash uchun ahamiyatsiz.</i>

<b>134</b>	<b>SOLID tamoyillaridagi "S" nimani anglatadi?</b>
	<i>Yagona javobgarlik tamoyili</i>
	<i>Oddiy amalga oshirish tamoyili</i>
	<i>Strukturaviy dasturlash printsipti</i>
	<i>Statik yozish printsipti</i>

<b>135</b>	<b>Qaysi SOLID tamoyili sinf o'zgarishi uchun faqat bitta sababga ega bo'lishi kerakligini bildiradi?</b>
	<i>Yagona javobgarlik tamoyili</i>
	<i>Ochiq/yopiq tamoyil</i>
	<i>Liskov almashtirish printsipti</i>
	<i>Interfeyslarni ajratish printsipti</i>

<b>136</b>	<b>SOLID tamoyillarida "O" ning asosiy g'oyasi nima?</b>
	<i>Sinflar uzaytirish uchun ochiq, lekin o'zgartirish uchun yopiq bo'lishi kerak.</i>
	<i>Sinflar yagona mas'uliyatga ega bo'lishi kerak.</i>
	<i>Ob'ektlarni ularning kichik tiplari misollari bilan almashtirish imkoniyati bo'lishi kerak.</i>
	<i>Interfeyslar mijozning ehtiyojlariga xos bo'lishi kerak.</i>

<b>137</b>	<b>Qaysi SOLID tamoyili olingan sinflar ularning asosiy sinflari o'rnini bosuvchi bo'lishi kerakligini ta'kidlaydi?</b>
	<i>Liskov almashtirish printsipti</i>
	<i>Ochiq/yopiq tamoyil</i>
	<i>Interfeyslarni ajratish printsipti</i>
	<i>Bog'liqlik inversiyasi printsipti</i>

<b>138</b>	<b>SOLID tamoyillariga ko'ra, sinfda nimani minimallashtirish kerak?</b>
	<i>Aniq amalga oshirishga bog'liqlik</i>
	<i>Usullar soni</i>
	<i>Atributlar soni</i>
	<i>Konstruktorning ortiqcha yuklanishi soni</i>

<b>139</b>	<b>SOLID tamoyillaridagi "L" ning asosiy yo'nalishi nima?</b>
	<i>Subtiplar o'zlarining asosiy turlari o'rnini bosadigan bo'lishi kerak.</i>
	<i>Sinflar uzaytirish uchun ochiq, lekin o'zgartirish uchun yopiq bo'lishi kerak.</i>
	<i>Sinflar yagona mas'uliyatga ega bo'lishi kerak.</i>
	<i>Interfeyslar mijozning ehtiyojlariga xos bo'lishi kerak.</i>

<b>140</b>	<b>Qaysi SOLID tamoyili umumiy maqsadli interfeyslar o'rniga mijozga xos interfeyslarni qo'llab-quvvatlaydi?</b>
	<i>Interfeyslarni ajratish printsipti</i>
	<i>Yagona javobgarlik tamoyili</i>
	<i>Bog'liqlik inversiyasi printsipti</i>
	<i>Liskov almashtirish printsipti</i>

<b>141</b>	<b>SOLID tamoyillaridagi "D" birinchi navbatda nimaga qaratilgan?</b>
	<i>Yuqori darajadagi modullar past darajadagi modullarga bog'liq bo'lmasligi kerak; ikkalasi ham abstraksiyalarga bog'liq bo'lishi kerak.</i>
	<i>Ob'ektlarni ularning kichik tiplari misollari bilan almashtirish imkoniyati bo'lishi kerak.</i>
	<i>Ob'ektlar yagona javobgarlik tamoyiliga amal qilishlari kerak.</i>

	<i>Sinflar yagona mas'uliyatga ega bo'lishi kerak.</i>
<b>142</b>	<b>Yuqori darajali modullar past darajali modullarga bog'liq bo'lmasligi kerak, lekin ikkalasi ham abstraksiyalarga bog'liq bo'lishi kerakligi qaysi SOLID tamoyilida ko'rsatilgan?</b>
	<i>Bog'liqlik inversiyasi printsipti</i>
	<i>Yagona javobgarlik tamoyili</i>
	<i>Ochiq/yopiq tamoyil</i>
	<i>Liskov almashtirish printsipti</i>
<b>143</b>	<b>SOLID tamoyillaridagi "I"ning asosiy maqsadi nima?</b>
	<i>Mijozlarni o'zlari foydalanmayotgan interfeyslarga qaram bo'lishga majburlamaslik kerak.</i>
	<i>Sinflar yagona mas'uliyatga ega bo'lishi kerak.</i>
	<i>Subtiplar o'zlarining asosiy turlari o'rnini bosadigan bo'lishi kerak.</i>
	<i>Yuqori darajadagi modullar past darajadagi modullarga bog'liq bo'lmasligi kerak.</i>
<b>144</b>	<b>Serversiz naqshning asosiy xususiyati nima?</b>
	<i>Ishlab chiquvchilardan server boshqaruvini abstrakt qiladi.</i>
	<i>Bu maxsus server resurslarini talab qiladi.</i>
	<i>Monolit arxitekturaga tayanadi.</i>
	<i>doimiy server monitoringini talab qiladi.</i>
<b>145</b>	<b>Serversiz naqsh server boshqaruvini qanday boshqaradi?</b>
	<i>Uni ishlab chiquvchilardan uzoqlashtiradi.</i>
	<i>Ishlab chiquvchining bevosita aralashuvini talab qiladi.</i>
	<i>Serverni qo'lda tayyorlashni talab qiladi.</i>
	<i>Murakkab server konfiguratsiyasiga tayanadi.</i>
<b>146</b>	<b>Serversiz naqshning muhim afzalligi nimada?</b>
	<i>Operatsion xarajatlarning qisqarishi.</i>
	<i>Serverga texnik xizmat ko'rsatishning ortishi.</i>
	<i>Infratuzilma xarajatlarining yuqoriligi.</i>
	<i>Kompleks server provayderi.</i>
<b>147</b>	<b>Serversiz rejimda server resurslarini kim boshqaradi?</b>
	<i>Bulutli provayderlar.</i>
	<i>Ishlab chiquvchilar.</i>
	<i>Uchinchi tomon sotuvchilari.</i>
	<i>Tizim administratorlari.</i>
<b>148</b>	<b>Serversiz odatda qanday arxitektura turidan foydalanadi?</b>
	<i>Voqealarga asoslangan arxitektura.</i>
	<i>Monolit me'morchilik.</i>
	<i>Mikroservislar arxitekturasi.</i>
	<i>An'anaviy mijoz-server arxitekturasi.</i>
<b>149</b>	<b>Serversiz hisoblashda narxlar odatda qanday tuzilgan?</b>
	<i>Foydalanish uchun to'lov modeli.</i>
	<i>Ruxsat etilgan oylik obuna.</i>
	<i>Litsenziyalash uchun oldindan to'lovlar.</i>
	<i>Xarajatlar tarkibi yo'q.</i>

<b>150</b>	<b>Serversiz shablon kontekstida “serversiz” atamasi nimani anglatadi?</b>
	<i>Ishlab chiquvchilar server boshqaruvidan abstraktlanadi.</i>
	<i>Serverlar jismonan yo'q.</i>
	<i>Serverlar butunlay ishlab chiquvchilar tomonidan boshqariladi.</i>
	<i>Serverlar bepul.</i>

<b>151</b>	<b>Serversiz hisoblash uchun umumiy foydalanish holati nima?</b>
	<i>Voqealarga asoslangan ilovalar.</i>
	<i>Statik veb-saytlar.</i>
	<i>Monolit ilovalar.</i>
	<i>Murakkab meros tizimlar.</i>

<b>152</b>	<b>Serversiz naqshning qaysi jihati uning tez rivojlanishi uchun mashhur bo'lishiga yordam beradi?</b>
	<i>Infratuzilmani boshqarishning qisqarishi.</i>
	<i>Server monitoringining kuchayishi.</i>
	<i>Murakkab server konfiguratsiyalari.</i>
	<i>To'g'ridan-to'g'ri serverni ta'minlash.</i>

<b>153</b>	<b>Mijoz-server modeli ma'lumotlarga kirish nuqtai nazaridan nimani belgilaydi?</b>
	<i>Qurilmalar serverlarda saqlangan ma'lumotlarga qanday kirishadi.</i>
	<i>Serverlar qurilmalarda saqlangan ma'lumotlarga qanday kirishadi.</i>
	<i>Qanday qilib qurilmalar bir-biri bilan bevosita ma'lumot almashadi.</i>
	<i>Mijoz ishtirokisiz serverlar ma'lumotlarni qanday saqlaydi.</i>

<b>154</b>	<b>Mijoz-server arxitekturasida serverning asosiy roli nimadan iborat?</b>
	<i>Mijozlarga turli xizmatlarni taklif qilish.</i>
	<i>Mijozlardan xizmatlar talab qilish.</i>
	<i>Ma'lumotlar ombori vazifasini bajarish.</i>
	<i>Mijozlar bilan muloqotni boshlash.</i>

<b>155</b>	<b>Mijoz-server arxitekturasi qurilmalar o'rtasida muvofiqlikni qanday ta'minlaydi?</b>
	<i>Bir nechta mijozlarga bitta serverdan ma'lumotlarga kirishga ruxsat berish orqali.</i>
	<i>Har bir qurilmada ma'lumotlarni alohida saqlash orqali.</i>
	<i>Serverlarga kirishni cheklash orqali.</i>
	<i>Mijoz va server aloqasini oldini olish orqali.</i>

<b>156</b>	<b>Mijoz-server modelida mijozning asosiy vazifasi nimadan iborat?</b>
	<i>Serverdan xizmatlar talab qilish.</i>
	<i>Serverga xizmat ko'rsatish.</i>
	<i>Lokal tarzda ma'lumotlarni saqlash.</i>
	<i>Server resurslarini boshqarish.</i>

<b>157</b>	<b>Fayl yoki ilovalar to'g'ridan-to'g'ri so'nggi nuqtaga joylashtirilmagan, balki uzoq serverga o'rnatiladigan texnikani qaysi atama tavsiflaydi?</b>
	<i>Mijoz-server modeli.</i>
	<i>Bulutli hisoblash.</i>
	<i>Taqsimlangan hisoblashlar.</i>
	<i>masofaviy hosting.</i>

<b>158</b>	<b>Mijoz-server modelidagi mijozdan so'rovni olgandan so'ng server nima qiladi?</b>
	<i>Hisob ma'lumotlarini tekshiradi va kerakli faylni taqdim etadi.</i>

	<i>So'rovni e'tiborsiz qoldiradi.</i>
	<i>So'rovni mahalliy darajada saqlaydi.</i>
	<i>Mijozga so'rovni boshlaydi.</i>

<b>159</b>	<b>Mijoz-server arxitekturasida mijoz qanday rol o'ynaydi?</b>
	<i>Serverdan xizmatlar talab qilish.</i>
	<i>Serverga turli xizmatlarni taklif qilish.</i>
	<i>Ma'lumotlar ombori vazifasini bajarish.</i>
	<i>Boshqa mijozlar bilan muloqotni boshlash.</i>

<b>160</b>	<b>Serversiz hisoblash patternining asosiy maqsadi nima?</b>
	<i>Server infratuzilmasiga bo'lgan ehtiyojni bartaraf etish uchun.</i>
	<i>Katta ma'lumotlar bazasini gorizontal ravishda kichikroq, boshqariladigan bo'limlarga bo'lish uchun.</i>
	<i>Talabning keskin o'sishini boshqarish uchun tashkilotlarga ommaviy bulut resurslaridan foydalanishga ruxsat berish</i>
	<i>Tizimlar nosozliklar yoki uzilishlar sodir bo'lganda ishlashini ta'minlash.</i>

<b>161</b>	<b>Adaptiv arxitektura patternining maqsadi nima?</b>
	<i>Tizimlar nosozliklar yoki uzilishlar sodir bo'lganda ishlashini ta'minlash.</i>
	<i>Bulutli manbalar va xizmatlarni boshqarish.</i>
	<i>Server infratuzilmasiga bo'lgan ehtiyojni bartaraf etish uchun.</i>
	<i>Talabga qarab dasturga ajratilgan hisoblash resurslari miqdorini sozlash</i>

<b>162</b>	<b>Qaysi dasturiy ta'minot arxitektura patterni modulli va "plug-and-play" yondashuvi orqali dasturiy ta'minot komponentlarini qayta ishlatish imkonini beradi?</b>
	<i>Komponentga asoslangan arxitektura</i>
	<i>Mikroservislar arxitekturasi</i>
	<i>Xizmatga yo'naltirilgan arxitektura (SOA)</i>
	<i>Voqealarga asoslangan arxitektura (EDA)</i>

<b>163</b>	<b>Dasturiy ta'minot arxitekturasida Repository patternining maqsadi nima?</b>
	<i>Ma'lumotlarga kirish operatsiyalari uchun markazlashtirilgan saqlashni ta'minlaydi.</i>
	<i>Mikroservislar arxitekturasida xizmatlarni taqsimlashni boshqaradi.</i>
	<i>Dasturiy ta'minot ilovasida foydalanuvchi interfeysi komponentlarini amalga oshiradi.</i>
	<i>Mijozlar va serverlar o'rtasida uzluksiz aloqani ta'minlaydi.</i>

<b>164</b>	<b>Qaysi arxitektura patterni dasturiy ta'minot komponentlari o'rtasidagi bog'liqlikni minimallashtirishga qaratilgan?</b>
	<i>Bo'shashgan ulanish</i>
	<i>Qattiq ulanish</i>
	<i>Komponentga asoslangan arxitektura</i>
	<i>Mijoz-server arxitekturasi</i>

<b>165</b>	<b>Mijoz-server arxitekturasida mijoz odatda qanday rol o'ynaydi?</b>
	<i>Xizmatlar yoki resurslar uchun so'rovlarni boshlaydi.</i>
	<i>Boshqa mijozlarga mezbonlik qiladi va xizmatlar ko'rsatadi.</i>
	<i>Mijozlar o'rtasidagi tarmoq ulanishini boshqaradi.</i>
	<i>Resurslarga kirish huquqini tasdiqlaydi va ruxsat beradi.</i>

<b>166</b>	<b>Qaysi dasturiy ta'minot arxitekturasi o'zining miqyosi, nosozliklarga chidamliligi va yuqori mavjudligi bilan mashhur?</b>
------------	---

	<i>Bulutli mahalliy arxitektura</i>
	<i>Voqealarga asoslangan arxitektura (EDA)</i>
	<i>Xizmatga yo'naltirilgan arxitektura (SOA)</i>
	<i>Monolitik arxitektura</i>

<b>167</b>	<b>Model-View-Controller (MVC) arxitekturasining asosiy maqsadi nima?</b>
	<i>komponentlarni ajratilishi va komponentlarning qayta ishlatilishi.</i>
	<i>Mijozlar va serverlar o'rtasida samarali aloqa.</i>
	<i>Resurslarni markazlashtirilgan nazorat qilish va boshqarish.</i>
	<i>Yaxshiroq ishlash uchun algoritmlarni optimallashtirish.</i>

<b>168</b>	<b>Qaysi dasturiy ta'minot arxitektura patterni foydalanuvchi interfeyslarini biznes mantiqiy va ma'lumotlarga kirish qatlamlaridan ajratish imkonini beradi?</b>
	<i>Model-View-Controller (MVC)</i>
	<i>Xizmatga yo'naltirilgan arxitektura (SOA)</i>
	<i>Voqealarga asoslangan arxitektura (EDA)</i>
	<i>Qatlamli arxitektura</i>

<b>169</b>	<b>Qaysi dasturiy ta'minot arxitektura patterni ilovani mustaqil ravishda ishlab chiqilishi va sinab ko'rilishi mumkin bo'lgan komponentlarga ajratadi?</b>
	<i>Mikroservislar arxitekturasi</i>
	<i>Monolitik arxitekturasi</i>
	<i>Mijoz-server arxitekturasi</i>
	<i>Peer-to-peer arxitekturasi</i>

<b>170</b>	<b>Dasturiy ta'minot arxitekturasi nima?</b>
	<i>Dasturiy ta'minot tizimining tuzilishi va tashkil etilishi.</i>
	<i>Xatolar va xatolar uchun dasturiy ta'minotni sinovdan o'tkazish jarayoni.</i>
	<i>Dasturiy ta'minot algoritmlarini amalga oshirish.</i>
	<i>Dasturiy ta'minotda foydalanuvchi interfeyslarini loyihalash.</i>

<b>171</b>	<b>Qaysi dasturiy ta'minot arxitektura modeli kichik, tez-tez yangilanishlar orqali dasturiy ta'minotni uzluksiz yetkazib berish va joylashtirish imkonini beradi?</b>
	<i>Uzluksiz integratsiya/uzluksiz joylashtirish (CI/CD)</i>
	<i>Monolitik arxitektura</i>
	<i>Mijoz-server arxitekturasi</i>
	<i>Xizmatga yo'naltirilgan arxitektura (SOA)</i>

<b>172</b>	<b>Docker kabi konteynerlashtirish arxitekturasining asosiy maqsadi nima?</b>
	<i>Dasturiy ta'minot ilovalarini ularning portativlikka bog'liqligi bilan qadoqlash.</i>
	<i>Resurslarni markazlashtirilgan nazorat qilish va boshqarish.</i>
	<i>Komponentlar o'rtasidagi real vaqtda aloqa.</i>
	<i>Yaxshiroq ishlash uchun algoritmlarni optimallashtirish.</i>

<b>173</b>	<b>Tarqalgan tizimda qaysi dasturiy ta'minot arxitektura patterni ma'lumotlarni bir nechta serverlar bo'ylab takrorlash orqali xatolarga chidamlilik va ortiqchalikni ta'minlaydi?</b>
	<i>Replikatsiya arxitekturasi</i>
	<i>Mikroservislar arxitekturasi</i>
	<i>Monolitik arxitektura</i>
	<i>Tengdosh-tengdosh arxitekturasi</i>

174	<b>Qaysi dasturiy ta'minot arxitektura patterni standartlashtirilgan interfeyslar orqali turli tizimlarni uzluksiz integratsiyalashuviga imkon beradi?</b>
	<i>Xizmatga yo'naltirilgan arxitektura (SOA)</i>
	<i>Mikroservislar arxitekturasi</i>
	<i>Komponentga asoslangan arxitektura</i>
	<i>Voqealarga asoslangan arxitektura (EDA)</i>

175	<b>Dasturiy ta'minot arxitekturasida dasturiy ta'minot dizayni patternining maqsadi nima?</b>
	<i>Takroriy dizayn muammolari uchun tasdiqlangan echimlarni taqdim etadi.</i>
	<i>Dasturiy ta'minotning foydalanuvchi interfeysi komponentlarini aniqlaydi.</i>
	<i>Mijozlar va serverlar o'rtasida aloqa protokollarini o'rnatadi.</i>
	<i>Ma'lumotlarni samarali qayta ishlash uchun algoritmlarni amalga oshiradi.</i>

176	<b>Qaysi dasturiy ta'minot arxitektura patterni katta hajmdagi ma'lumotlarni bir nechta serverlar bo'ylab tarqatish orqali ishlov berish uchun mo'ljallangan?</b>
	<i>Big data arxitekturasi</i>
	<i>Mikroservislar arxitekturasi</i>
	<i>Monolitik arxitektura</i>
	<i>Peer-to-peer arxitekturasi</i>

177	<b>Serversiz arxitekturaning asosiy afzalligi nimada?</b>
	<i>Serverni boshqarish va miqyoslash bilan bog'liq muammolarni bartaraf etish.</i>
	<i>Resurslarni markazlashtirilgan nazorat qilish va boshqarish.</i>
	<i>Optimallashtirilgan algoritmlar orqali yaxshilangan ishlash.</i>
	<i>Qisqartirilgan murakkablik va soddalashtirilgan texnik xizmat ko'rsatish.</i>

178	<b>Qaysi dasturiy ta'minot arxitekturasi aniq belgilangan interfeyslarni ta'sir qilish orqali funktsionallikni almashish va qayta foydalanish imkonini beradi?</b>
	<i>Service Oriented arxitekturasi (SOA)</i>
	<i>Monolitik arxitekturasi</i>
	<i>Komponentga asoslangan arxitekturasi</i>
	<i>Cloud native arxitekturasi</i>

179	<b>Qatlamli arxitekturaning asosiy maqsadi nima?</b>
	<i>Xavotirlarni ajratish va modullik.</i>
	<i>Komponentlar o'rtasidagi real vaqtda aloqa.</i>
	<i>Uskuna resurslaridan samarali foydalanish.</i>
	<i>Ma'lumotlarni markazlashtirilgan nazorat qilish va boshqarish.</i>

180	<b>Qaysi dasturiy ta'minot arxitektura patterni voqealar va xabarlarni qayta ishlash va ularga munosabat bildirishga qaratilgan?</b>
	<i>Voqealarga asoslangan arxitektura (EDA)</i>
	<i>Model-View-Controller (MVC)</i>
	<i>Xizmatga yo'naltirilgan arxitektura (SOA)</i>
	<i>Qatlamli arxitektura</i>

181	<b>Mikroservislar arxitekturasining asosiy afzalligi nimada?</b>
	<i>Mustaqil xizmatlar orqali kengaytirilishi va moslashuvchanligi.</i>
	<i>Resurslarni markazlashtirilgan nazorat qilish va boshqarish.</i>
	<i>Optimallashtirilgan algoritmlar orqali yaxshilangan ishlash.</i>
	<i>Qisqartirilgan murakkablik va soddalashtirilgan texnik xizmat ko'rsatish.</i>

<b>182</b>	<b>Mijoz-server arxitekturasining qaysi komponenti xavfsizlik siyosatini amalga oshirish va resurslarga kirishni nazorat qilish uchun javobgardir?</b>
	<i>Server</i>
	<i>Mijoz</i>
	<i>Xavfsizlik devori</i>
	<i>Router</i>

<b>183</b>	<b>Mijoz-server modelida mijoz odatda xizmat yoki resurs so'rash uchun server bilan qanday munosabatda bo'ladi?</b>
	<i>Serverga so'rov xabarini yuboradi</i>
	<i>Server bilan to'g'ridan-to'g'ri aloqa o'rnatadi</i>
	<i>So'rovni tarmoqdagi barcha serverlarga uzatadi</i>
	<i>So'rovni tarmoqdagi boshqa mijozlar orqali uzatadi</i>

<b>184</b>	<b>Mijoz-server arxitekturasining peer-to-peer tarmog'iga nisbatan asosiy afzalligi nimada?</b>
	<i>Resurslarni markazlashtirilgan boshqarish va nazorat qilish</i>
	<i>Kengaytirish qobiliyati va xatolarga chidamlilik</i>
	<i>Maxfiylik va xavfsizlik yaxshilandi</i>
	<i>Bitta nosozlik nuqtasiga bog'liqlik kamayadi</i>

<b>185</b>	<b>Mijoz-server modelidagi serverlarga masofaviy kirish uchun odatda qaysi protokoldan foydalaniladi?</b>
	<i>SSH (Secure Shell)</i>
	<i>SNMP (Simple Network Management Protocol)</i>
	<i>RDP (Remote Desktop Protocol)</i>
	<i>ICMP (Internet Control Message Protocol)</i>

<b>186</b>	<b>Mijoz-server arxitekturasida server odatda qayta ishlash va saqlash nuqtai nazaridan qanday rol o'ynaydi?</b>
	<i>Xizmatlarni taqdim etadi va mijozlar so'rovlarini qayta ishlaydi</i>
	<i>Mijozlar nomidan ma'lumotlarni saqlaydi va boshqaradi</i>
	<i>Mijozlar o'rtasida tarmoq ulanishlarini o'rnatadi</i>
	<i>Autentifikatsiya qiladi va mijozlarga manbalarga kirish huquqini beradi</i>

<b>187</b>	<b>Mijoz-server modelida serverning asosiy javobgarligi nimadan iborat?</b>
	<i>Mijozlarning so'rovlariga javob bering va xizmatlar yoki resurslarni taqdim eting</i>
	<i>Mijozlar o'rtasida tarmoq ulanishlarini o'rnatish</i>
	<i>Resurslarga kirish uchun mijozlarni autentifikatsiya qilish va avtorizatsiya qilish</i>
	<i>Mijozlar nomidan mijoz ma'lumotlarini saqlash va boshqarish</i>

<b>188</b>	<b>Mijoz-server modelida veb-sahifalarni ko'rish uchun qaysi protokol odatda qo'llaniladi?</b>
	<i>HTTP (Hypertext Transfer Protocol)</i>
	<i>DNS (Domain Name System)</i>
	<i>FTP (File Transfer Protocol)</i>
	<i>DHCP (Dynamic Host Configuration Protocol)</i>

<b>189</b>	<b>Mijoz-server arxitekturasi resurslarni boshqarish nuqtai nazaridan qanday afzalliklarga ega?</b>
	<i>Resurslarni markazlashtirilgan nazorat qilish va boshqarish</i>
	<i>Resurslarni markazlashmagan taqsimlash</i>
	<i>Resurslarni mijozlar o'rtasida teng taqsimlash</i>
	<i>Mijoz talablari asosida resurslarni avtomatik taqsimlash</i>

<b>190</b>	<b>Quyidagilardan qaysi biri mijoz-server modelidagi mijozga xos xususiyatdir?</b>
	<i>Serverga so'rovlar yuboradi</i>
	<i>Boshqa mijozlarga xizmat ko'rsatadi</i>
	<i>Server ma'lumotlarini saqlaydi va boshqaradi</i>
	<i>Mijozlar o'rtasidagi tarmoq ulanishini boshqaradi</i>

<b>191</b>	<b>Mijoz-server arxitekturasida qaysi komponent tarmoq trafiginı boshqaradi va so'rovlarnı tegishli serverga yo'naltiradi?</b>
	<i>Load balancer</i>
	<i>Gateway</i>
	<i>Switch</i>
	<i>Modem</i>

<b>192</b>	<b>Mijoz va server o'rtasidagi elektron pochta aloqasi uchun qanday protokol odatda qo'llaniladi?</b>
	<i>POP3 (Post Office Protocol 3)</i>
	<i>IMAP (Internet Message Access Protocol)</i>
	<i>SMTP (Simple Mail Transfer Protocol)</i>
	<i>FTP (File Transfer Protocol)</i>

<b>193</b>	<b>Repository Design Pattern qanday o'zgarishlarga osongina moslashadi?</b>
	<i>Ma'lumotlar manbalarining o'zgarishi.</i>
	<i>Foydalanuvchi interfeysi dizaynidagi o'zgarishlar.</i>
	<i>Uskunaning texnik xususiyatlarining o'zgarishi.</i>
	<i>Tarmoq konfiguratsiyasining o'zgarishi.</i>

<b>194</b>	<b>Repository Design Pattern birinchi navbatda quyidagilarga kirishni tashkil qiladi va soddalashtiradi:</b>
	<i>Axborot.</i>
	<i>Foydalanuvchi interfeyslari.</i>
	<i>Tarmoq resurslari.</i>
	<i>Tizimning texnik vositalari.</i>

<b>195</b>	<b>Repository Design Pattern kontekstida ma'lumotlarnı saqlash tafsilotlari:</b>
	<i>Abstraktlashtirilgan.</i>
	<i>Ochiq.</i>
	<i>Murakkab.</i>
	<i>E'tiborga olinmagan.</i>

<b>196</b>	<b>Repository Design Pattern foydalanish quyidagini oshiradi:</b>
	<i>Dasturiy ta'minot tizimlarining samaradorligi va moslashuvchanligi.</i>
	<i>Dasturiy ta'minot tizimlarining murakkabligi.</i>
	<i>Dasturiy ta'minotni ishlab chiqish xarajatlari.</i>
	<i>Dasturiy tizimlarning quvvat sarfi.</i>

<b>197</b>	<b>Repository dizayn namunasi quyidagilar hisoblanadi:</b>
	<i>Dasturiy ta'minotni loyihalash namunasi.</i>
	<i>Uskunani loyihalash namunasi.</i>
	<i>Tarmoqni loyihalash namunasi.</i>
	<i>Xavfsizlik dizayn namunasi.</i>

<b>198</b>	<b>Quyidagilardan qaysi biri repozitoriy dizayn namunasining foydasi EMAS?</b>
------------	--



	<i>Axborotga kirishni osonlashtiradi.</i>
	<i>U past darajadagi apparat boshqaruvini birlashtiradi.</i>
	<i>Ma'lumotlarga kirishni tashkil qiladi.</i>
	<i>Dasturiy ta'minotning moslashuvchanligini oshiradi.</i>

<b>199</b>	<b>Repository Design Pattern quyidagilarga yordam beradi:</b>
	<i>Biznes mantiqini ma'lumotlarga kirishdan ajratish.</i>
	<i>Tarmoq protokollarini yaratish.</i>
	<i>Foydalanuvchining grafik interfeyslarini boshqarish.</i>
	<i>Operatsion tizim funksiyalarini boshqarish.</i>

<b>200</b>	<b>Dasturiy ta'minot tizimlarining samaradorligi quyidagilardan foydalangan holda oshiriladi:</b>
	<i>Repozitariy dizayn namunasi.</i>
	<i>Tasodifiy kodlash amaliyotlari.</i>
	<i>Ma'lumotlarni qa'lda boshqarish.</i>
	<i>Apparat darajasida dasturlash.</i>

<b>201</b>	<b>Repository Design Pattern asosan nimani boshqaradi?</b>
	<i>Ma'lumotlar.</i>
	<i>Tarmoq trafigi.</i>
	<i>Foydalanuvchi seanslari.</i>
	<i>Texnikaviy uzilishlar.</i>

<b>202</b>	<b>Repository Design Patternning asosiy xususiyatlaridan biri quyidagilardan iborat:</b>
	<i>Ma'lumotlarga kirishni standartlashtirish.</i>
	<i>Foydalanuvchi interfeyslarini moslashtirish.</i>
	<i>Grafik tasvirni yaxshilash.</i>
	<i>Tarmoqning o'tkazuvchanligini oshirish.</i>

<b>203</b>	<b>Quyidagi qatlamlardan qaysi biri to'g'ridan-to'g'ri Repository Design Pattern bilan o'zaro aloqada EMAS?</b>
	<i>Fizikaviy apparat qatlami.</i>
	<i>Amaliy biznes mantiqiy qatlami.</i>
	<i>Ma'lumotlarni saqlash qatlami.</i>
	<i>Ma'lumotlar bazasi qatlami.</i>

<b>204</b>	<b>Repository Design Pattern katta va murakkab tizimlarga qanday foyda keltiradi?</b>
	<i>Ma'lumotlarga kirishni boshqarishning tashkiliy usulini taklif qilish orqali.</i>
	<i>Ma'lumotlar bilan ishlashning murakkabligini oshirish orqali.</i>
	<i>Ma'lumotlarga kirishni cheklash orqali.</i>
	<i>Xotirani taqsimlashni boshqarish orqali.</i>

<b>205</b>	<b>Repository Design Pattern sinov muhitida qanday afzalliklarni beradi?</b>
	<i>Ma'lumotlarga kirishni simulyatsiya qilish uchun soxta omborlarni yaratishga imkon beradi.</i>
	<i>Test holatlarining murakkabligini oshiradi.</i>
	<i>Sinovni faqat ishlab chiqarish ma'lumotlari bilan cheklaydi.</i>
	<i>Sinov muhitlariga bo'lgan ehtiyojni yo'q qiladi.</i>

<b>206</b>	<b>Repository Design Pattern ma'lumotlarni ko'chirishda qanday yordam beradi?</b>
	<i>Yangi ma'lumotlar manbasiga yumshoqroq o'tishga imkon berish orqali.</i>
	<i>Migratsiya jarayonini murakkablashtirish orqali.</i>
	<i>Ma'lumotlar migratsiyasini foydalanuvchi interfeysiga integratsiyalashgan holda.</i>

	<i>Migratsiya vaqtida ma'lumotlardan foydalanishni cheklash orqali.</i>
<b>207</b>	<b>Repository Design Pattern-dan foydalanganda veb-ilovalarda nima abstraktlanadi?</b>
	<i>Ma'lumotlar bazasi bilan o'zaro aloqasi.</i>
	<i>Foydalanuvchi interfeysi dizayni.</i>
	<i>tarmoq aloqasi.</i>
	<i>apparat boshqaruvi.</i>
<b>208</b>	<b>Repository Design Pattern API va mikroservislarga qanday foyda keltiradi?</b>
	<i>Ma'lumotlarning uzluksiz va tartibli o'zaro ta'sirini ta'minlash orqali.</i>
	<i>Foydalanuvchi seanslarini boshqarish orqali.</i>
	<i>Grafik ish faoliyatini optimallashtirish orqali.</i>
	<i>Texnikani bevosita boshqarish orqali.</i>
<b>209</b>	<b>Ma'lumotlarga kirishning murakkab va tartibsiz mantig'i tufayli Repository Design Pattern qaysi turdagi tizimlarda ayniqsa foydali?</b>
	<i>Katta va murakkab tizimlar.</i>
	<i>Kichik o'rnatilgan tizimlar.</i>
	<i>Oddiy ish stoli ilovalari.</i>
	<i>Bir foydalanuvchiga mo'ljallangan mobil ilovalar.</i>
<b>210</b>	<b>Haqiqiy ma'lumotlar manbasini o'zgartirmasdan ma'lumotlarga kirishni taqlid qilish uchun sinov muhitida nimadan foydalanish mumkin?</b>
	<i>Soxta omborlar.</i>
	<i>Ishlab chiqarish ma'lumotlar bazalari.</i>
	<i>Real vaqt rejimidagi ma'lumotlar manbalari.</i>
	<i>To'g'ridan-to'g'ri apparat vositalariga kirish.</i>
<b>211</b>	<b>Repository Design Pattern ma'lumotlarni ko'chirishda qanday yordam beradi?</b>
	<i>Yangi ma'lumotlar manbai uchun omborni amalga oshirishni almashtirish orqali.</i>
	<i>Ilovani to'liq qayta yozishni talab qilish orqali.</i>
	<i>Yangi ma'lumotlar manbasini qattiq kodlash orqali.</i>
	<i>Eski ma'lumotlar manbasini e'tiborsiz qoldirib.</i>
<b>212</b>	<b>Repository Design Pattern veb-ilovalarda nimani abstrakt qiladi?</b>
	<i>Ma'lumotlar bazasining o'zaro ta'siri.</i>
	<i>Foydalanuvchi kiritish usullari.</i>
	<i>Uskunaga bog'liqliklar.</i>
	<i>Tarmoq protokollari.</i>
<b>213</b>	<b>Quyidagilardan qaysi biri API-larda repozitoriy dizayn namunasidan foydalanishning afzalligi hisoblanadi?</b>
	<i>Bir nechta xizmatlarning ma'lumotlar bilan uzluksiz ishlashiga imkon beradi.</i>
	<i>Ma'lumotlarni tekshirishga bo'lgan ehtiyojni kamaytiradi.</i>
	<i>Ma'lumotlarga kirish usullarini murakkablashtiradi.</i>
	<i>Foydalanuvchi interfeyslarini bevosita boshqaradi.</i>
<b>214</b>	<b>Repository Design Pattern qaysi turdagi tizimlar uchun ma'lumotlarga kirishni tashkiliy boshqarishni ta'minlaydi?</b>
	<i>Katta va murakkab tizimlar.</i>
	<i>Kichik va oddiy ilovalar.</i>
	<i>Ish stolining mustaqil vositalari.</i>

	<i>Bir funksiyali skriptlar.</i>
--	----------------------------------

<b>215</b>	<b>Sinov muhitida soxta omborlardan foydalanishning asosiy afzalligi nimada?</b>
	<i>Ular ma'lumotlarga kirishni izolyatsiya qilish testlarini o'tkazishga imkon beradi.</i>
	<i>Ular haqiqiy ma'lumotlar manbalarini talab qiladi.</i>
	<i>Ular test jarayonini murakkablashtiradi.</i>
	<i>Ular birlik sinovini oldini oladi.</i>

<b>216</b>	<b>Yangi ma'lumotlar bazasiga o'tayotganda, "Repository Design Pattern" nimadan qochishga yordam beradi?</b>
	<i>Ilovaning qolgan qismini o'zgartirish.</i>
	<i>Foydalanuvchi interfeysini o'zgartirish.</i>
	<i>Biznes mantiqini qayta yozish.</i>
	<i>Tarmoq konfiguratsiyasini o'zgartirish.</i>

<b>217</b>	<b>Nega omborlar ma'lumotlar bazasi bilan o'zaro aloqada bo'lish uchun veb-ilovalarda foydali?</b>
	<i>Ular ma'lumotlar bazasi o'zaro ta'sirini abstraktlashtiradi va soddalashtiradi.</i>
	<i>Ular bevosita DOMni manipulyatsiya qiladilar.</i>
	<i>Ular grafik tasvirni kuchaytiradi.</i>
	<i>Ular server konfiguratsiyasini boshqaradi.</i>

<b>218</b>	<b>Ma'lumotlar bilan ishlashda real vaqt arxitekturasi nimalarga ustunlik beradi?</b>
	<i>Past kechikish va yuqori o'tkazuvchanlik.</i>
	<i>Yuqori kechikish va past o'tkazuvchanlik.</i>
	<i>Ma'lumotlarni tasodifiy qayta ishlash.</i>
	<i>Vahti-vaqti bilan ma'lumotlarni zahiralash.</i>