Kao dioniku u ocjeni kakvoce programskog produkta kupcu je najvaznije: a)Lakoca ponovne uporabe dijelova, b) lakoca odrzavanja, c)sve navedeno, d)lakoca oblikovanja **e)Nijedno od navedenog**

U inženjerstvu zahtjeva programske potpore za zahtjeve vrijede tvrdnje: **to su opisi usluga sustava**, **to su opisi ograničenja sklopovlja, to su ograničenja programske potpore, mogu se izražavati prirodnim jezikom**

U iteracije spiralnog modela procesa inženjerstva zahtjeva ne pripadaju: **izrada prototipa**, **studija izvedivosti**

Ako svi zahtjevi obuhvaćeni u dokumentu inženjerstva zahtjeva imaju samo jednu moguću interpretaciju tada možemo tvrditi da je dokument specifikacije zahtjeva: **jednoznačan**

U koje skupine UML dijagrama spadaju dijagrami obrazaca uporabe i sekvencijski dijagrami?: **Dijagram obrazaca uporabe je statični tim dijagrama, sekvencijski dijagram je dinamični tip dijagrama.**

Stvarima označavanja odgovara kao UML opis: **ozna elemenata modela**

Za sinkrone poruke u sustavu vrijede slijedeće tvrdnje: **pošiljatelj čeka na odgovor, na sekvencijskom dijagramu odgovor se ne prikazuje(podrazumijeva se)**

Gtanica sustava (boundary) u UML dijagramu obrazaca uporabe (use case diagram) predstavlja: **granicu između aktora i sustava**

Vrsta poveznice u dijagramu obrazaca uporabe s prikazom (Ravna horizontalna crta) naziva se: **asocijacija**

Što je potrebno uključiti u dokument oblikovanja arhitekture programske potpore? **Prioritete koji su vodili proces oblikovanja, referenciju prema dokumentu zahtjeva, opis s najviše razine promatranja**

Odredite istinite tvrdnje za svojstva modela procesa programskog inženjersva: **najmanje novog koda generira se u modelu zasnovanom na komponentama**

Koji je model procesa programskog inženjersva najpogodniji za male i srednje interaktivne sustave? **Evolucijski**

Što vijedi za sipralni model razvoja programske potpore? **Barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja**

Koje od navedenih su glavne značajke unificiranog procesa: **obrasci uporabe(engl.use cases), fokus na arhitekturu sustava, iterativni i inkramentalni pristup**

Dopunite sljedeću tvrdnju: „Obrasci uporabe specificiraju **funkcije**, a arhitektura **formu**.

Koje su od navedenih tvrdnji točne? **Polimorfizam označava postojanje više metoda istog naziva, tipa te broja i vrste parametara u podrazredima koji implementiraju istu apstraktnu operaciju superrazreda. Overriding predstavlja inačicu polimorfizma, Svi podrazredi implicitno nasljeđuju sve značajke definirane u superrazredu.**

Osnovu opisa funkcionalnih zahtjeva u dokumentacij projekta čine: **UML sekvencijski dijagrami, dijagrami obrazaca uporabe**

Za sustav Subversiona vrijede tvrdnje:**Izbrisane datoteke je moguće povratiti, broj revizije globalno određuje stanje repozitorija u određenom trenutku**

Koji model procesa programskog inženjerstva je najpogodniji za velike inženjerskve projekte gdje se sustav razvija na nekoliko odvojenih mjesta?: **vodopadni**

Koji od ponuđenih odgovora NE PREDSTAVLJA jedan od 21 principa oblikovanja? **Oblikuj za budućnost**

Koji principi oblikovanja su direktno povezani s principom povećanja ponovne uporabivosti (reusability)? **Povećanje kohezije, smanjenje međuovisnosti, viša razina apstrakcije**

Ako dva objekta imaju identične podatke (atribute) zapravo se radi o jednom te istom objektu koji je dvaput referenciran: **ne**

Koje od navedenih tvrdnji vrijede za odnos agregacije u UML dijagramu : **agragat predstavlja IS-PART-OF odnos, agregacija je slaboja od kompozicije**

Koja se od ponuđenih tvrdnji ne odnosi na pojam razred u objektnoj paradigmi? **Program u radu može se sagreldati kao skup razreda u međusobnoj komunikaciji**

Odnos podrazreda prema svojem superrazredu (engl.superclass) nazivamo: **specijalizacija**

U procesu oblikovanja programske potvore, arhitektura protrama rezultat je **high**- level dizajna, a implementacija **low** level dizajna

Proces inženjerstva zahtjeva ne uključuje: **validaciju programske potpore**

Stvarima označavanja odgovara kao UML opis: **oznake elemenata modela**

Za sekvencijske dijagrame vrijede sljedeće tvrdnje: **prikazuju objekte kao životne crte, dobri su a prikaz komunikacije među objektima, prikazuju interakciju u sustavu**

Kod oblikovanja kojih zahtjeva se koriste UML obrasi uporabe? **Funkcionalnih zahtjeva**

Vrsta poveznice u dijagramu obrazaca uporabe s prikazom (horizonalno položena strelica na desno): **generalizacija**

Princip višestrukog korištenja u oblikovanju programske potpore koristi se u: **ponovnom korištenju konzistencije(programski jezici), ponovnom korištenju fragmenata rješenja (knjižice), ponovnom korištenju arhitekture sustava(arhitekturni obrsasci), ponovnom korištenju arhitekture, ponovnom korištenju cjelokupne arhitekture sustava**

Koji je model procesa programskog inženjerstva najpogoniji za male i srednje interaktivne sustave?: **evolucijski**

Što vrijedi za spiralni model razvoja programske potpore?**: barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja**

Unificirani model procesa (engl.Unified Process Model) razlikuje se od generičkih procesa programskog inženjerstva jer: **povezuje temeljne aktivnosti s modelima, amodele s dijagramima, definira ključne točke (engl.milestones) u procesu programskog inženjerstva, definira faze životnog ciklusa programske potpore**

Prednosti uporabe vodopadnog modela nad „ad hoc“ modelom su: **definirani prijelazi između strogo definiranih faza, dobro definirana slijedna organizacija procesa.**

Što znači princip „oblikuj po ugovoru“? **sve metode imaju ugovor s pozivateljem**

Arhitektura programske potpore rezultat je koje aktivnosti procesa programskog inženjerstva? **Razvoja i oblikovanja**

Koji od ponuđenih odgovora ne predstavlja jedan od 12 principa oblikovanja? **Oblikuj za budućnost**

Za Liskovin princip zamjene vrijedi: **ako je objekt proširenje drugog objekta tada ga je moguće upotrebljavati na svim mjestima tog objekta, odnosi se na interoperabilnost objekta u hijerarhiji nasljeđivanja**

Na glavnoj metodii programa smijemo koristiti nadjačavanje metoda:**netočno**