**Daudzizvēļu Jautājumi:**

1. Kas ir programmēšanas paradigma?
   * A) Rīku kopums koda atkļūdošanai.
   * B) Noteikts veids, kā organizēt un rakstīt programmas.
   * C) Veids, kā ātrāk kompilēt programmas.
   * D) Datu bāzes pārvaldības sistēma.
2. Kura no šīm NAV objektorientētās programmēšanas pīlārs?
   * A) Enkapsulācija
   * B) Mantošana
   * C) Rekursija
   * D) Polimorfisms
3. Kā parasti tiek apstrādāti dati procedūrprogrammēšanā?
   * A) Dati un uzvedība tiek grupēti kopā objektos.
   * B) Dati tiek glabāti atsevišķi no uzvedības tīrās funkcijās.
   * C) Dati tiek manipulēti tieši secīgu instrukciju komplektā.
   * D) Dati tiek glabāti datubāzēs un tiem piekļūst caur SQL vaicājumiem.
4. Kas ir abstrakcijas piemērs reālās pasaules programmēšanā?
   * A) Tieša uzdevuma statusa maiņa uzdevumu sarakstā.
   * B) E-pasta nosūtīšana, nezinot, kā tas tiek apstrādāts.
   * C) Jaunas programmēšanas valodas izveidošana.
   * D) Visas programmas rakstīšana assemblera valodā.
5. Kas ļauj klasei mantot citas klases funkcionalitāti OOP?
   * A) Polimorfisms
   * B) Enkapsulācija
   * C) Mantošana
   * D) Abstrakcija

**Patiesi/Nepariesi Jautājumi:**

1. Objektorientētajā programmēšanā enkapsulācija palīdz apvienot datus un uzvedību, kā arī ierobežo piekļuvi noteiktām koda daļām.
   * Patiesi / Nepatiesi
2. Funkcionālajā programmēšanā dati vienmēr tiek tieši mainīti, veicot operācijas.
   * Patiesi / Nepatiesi
3. Polimorfisms ļauj dažādu klašu objektus apstrādāt kā kopējās virsklases objektus.
   * Patiesi / Nepatiesi
4. OOP abstrakcija slēpj ieviešanas detaļas un atklāj tikai lietotājam nepieciešamo funkcionalitāti.
   * Patiesi / Nepatiesi
5. Klase "uzdevums" un klase "termiņuzdevums" ir piemēri tam, kā mantošanu var izmantot, lai izveidotu specifiskākas klases no vispārīgām.
   * Patiesi / Nepatiesi

**Īsās Atbildes Jautājumi:**

1. **Paskaidrojiet saviem vārdiem, kas ir enkapsulācija un kāpēc tā ir svarīga OOP.**
2. **Aprakstiet, kā funkcionālā programmēšana atšķiras no objektorientētās programmēšanas attiecībā uz datu un uzvedību apstrādi.**
3. **Kas ir polimorfisms un kā tas dod labumu objektorientētajai programmēšanai?**
4. **Dodiet piemēru, kā mantošanu varētu izmantot reālās pasaules pielietojumā ārpus uzdevumu saraksta lietotnes.**
5. **Kāpēc ir svarīgi izprast četrus OOP pīlārus, lai apgūtu jebkuru objektorientētu programmēšanas valodu?**