

Programiranje za web – 2. razred – pitanja i odgovori za ponavljanje (teorijski dio)

1. Što je program?
Program je skup instrukcija ili naredbi dizajniranih da se izvršavaju na računalu ili drugom uređaju. Ti programi obavljaju specifične funkcije i zadatke za korisnike ili druge programe. Programi se mogu sastojati od kratkih naredbi za jednostavne zadatke i složene zadatke. Programski jezici su npr. Python, JavaScript, PHP, C++.
2. Što je algoritam?
Algoritam je skup jasnih i precizno definiranih uputa (koraka) ili pravila za rješavanje određenog problema ili izvođenje zadataka.
3. Što je sintaksa?
Sintaksa u programiranju odnosi se na skup pravila koja određuju kako se kod mora napisati kako bi bio valjan i razumljiv za kompajler ili interpreter određenog programskog jezika. To uključuje specifikacije o pravilnoj uporabi simbola, ključnih riječi, strukturi i formatiranju koda. Baš kao što sintaksa u jeziku određuje pravilnu strukturu rečenica, sintaksa u programiranju definira kako napisati ispravne naredbe koje računalo može izvršiti.
4. Što je varijabla?
Varijabla u programiranju djeluje kao spremnik ili prostor u memoriji gdje programi mogu spremati, ažurirati i preuzimati vrijednosti tijekom izvođenja. Svaka varijabla je označena jedinstvenim imenom i povezana je s određenim tipom podataka, koji određuje vrstu vrijednosti.
5. Što je konstanta?
Konstanta u programiranju je vrsta varijable čija vrijednost se ne može promijeniti nakon što je inicijalno postavljena. Konstante se koriste kada želite osigurati da određene vrijednosti ostanu nepromijenjene kroz cijeli životni vijek programa, pružajući time veću sigurnost i predvidljivost u ponašanju koda.
6. Što je dijagram tijeka/toka?
Dijagram tijeka ili dijagram toka je vizualni prikaz koraka u algoritmu tj. programu. Prikazuje slijed koraka, moguća grananja i ponavljanja (petlju).
7. Što je pseudokod?
Pseudokod je način pisanja algoritama koristeći strukturirani jezik koji nalikuje programskom jeziku, ali je jednostavniji i razumljiviji. Pseudokod nije namijenjen izvršavanju na računalima, već služi kao alat za planiranje i komunikaciju kako bi se olakšalo razumijevanje algoritma prije njegove implementacije u stvarnom programskom jeziku.

8. Kako se klasificiraju računalni programi?

Računalni programi se mogu klasificirati na različite načine, ovisno o nekoliko faktora koji uključuju njihovu namjenu, vrstu izvođenja, pristup izvornom kodu, i vrstu korisničkog sučelja

Npr. prema funkciji ili namjeni:

- sustavski softver - uključuje operativne sustave i alate koji pomažu u upravljanju hardverom i osnovnim sustavnim funkcijama. Primjeri uključuju Windows, macOS, Linux.
- aplikacijski softver - specijalizirani programi namijenjeni za izvođenje specifičnih zadataka korisnika, kao što su obrada teksta, tablični kalkulatori, multimedija, i slično, što uključuje npr. Microsoft Office, Photoshop, AutoCAD.
- programski alati koji omogućuju programerima kreirati, testirati i održavati softver. Uključuje kompilatore, linkere, debuggere, i IDE (integriano razvojno okruženje) poput Visual Studio ili Eclipse.

Prema platformi:

- desktop programi - dizajnirani za upotrebu na desktop ili prijenosnim računalima.
- mobilni programi - dizajnirani za mobilne uređaje - pametni telefoni i tableti.
- web aplikacije - programi koji se izvode unutar web preglednika i dostupni su na bilo kojoj platformi s pristupom internetu.

9. Objasni klijentski jezik.

Klijentski jezik (u kontekstu web razvoja) odnosi se na programski jezik koji se koristi za pisanje softvera koji se izvršava na klijentskoj strani, odnosno u web pregledniku korisnika, umjesto na poslužitelju. Klijentski jezici su ključni za razvoj interaktivnih i dinamičkih web stranica (JavaScript, HTML, CSS).

10. Opiši osnove jezika za skriptiranje serverske strane.

Jezici za skriptiranje serverske strane omogućavaju razvoj dinamičkih web stranica i aplikacija koje mogu komunicirati s bazama podataka, upravljati korisničkim sesijama, izvoditi složene operacije obrade podataka, itd. Ti jezici rade na web poslužitelju, a rezultati njihovog izvođenja šalju se klijentu obično u obliku HTML-a, ali i JSON-a, XML-a ili nekog drugog formata, ovisno o potrebi aplikacije.

Osnovni principi:

- obrada zahtjeva i odgovora: serverski skriptni jezici rade tako što obrade zahtjev koji je poslan s klijentske strane (npr. kada korisnik klikne na link ili pošalje formular) i zatim šalju odgovor nazad klijentu.

- interakcija s bazom podataka: većina web aplikacija zahtijeva spremanje i dohvat podataka, što se obavlja kroz komunikaciju s bazom podataka. Jezici serverskog skriptiranja pružaju funkcije i metode za upravljanje bazama podataka.
- upravljanje stanjem: HTTP je po svojoj prirodi stateless protokol, što znači da ne zadržava informacije između zahtjeva. Serverski jezici omogućuju upravljanje stanjem pomoću sesija i kolačića kako bi se pratilo stanje korisnika kroz više zahtjeva.
Popularni jezici serverskog skriptiranja: PHP, Python (Django, Flask), Node.js (JavaScript).

11. Što su jezični prevoditelji, koja je njihova uloga i što rade?

Jezični prevoditelji su softverski alati koji pretvaraju izvorni kod napisan u jednom programskom jeziku u drugi oblik pogodan za izvršavanje na računalu. Ovaj proces omogućuje da programi napisani u programskim jezicima koji su razumljivi ljudima postanu razumljivi i izvršivi na računalu. Pristupi razvoju programa. Tu spadaju kompajleri i interpreteri.

12. Nabroji faze razvoja programa.

Planiranje, analiza zahtjeva, dizajn i arhitektura, implementacija (kodiranje), testiranje, uvođenje u rad, održavanje i nadogradnja.

13. Nabroji razlozi uvođenja standarda imenovanja i pisanja čitkog koda.

Poboljšana čitljivost i razumljivost, održavanje koda, smanjenje rizika od pogrešaka, ubrzanje procesa razvoja, povećanje suradnje unutar tima, profesionalizam i kvaliteta, lakše automatizirano testiranje.

14. Općenita pravila imenovanja objekata.

Pravila imenovanja objekata u programiranju su ključna za održavanje čitljivosti i organiziranosti koda. Ključna pravila: jasnoća i opisnost, smisleni nazivi, izbjegavati skraćénice, dosljednost, izbjegavati nazive/imena koja mogu izazvati zabunu, ne koristiti ključne riječi (naredbe) programskog jezika za nazive, ograničiti duljinu naziva/imena.

15. Razlika između varijable i konstante.

Varijable i konstante su temeljni koncepti u programiranju, koji omogućavaju pohranjivanje podataka unutar programa. Varijable mogu mijenjati svoje vrijednosti tijekom izvođenja programa, dok su konstante fiksirane i njihove vrijednosti se ne mogu mijenjati nakon inicijalizacije.

16. Nabroji tipove podataka.

Najčešći tipovi podataka u programiranju su: brojevni, znakovni, tekstualni, logički, pokazivači, složeni (strukture, unije, klase).

17. Nabroji vrste operatora.

Aritmetički operatori, operatori usporedbe, logički operatori, operatori dodjele.

18. Koje su osnovne programske strukture?

Sekvenca/slijed, grananje i iteracija/petlja.

19. Upravljanje iznimkama.

Upravljanje iznimkama je ključan dio razvoja robusnog softvera. Iznimke su problemi koji se javljaju tijekom izvođenja programa, a koji mogu prekinuti normalan tok programa. Pogreške kao što su pokušaj dijeljenja s nulom, pristup elementu izvan granica polja, ili pokušaj otvaranja nepostojeće datoteke su primjeri situacija koje mogu generirati iznimke. Upravljanje iznimkama omogućava programerima da na organiziran način obrade ove greške, umjesto da dozvole programu da se sruši.

20. Što je funkcija?

Funkcija u programiranju je blok koda koji je dizajniran da izvrši specifičan zadatak. Funkcije su temeljni alat za modularno programiranje jer omogućavaju ponovno korištenje koda, bolju organizaciju i lakše upravljanje složenim programima. Funkcije se obično definiraju jednom i mogu se pozivati bilo gdje unutar programa, često s različitim ulaznim parametrima, kako bi se izvršili definirani zadaci.

21. Jednostavne strukture podataka.

U programiranju, jednostavne (ili primitivne) strukture podataka su osnovni tipovi podataka koji čine temelje na kojima se grade složenije strukture. Ovi tipovi podataka su obično ugrađeni u programski jezik i imaju operacije definirane direktno od strane samog jezika. Evo nekoliko najčešćih jednostavnih struktura podataka: cijeli brojevi (integer), brojevi s pomičnim zarezom, boolean (true, false), znak, niz znakova (string), null i undefined.

22. Upotreba biblioteka u izradi programa.

Upotreba biblioteka u izradi programa je ključna praksa u modernom softverskom razvoju. Biblioteke su kolekcije prethodno napisanog koda koje programeri mogu koristiti za izvođenje uobičajenih ili složenih zadataka bez potrebe za ponovnim pisanjem koda od nule. Ove biblioteke mogu uključivati sve od jednostavnih funkcija do složenih sustava s mnoštvom funkcionalnosti.

Kristijan Fučko, nastavnik informatike/računalstva