

Tantusz Könyvek – Programozás MI-vel

Rövid összefoglaló

A generatív mesterséges intelligencián alapuló programozási eszközök képesek növelni a termelékenységedet programozóként, emlékeztetni a szintaxisra, és még a tesztelésben, a hibakeresésben, a refaktorálásban és a dokumentálásban is tudnak segíteni, az viszont csak rajtad múlik, hogy tudod-e helyesen használni őket. Itt tíz prompttervezési tippet olvashatsz, amelyekkel elérheted, hogy az MI ahelyett, hogy használhatatlan és strukturálatlan „spagettikódot” dobna vissza, elegáns, működő kódot készítsen.

Az MI-alapú programozási eszközök egyedi kihívásokkal és veszélyekkel járnak a szoftverfejlesztő csapatok számára, ezért fúsd át ezt a néhány egyszerű szabályt, hogy biztos lehess benne, a generatív MI nem teszi tönkre a projektedet. Végül pedig nézd meg, mit válaszolt a ChatGPT arra a kérésre, hogy sorolja fel a legfontosabb olyan dolgokat, amelyeket az emberi programozók csinálnak, és az MI soha nem tudja helyettesíteni őket bennük.

Promptolási tippek az MI-alapú programozáshoz

Prompttervezésnek nevezzük azt a folyamatot, amely során kitalálsz, hogyan adhatod a megfelelő promptokat a generatív MI-eszközeidnek ahhoz, hogy azok kiváló minőségű tartalmat generáljanak. A valóságban azonban a promptok megtervezése inkább kreatív művészet, mint hivatalos mérnöki tudományterület. Az emberek próbálgatással felfedeztek olyan technikákat és kifejezéseket, amelyek megbízhatóan növelik a generált kód minőségét. Íme a tíz legjobb tippem ahhoz, hogy hatékony promptokat küldj a generatív MI-chatbotoknak:

- » Fogalmazz aktívan. Ahelyett, hogy azt írnád: „Kíváncsi vagyok, hogy...”, írd azt, hogy „A kódban lennie kell...” vagy „Azt akarom, hogy...”
- » Kerüld a tagadás használatát. Ahelyett, hogy azt írnád: „Ne használj ciklusokat”, írd azt, hogy „Használj rekurziót ciklusok helyett.”
- » Adj meg egy szerepet. A promptod elején írd le a szerepet, amelyet az MI-nek el kell játszania. Például: „Tapasztalt és magasan képzett programozó vagy.”
- » Hangsúlyozd a kód minőségét. Használj a promptjaidban olyan kifejezéseket, mint például „tisztá kód”, „olvasható” és „jól dokumentált”.
- » Említs meg konkrét könyvtárakat, keretrendszereket és API-kat. Ha elvárod, hogy a generált kód bizonyos eszközöket használjon, említsd meg őket (adott esetben a verzióikkal együtt) a promptodban.
- » Először írd le az általános problémát. Így az MI-eszközök alapvető ismereteket szereznek arról, hogy milyen eredményt szeretnél kapni.
- » Írd le a bemeneti és a kimeneti adatformátumokat. Még ha nyilvánvaló is, hogy a `sumNumbers()` (számok összeadása) nevű függvénynek számokat kell fogadnia argumentumként, és egy számot kell visszaadnia, ha megadod ezt az információt a promptban, azzal korlátozod a döntések számát, amelyeket az MI-nek meg kell hoznia, és javítod a válaszában minőségét.
- » Bontsd kisebb részekre a bonyolult dolgokat. Mindig koncentrálj a probléma egy-egy kis részének megoldására.
- » Fogalmazz világosan és tömören. Kerüld a kétértelműséget és az olyan homályos kifejezéseket, mint „jó” vagy „hatékony”.

- » Dolgozz iteratív módon. Ne elégedj meg az első generált eredménnyel. Ismételd meg és finomítsd a promptodat, és adj visszajelzést az MI-eszköznek, hogy a kívánt eredmény felé tereld. Néha akár egyetlen szó megváltoztatása is drámaian befolyásolhatja a kimenetet.

Az MI által generált kód lehetséges veszélyei

Bár az MI-chatbotok és az MI-alapú programozópárok, például a ChatGPT és a Copilot segítségével hatékonyabb programozóvá válhatsz, de az MI által generált kódnak vannak veszélyei is. Lássuk a legrosszabb lehetséges eredményeit annak, ha egy generatív MI-eszkőzzel írsz kódot:

- » Az MI-vel helyettesíted azt, hogy jobb programozóvá válj. Nem számít, milyen eszközt használsz, programozóként még mindig a te felelősséged, hogy folyamatosan fejlődj és tanulj.
- » Megfelelő kódellenőrzés nélkül helyezel üzembe MI által generált kódot. Bár a generált kód látszólag helyesen működik és nincsenek benne nyilvánvaló hibák, mégis elengedhetetlen, hogy emberek ellenőrizzék a generált kódot, mielőtt az a végfelhasználókhoz kerülne.
- » Elavult függvényeket, könyvtárakat vagy API-kat használsz. Valószínű, hogy a kód, amelyen az MI-asszisztenseket betanították, elavult és esetleg nem biztonságos függőségekre is tartalmaz hivatkozásokat. Manuálisan kell ellenőrizned, hogy a generált kódban mindegyik biztonságos, és helyes-e használni.
- » Zavaros vagy terjengős dokumentáció. A generatív MI híres arról, hogy olyan dokumentációt ír, amely túlságosan is terjedelmes és esetleg zavaros, még a legegyszerűbb funkciókhoz is. Ha egy tapasztalt emberi író szerkeszti vagy átírja a generált dokumentációt, hosszú távon időt takaríthatsz meg, és elkerülheted a frusztrációt.
- » Összefüggő funkciók, amelyeknek semmi köze egymáshoz. Minden emberi programozónak megvan a maga stílusa és sajátos módszere, amellyel megold dolgokat. Ez teszi összefüggővé és következetessé a kódot, amelyet egy ember vagy akár egy csapat ír. Egy generatív MI-eszköznek nincs stílusa, és funkcióról funkcióra gyökeresen eltérő döntéseket hozhat. Add meg a megfelelő kontextust a következtetések minimálisra csökkentése érdekében.

Amire az MI-alapú programozási asszisztensek nem képesek

A jelenlegi MI-alapú asszisztensek olyan eszközök, amelyek segíthetnek növelni a termelékenységedet, de jelenleg csak akkor működnek jól, ha rövid pórázon tartod őket.

Ahogy az MI-alapú programozópárok egyre kifinomultabbá válnak, és egyre nagyobb adathalmazokon tanítják be őket, jó esély van arra, hogy a generatív MI-eszközök ugyanúgy elsajátítják majd a kódolást, mint ahogy az MI megtanult sakkozni és minden más társasjátékot, kártyajátékot és videójátékot játszani. Figyelemre méltó azonban, hogy az MI-chatbotok még mindig nem tudják felvenni a versenyt még a legtapasztalatlanabb emberi kosárlabdázóval sem.

Kíváncsi vagy, mi marad az emberi programozóknak, ha az MI elhúz mellettünk? Arra kértem a ChatGPT-t, hogy állítsa össze azoknak a dolgoknak a listáját, amelyeket az emberi programozók meg tudnak csinálni, az MI pedig soha nem lesz rájuk képes. A következő listát a ChatGPT készítette, de némileg módosítottam, hogy stílusosabb és humorosabb legyen.

ChatGPT 4:

Íme néhány humoros különbség az MI-alapú programozási asszisztensek és az emberi programozók között:

- **A pontosvesszők helytelen elhelyezése, majd órákig tartó keresése:** Az MI-alapú programozási asszisztensek precízen alkalmazzák a szintaxist, de soha nem fogják megismerni azt az „örömet”, amikor rossz helyre tesznek egy pontosvesszőt, és órákig hibakereséssel töltik az hibátidőt, hogy aztán rájöjjenek, hogy végig ott volt a szemük előtt.
- **Bőséges mennyiségű kávé elfogyasztása:** Bár az MI képes a nap 24 órájában kódot gyártani, soha nem fogja megtapasztalni azt a programozói rituálét, amely során bőséges mennyiségű kávéfogyasztanak el, hogy biztosítsák az üzemanyagot az éjszakai kódoláshoz.
- **Változók elnevezése háziállatokról vagy kedvenc rágcsálnivalókról:** Az MI javasolhat ugyan hatékony változóneveket, de nem fog kreatívan elnevezni egy változót egy szeretett háziállatról vagy egy kedvenc rágcsálnivalóról, például „bolyhosKiscicaSzámológép”-nek vagy „csokidarabosTömb”-nek.
- **A kisebb sikerek megünneplése győzelmi tánccal:** Csak az emberi programozóknak adatik meg, hogy győzelmi táncot járjanak, miután végre kijavítottak egy hibát, vagy működésre bírtak egy összetett függvényt.
- **Véletlenül kiváltott végtelen ciklus, amitől összeomlik a számítógépük:** Az MI túl logikusan gondolkodik ehhez. Nem tudja, hogy milyen izgalmas, amikor véletlenül végtelen ciklust hoz létre, és végignézi, hogy minden lefagy, majd rituálisan újra kell indítania a számítógépet.
- **A közösségi média vagy a macskás videók által elvont figyelem:** Az MI-alapú programozói asszisztensek a feladatra koncentrálnak, és soha nem szippantja be őket a közösségi média, és nem néznek meg „csak még egy” macskás videót, mielőtt folytatnák a munkát.
- **Gumikacsák használata a hibakereséshez:** Az MI nem használ gumikacsát (vagy bármilyen élettelen tárgyat) ahhoz, hogy hibakeresés céljából hangosan elmagyarázza a kódját – ez egy egyedülálló emberi hibakeresési technika.
- **A rejtély átélése, amikor a kód működik, de nem tudod miért:** Az MI tudja, hogy miért működik a kódja, de az emberi programozók néha találkozhatnak azzal a misztikus jelenséggel, amikor a kód bármilyen logikus magyarázat nélkül tökéletesen működik.
- **Határozott vélemény a kódszerkesztőkről:** Az MI nem fog szenvedélyes vitákba bocsátkozni arról, hogy a Vim miért jobb, mint az Emacs, vagy hogy a szöközők miért jobbak a tabulátoroknál; ezek az emberi programozók hitvitái.
- **Olyan megjegyzések írása a kódhoz, mintha azok naplóbejegyzések lennének:** Az MI ragaszkodik a tényszerű megjegyzésekhez, de az emberi programozók a kódhoz fűzött megjegyzéseket néha személyes naplójegyzetekké változtatják, viccekkel, egzisztenciális kérdésekkel vagy merengéssel az ebédre.