

POGOVOR Z LLM – CHAT GPT

JAZ: Izberi si poljubno temo iz učnega načrta za računalništvo v osnovni šoli.

ChatGPT: Seveda, izbral bom temo "Uvod v algoritme".

Algoritmi so bistveni koncept v računalništvu. V osnovni šoli bi se lahko otroci naučili osnovne koncepte algoritmov, kot so koraki za reševanje problema ali izvajanje določenih nalog. Na primer, lahko bi se naučili, kako sestaviti algoritem za seštevanje dveh števil ali za iskanje največjega elementa v seznamu števil.

Pomembno je, da se otroci naučijo, da algoritem ni le zaporedje korakov, temveč tudi način razmišljanja, ki omogoča reševanje problemov na sistematičen način. Lahko bi se učili tudi osnovnih konceptov, kot so zanke (loops) in pogoji (conditions), ki omogočajo izvajanje določenih korakov večkrat ali glede na določene pogoje.

Na ta način bi otroci dobili temeljno razumevanje računalniškega razmišljanja in problemov, kar bi jim koristilo pri nadaljnjem učenju računalništva in tudi pri reševanju problemov v vsakdanjem življenju.

JAZ: Glede na učni načrt, pripravi načrt obravnave teme "Uvod v algoritme", ki vsebuje:

- operativne učne cilje,
- izbrano programsko opremo oziroma aplikacijo,
- navodila za izvedbo aktivnosti, s katerimi bodo učenci usvajali učne cilje,
- preverjanje znanja, s katerim bo učitelj preveril znanje učencev.

ChatGPT: Tema: Uvod v algoritme

Operativni učni cilji:

1. Razumeti osnovne koncepte algoritmov.
2. Sestaviti preproste algoritme za reševanje problemov.
3. Uporabljati osnovne konstrukte algoritmov, kot so zanke in pogoji.
4. Razviti sposobnost sistematičnega reševanja problemov.

Programsko orodje:

Za izvedbo aktivnosti bomo uporabili spletno aplikacijo "Blockly". Blockly je vizualni programski jezik, ki omogoča uporabnikom, da sestavljajo programe sestavljenih iz blokov, kar omogoča učencem, da vizualno razvijajo algoritme brez potrebe po znanju o sintaksi programskih jezikov.

Navodila za izvedbo aktivnosti:

1. **Predstavitev konceptov:** Učitelj bo učencem predstavil osnovne koncepte algoritmov, kot so koraki, zanke in pogoji, s poudarkom na njihovi pomembnosti pri reševanju problemov.
2. **Ustvarjanje preprostih algoritmov:** Učenci bodo uporabili aplikacijo Blockly za sestavljanje preprostih algoritmov, kot so algoritem za seštevanje dveh števil ali algoritem za iskanje največjega elementa v seznamu.
3. **Uporaba zank in pogojev:** Učenci bodo dodali zanke in pogoje v svoje algoritme, da bi bolje razumeli, kako lahko algoritmi obravnavajo različne situacije.
4. **Razvoj lastnih algoritmov:** Učenci bodo samostojno razvili algoritme za reševanje preprostih problemov, kot so izračun povprečja seznamov števil ali iskanje določenega elementa v seznamu.
5. **Pogovor in analiza:** Učitelj bo spodbujal učence, da delijo svoje algoritme s sošolci in razpravljajo o različnih pristopih k reševanju istega problema.

Preverjanje znanja:

Učitelj bo preveril znanje učencev z naslednjimi metodami:

- Spremljanje učenčevega napredka med izvajanjem aktivnosti.
- Individualne naloge, kjer bodo učenci samostojno sestavljali algoritme za določene probleme.
- Skupinske diskusije, kjer bodo učenci predstavljali svoje algoritme in jih razložili sošolcem.
- Končni preizkus z različnimi nivoji kompleksnosti, kjer bodo učenci sestavljali algoritme za različne naloge.

JAZ: Navedi bolj konkretna navodila za izvedbo aktivnosti.

ChatGPT: Seveda, spodaj podajam bolj konkretna navodila za izvedbo aktivnosti za učence:

Navodila za izvedbo aktivnosti:

1. **Predstavitev konceptov algoritmov:**
 - Učitelj bo na začetku razložil osnovne koncepte algoritmov, kot so koraki, zanke in pogoji, s poudarkom na njihovi pomembnosti pri reševanju problemov.
 - Uporabljajo se lahko primeri iz vsakdanjega življenja, kot je recept za pripravo sendviča, da bi bolje razumeli koncepte.

2. Uporaba aplikacije Blockly:

- Učitelj bo učencem pokazal, kako uporabljati aplikacijo Blockly za sestavljanje algoritmov s pomočjo blokov.
- Učenci bodo sledili navodilom učitelja in poskušali sestaviti preproste algoritme, na primer algoritem za seštevanje dveh števil.

3. Dodajanje zank in pogojev:

- Učitelj bo predstavil koncepte zank in pogojev ter pokazal, kako jih lahko vključimo v algoritme.
- Učenci bodo poskusili dodati zanke in pogoje v svoje algoritme, da bi bolje razumeli, kako lahko algoritmi obravnavajo različne situacije.

4. Razvoj lastnih algoritmov:

- Učenci bodo samostojno razvili algoritme za reševanje preprostih problemov.
- Lahko se jim postavi izziv, da razvijejo algoritem za izračun povprečja seznamov števil ali iskanje določenega elementa v seznamu.

5. Pogovor in analiza:

- Učitelj bo spodbujal učence, da delijo svoje algoritme s sošolci in razpravljajo o različnih pristopih k reševanju istega problema.
- Učitelj bo postavljajl vprašanja, ki spodbujajo razmišljanje o učinkovitosti algoritmov in možnostih izboljšav.

Preverjanje znanja:

- Spremljanje učenčevega napredka med izvajanjem aktivnosti.
- Individualne naloge, kjer bodo učenci samostojno sestavljali algoritme za določene probleme.
- Skupinske diskusije, kjer bodo učenci predstavljali svoje algoritme in jih razložili sošolcem.
- Končni preizkus z različnimi nivoji kompleksnosti, kjer bodo učenci sestavljali algoritme za različne naloge.

JAZ: Sestavi preverjanje znanja na temo "Uvod v algoritme", s katerim bo učitelj preveril znanje učencev.

ChatGPT: Za preverjanje znanja učencev na temo "Uvod v algoritme" lahko uporabimo naslednje metode:

1. Test z raznolikimi nalogami:

- Pripravimo test, ki vključuje različne vrste nalog, kot so reševanje preprostih matematičnih problemov s pomočjo algoritmov, identifikacija in poprava napak v algoritmih, ter načrtovanje algoritmov za določene scenarije.
- Naloge naj bodo oblikovane tako, da preverjajo razumevanje osnovnih konceptov algoritmov, kot so koraki, zanke in pogoji, ter sposobnost njihove uporabe v praksi.

2. Praktične naloge s programiranjem:

- Učencem lahko postavimo nalogo, da v aplikaciji Blockly ali podobni vizualni programski okolici sestavijo algoritem za reševanje določenega problema.
- Spremljamo njihov napredek in kakovost algoritma ter ocenimo njihovo razumevanje konceptov algoritmov in njihovo sposobnost uporabe v praksi.

3. Ustno preverjanje:

- Učitelj lahko z učenci izvaja ustna vprašanja, ki preverjajo njihovo razumevanje osnovnih konceptov algoritmov in njihovo sposobnost pojasnjevanja algoritmov ter njihovega delovanja.
- Vprašanja lahko zajemajo primerjavo različnih pristopov k reševanju problema, razlago delovanja določenega algoritma ali prepoznavanje napak v algoritmih.

4. Projektna naloga:

- Učenci lahko dobijo nalogo, da samostojno razvijejo algoritem za reševanje določenega problema.
- Pri ocenjevanju projektnih nalog lahko upoštevamo njihovo kreativnost, učinkovitost algoritma, razumevanje konceptov ter sposobnost implementacije algoritma.

Z uporabo teh metod lahko učitelj pridobi celovit vpogled v znanje učencev na temo "Uvod v algoritme" ter oceni njihovo razumevanje konceptov in sposobnost njihove uporabe v praksi.

JAZ: Sestavi konkretno preverjanje znanja.

ChatGPT: Seveda, tukaj je konkreten primer preverjanja znanja na temo "Uvod v algoritme":

Preverjanje znanja - Uvod v algoritme

Test

1. Naloga 1: Izdelava algoritma

- Sestavite algoritem v Blocklyju za izračun povprečja petih števil. Uporabite zanko za ponavljanje vnosov in pravilno uporabite spremenljivke za shranjevanje vrednosti vsote in števila števil.

2. Naloga 2: Identifikacija napake

- Spodaj je podan algoritem za iskanje največjega števila v seznamu. Identificirajte napako v algoritmu in jo popravite:

```
max_number = 0
numbers = [5, 3, 9, 7, 2]
for num in numbers:
    if num > max_number:
        max_number = num
print("Največje število je:", max_number)
```

3. Naloga 3: Načrtovanje algoritma

- Predstavljena je naloga: "Poiščite vsoto prvih desetih naravnih števil." Načrtujte algoritem, ki bo izvedel to nalogo. Uporabite zanko za ponavljanje in spremenljivko za shranjevanje vsote.

Ocena:

- Vsaka naloga je vredna 10 točk.
- Za vsako pravilno rešeno nalogo se podeli 10 točk.
- Skupna ocena je 30 točk.

Navodila učitelju za izvedbo preverjanja:

1. Razdelite testne liste učencem.
2. Dajte jim dovolj časa za reševanje nalog.
3. Po končanem času zbirajte teste.
4. Preverite odgovore in ocenite pravilnost rešitev.
5. Za napake v rešitvah nudite povratne informacije in pojasnite, kje so naredili napako ter kako jo popraviti.
6. Končno ocenite vsakega učenca glede na dosežene točke in dodelite ustrezne ocene.

JAZ: Sestavi konkretno preverjanje znanja na temo "Uvod v algoritme" za aplikacijo Blockly.

ChatGPT: Seveda, tukaj je konkretno preverjanje znanja na temo "Uvod v algoritme" za aplikacijo Blockly:

Preverjanje znanja v aplikaciji Blockly - Uvod v algoritme

Naloga 1: Izdelava algoritma za seštevanje

1. Odprite aplikacijo Blockly.
2. Uporabite bloke za sestavljanje algoritma, ki bo seštel dve števili.
3. Preverite, ali vaš algoritem pravilno izračuna vsoto dveh števil.
4. Shrani svoj algoritem pod imenom "Seštevanje".

Naloga 2: Sestavljanje algoritma zanko

1. V algoritem "Seštevanje" dodajte zanko, ki bo omogočila večkratno izvajanje seštevanja.
2. Uporabite bloke za ponavljanje, da se seštevanje izvede desetkrat.
3. Preverite, ali vaš algoritem pravilno izračuna vsoto prvih desetih naravnih števil.
4. Shrani spremenjen algoritem pod imenom "Vsota prvih 10".

Naloga 3: Sestavljanje algoritma z zankami in pogoji

1. Ustvarite nov algoritem, ki bo iskal največje število v danem seznamu.
2. Uporabite zanko za pregled vsakega števila v seznamu.
3. Dodajte pogoj, ki bo primerjal trenutno število s trenutno največjim in ga posodobil, če je potrebno.
4. Preverite, ali vaš algoritem pravilno najde največje število v seznamu.
5. Shrani svoj algoritem pod imenom "Iskanje največjega".

Navodila učitelju za izvedbo preverjanja:

1. Pred začetkom preverjanja zagotovite, da so učenci seznanjeni z uporabo aplikacije Blockly.
2. Razdelite učencem navodila za vsako nalogo.
3. Dovolite učencem dovolj časa za reševanje vsake naloge.
4. Po končanem času preverite učenčeve algoritme in preverite njihovo pravilnost.
5. Za dodatno razpravo lahko učence vprašate, kako so zasnovali svoje algoritme in zakaj so izbrali določene bloke.
6. Oceni vsakega učenca na podlagi pravilnosti in učinkovitosti njihovih algoritmov ter jim nudite povratne informacije o njihovem delu.