## Integreerimise metoodika

### Esimene harjutusviis. Koodi assistent

Antud ülesannete lahendamise kord

1. Jagage oma kodeerimiseesmärk väiksemateks alamülesanneteks (kodeerimisülesanneteks). Seejärel kujundage mõttes, kuidas juhendaksite teist arendajat iga sammuga tegelema. See aitab teil pakkuda AI-le keskendunud päringute loogilist jada.
2. Küsige tehisintellekti teooria või ülesande osade kohta, mis on mõistmiseks liiga keerulised, ja paluge anda ka vastuseid näidetega. Kasutage AI-d ideede generaatorina ja integreerige ideid või näiteid, kommenteerides ja vastavalt tingimustele ja nõuetele koodi ümber kirjutades.
3. Keeruliste kodeerimisülesannete puhul jagage töö veelgi edasi etappideks (Vt Joonis 6. Keerulise koodiülesande jaotuse etapid). Küsi iga etapi kohta eraldi.



Joonis 6. Keerulise koodiülesande jaotuse etapid

1. Kirjeldage, millised on erinevad olukorrad, millega programm võib kokku puutuda ja kuidas neid olukordi käsitleda.
2. Laske tehisintellektil koostada selge dokumentatsioon ja kommentaarid, mis selgitavad koodi iga osa. Küsi lühikest selgitust, samm-sammult koos koodi eesmärgiga kommentaarides. Hästi dokumenteeritud kood võimaldab tulevastel toimetajatel hõlpsasti mõista tehisintellekti loogikat ja kavatsusi.
3. Testige tehisintellekti koodi põhjalikult ja kontrollige seda vigade, jõudlusprobleemide, semantiliste vigade ja turvaaukude suhtes.

Praktilise ülesande lõpus peaks õppejõud aga graafiliselt koostatud või genereeritud algoritmid üle vaatama, või viima läbi küsimustik/arutelu, et saada aru, millisel tasemel on üliõpilane teooriast aru saanud ja kas ta saab kodutööga praktiliselt hakkama.

Sellist tööviis saab teha mis tahes praktilise ülesande puhul, kui tutvustatakse uut teooria materjali, kuid autor soovitab kaks nädalat enne esimest kodutöö ülesannet eraldada paar tundi, et kontrollida, kas kõik üliõpilased osalevad umbes samas tempos. Kui keegi jääb maha, võimaldab see meetod neil järele jõuda ja ülesande õigeaegselt täita.

### Teine harjutusviis. Koodiülevaatamine

1. Esitage 2–3 näidet, mis näitavad teie või kellegi teise loodud koodi.
2. Selgitage lühidalt selle koodi eesmärki. Küsige soovitusi, milles võib viga olla ja kuidas seda parandada.
3. Tooge näiteid servajuhtumite, vigade (tõrketeadete) ja piirtingimuste kohta, mida soovite, et loogika käsitleks.
4. Paluge tehisintellektil esitada selged kommentaarid ja analüüsid, mis selgitavad koodi iga osa koos parandusettepanekutega.
5. Kasutage koodi muutmiseks antud teadmisi.

### Kolmas harjutusviis. Ümberehitamine

Tudengitel tuleb valida teema ja sellel teemal üks tegevus, alates sortimistehnikatest kuni mälujaotuseni.

Tööprotsessi sammud:

1. Sõnastage põhiülesanded ja küsimused teemast.
2. Esitage 2–3 näidet, mis näitavad, millist koodi soovite luua. Oluliselt mainida erinevaid väljavõtteid, mis vormindavad andmeid.
3. Kasutage funktsiooni "Ümberehitamine" [19] (Vt Järeldus, lk 54; Joonis 4. Funktsioonid. BlackBox, lk 52, ja Joonis 7. Funktsioonid. ChatGPT, lk 57, mõlemal märgitud kollasega), et näha, kuidas iga tööriist soovitab erinevaid lahendusi. Ümberehitamine esitab uut versiooni, kuna on võimalik vaadata kõiki versioone (Vt Joonis 7. Funktsioonid. ChatGPT, lk 57, esimene nimikiri „3/3“), või ja antud koodi jaoks teist/alternatiivset lähenemist. BlackBox’il aga versiooni salvestamise funktsioon puudub, saadud koodi või vastuse saab hilisemaks võrdlemiseks dokumenti salvestada. Või on võimalik veebiversioonis samas aknas avada uus vestlus. (Vt BlackBox, lk 52)



Joonis 7. Funktsioonid. ChatGPT

1. Vaadake üle AI tulemused ja tõstke esile kodeerimise erinevused. Võtke arvesse selliseid tegureid nagu tõhusus, loetavus ja kaasaegsed kodeerimistavad.

Erinevate stsenaariumite näidete vaatamine ja analüüsimine suurendab teadmiste baasi, mis annab omal moel võimaluse valida iga konkreetse ülesande jaoks optimaalne lahendus.

### Neljas harjutusviis. Koostöö

Tudengid peavad jagunema paaridesse nii, et igaüks teeb 2 harjutust. Harjutused koosnevad 2 osast.

Harjutuse esimene osa on etteantud funktsiooni rakendamine, luues korrelatsioonidiagrammi, mille saab AI abil pseudokoodiks tõlkida.

Ülesande teine osa sisaldab töö esimesest osast saadud diagrammi/pseudokoodi realiseerimist programmeerimiskeeles (näiteks С) kirjutatud algoritmis, kasutades jällegi AI.

Töö vajalikud sammud:

1. Võtke erinevaid harjutusi.
2. Viige end selle funktsiooniga kurssi.
3. Koostage korrelatsiooniskeem (UML-algoritm), mis sisaldab harjutuste jaoks vajalikke objekte, alamfunktsioone, klasse jne.
4. Pseudokoodi saamiseks pöörduge tehisintellekti poole.
5. Kontrollige saadud koodi õigsust.
6. Vajadusel korrake punkti 4. sammu.
7. Vahetage tulemusi oma partneriga ja meeskonnakaaslase koodi vastu.
8. Kasutage tehisintellekti näpunäiteid ja kirjutage/paluge tehisintellektil kirjutada töötav programmikood programmeerimiskeeles.

Kokkuvõttes oli rühmatöö mitte ainult tehniline harjutus, vaid ka oluline sotsiaalsete ja professionaalsete oskuste arendamise vahend. See ülesanne andis üliõpilastele võimaluse oma oskusi demonstreerida ja arendada, töötades koos kaaslastega keerukate probleemide lahendamisel.