Završni zadatak OOP

Napravite program koji će riješiti neki problem kojega sami izmislite. Možete izmijeniti i neki postojeći zadatak (npr. C++ Casino) na način da mu kod izmijenite da bude objektno orijentiran. Neka program ima nekakvu svrhu (neka ne bude banalan) i neka ima nekoliko različitih funkcionalnosti (npr. Bankomat – pregled stanja, podizanje novca, itd..; Knjižnica – posuđivanje i vraćanje knjiga). Problem treba najprije modelirati – u MS Wordu napisati use case-eve te kasnije u draw.io napraviti potrebne dijagrame, a tek nakon toga prebaciti u kod. Kod mora prikazivati koncepte apstrakcije i enkapsulacije, a po potrebi i nasljeđivanja i polimorfizma.

VAŽNO: Morate sami razmisliti o svom programu. Zadovoljava li on potrebe korisnika? Ima li dovoljno funkcionalnosti? Jesu li funkcionalnosti koje ste odabrali zbilja potrebne te postoje li neke koje su potrebne, a izostavili ste ih? Razmišljajte na način da želite riješiti neki problem iz realnog svijeta.

Završni zadatak OOP

Napravite program koji će riješiti neki problem kojega sami izmislite. Možete izmijeniti i neki postojeći zadatak (npr. C++ Casino) na način da mu kod izmijenite da bude objektno orijentiran. Neka program ima nekakvu svrhu (neka ne bude banalan) i neka ima nekoliko različitih funkcionalnosti (npr. Bankomat – pregled stanja, podizanje novca, itd..; Knjižnica – posuđivanje i vraćanje knjiga). Problem treba najprije modelirati – u MS Wordu napisati use case-eve te kasnije u draw.io napraviti potrebne dijagrame, a tek nakon toga prebaciti u kod. Kod mora prikazivati koncepte apstrakcije i enkapsulacije, a po potrebi i nasljeđivanja i polimorfizma.

VAŽNO: Morate sami razmisliti o svom programu. Zadovoljava li on potrebe korisnika? Ima li dovoljno funkcionalnosti? Jesu li funkcionalnosti koje ste odabrali zbilja potrebne te postoje li neke koje su potrebne, a izostavili ste ih? Razmišljajte na način da želite riješiti neki problem iz realnog svijeta.

Završni zadatak OOP

Napravite program koji će riješiti neki problem kojega sami izmislite. Možete izmijeniti i neki postojeći zadatak (npr. C++ Casino) na način da mu kod izmijenite da bude objektno orijentiran. Neka program ima nekakvu svrhu (neka ne bude banalan) i neka ima nekoliko različitih funkcionalnosti (npr. Bankomat – pregled stanja, podizanje novca, itd..; Knjižnica – posuđivanje i vraćanje knjiga). Problem treba najprije modelirati – u MS Wordu napisati use case-eve te kasnije u draw.io napraviti potrebne dijagrame, a tek nakon toga prebaciti u kod. Kod mora prikazivati koncepte apstrakcije i enkapsulacije, a po potrebi i nasljeđivanja i polimorfizma.

VAŽNO: Morate sami razmisliti o svom programu. Zadovoljava li on potrebe korisnika? Ima li dovoljno funkcionalnosti? Jesu li funkcionalnosti koje ste odabrali zbilja potrebne te postoje li neke koje su potrebne, a izostavili ste ih? Razmišljajte na način da želite riješiti neki problem iz realnog svijeta.