

Projektowanie i programowanie obiektowe

Paradygmat obiektowy

- modelowanie świata za pomocą obiektów

mgr inż. Krzysztof Rewak

3 października 2017

1 Programowanie strukturalne a obiektowe

Należy skompilować i uruchomić w dowolnym środowisku programistycznym załączone pliki `lab02a.cpp` oraz `lab02b.cpp`. Oba programy wykonują dokładnie to samo zadanie: generują listę n studentów z losowymi numerami indeksów, a następnie wypisują te numery indeksów na standardowym wyjściu. Plik `lab02a.cpp` został napisany w formie strukturalnej, natomiast `lab02b.cpp` - zgodnie z regułami programowania obiektowego.

Należy utworzyć plik `lab02.cpp`, którego zawartość będzie rozszerzeniem funkcjonalnym załączonych do zadania programów:

- student, poza numerem indeksu, powinien posiadać też imię i nazwisko; proponowanym rozwiązaniem jest utworzenie tablice stringów, które będą przechowywały przykładowe imiona i nazwiska oraz metody, która będzie je losowała i przydzielała studentom;
- student, poza numerem indeksu, imieniem i nazwiskiem, powinien móc mieć ustawiony status studenta; proponowanym rozwiązaniem jest utworzenie binarnej zmiennej, która będzie wskazywała czy student jest aktywny, czy nie; można wykorzystać metodę `rand()`, aby losowo dopasować każdemu studentowi jego status;
- program powinien na końcu wypisać na standardowym wyjściu listę wygenerowanych aktywnych studentów w formie `nazwisko imię (numer indeksu)`.

2 System kontroli wersji

Należy utworzyć konto użytkownika na dowolnym systemie hostującym system kontroli wersji Git; proponowane rozwiązania to Github lub Bitbucket. Po zakończeniu procedury aktywacyjnej, należy utworzyć repozytorium o nazwie `ppo`.

Należy zainstalować i uruchomić Git for Windows (<https://git-scm.com/download/win>) oraz dowolne oprogramowanie służące do zarządzania repozytoriami systemu kontroli wersji; proponowane rozwiązanie to SourceTree z graficznym interfejsem użytkownika lub Cmder jako rozwiązanie w terminalu. Należy zainicjować repozytorium w dowolnym folderze, dodać folder z zadaniem z punktu 1., dodać pliki do repozytorium, zacommitować zadanie oraz wysłać je na serwer.

Zbiór instrukcji do wykonania z poziomu terminala:

```
git init
git remote add origin http://user@repo/user/ppo.git
git pull origin master
git add .
git commit -am "Zadanie laboratoryjne 2"
git push origin master
```

Przed dodaniem plików do repozytorium warto dodać w katalogu głównym plik `.gitignore`, którego treścią będzie jednolinijkowiec `*.exe`. Uniemożliwi to dodawanie plików wykonawczych do repozytorium, co drastycznie zmniejszy jego wielkość.