



## Четврти дан

### Вежба 1

Циљ овог задатка је имплементирати свој тип јединственог показивача (`MyUniquePtr`) који ће имати основне могућности библиотечног типа `unique_ptr`. У `main.cpp` датотеци налази се код који користи `unique_ptr`. Ваш тип јединственог показивача би требало да може заменити `unique_ptr` у том коду.

У датотеци `MyUniquePtr.h` дефинисати шаблонски тип `MyUniquePtr`. Претпроцесорским директивама `#define` на почетку датотеке `main.cpp` заменити `unique_ptr` са `MyUniquePtr`. Дати код треба да се након тога преводи и понаша исто, а закоментарисани код (блок „GRESKE“) треба да проузрокује грешку током превођења. Обратите пажњу и на `@TODO` коментар на крају функције `main`.

Напомена: Оператор `->` је мали изузетак. Пише се као метода и не прима параметре. Треба да врати показивач на тип који има поље које одговара називу са десне стране стрелице.

### Вежба 2

У овом задатку треба да се упознате са једним примером употребе дељеног показивача (`shared_ptr`). Уједно, треба да уочите какав проблем изазива циклична зависнот дељених показивача и како се тај проблем може решити употребом слабих показивача (`weak_ptr`).

1. Проширити класу `Node` са дељеним показивачем на претка, нпр. поље `parent` (ако нема претка, `parent` у чвору треба да буде `nullptr`). Поставити те показиваче на одговарајуће вредност у функцији `foo`. Проширити функцију `toString` да испишује и информацију о претку (само његово име, не целе податке; зашто је ово важно?). Анализирати резултат.

2. Решити проблем из претходног корака увођењем слабог показивача на претка, уместо дељеног показивача.