

Zadatak

Implementirati sljedeći *Observer pattern*.

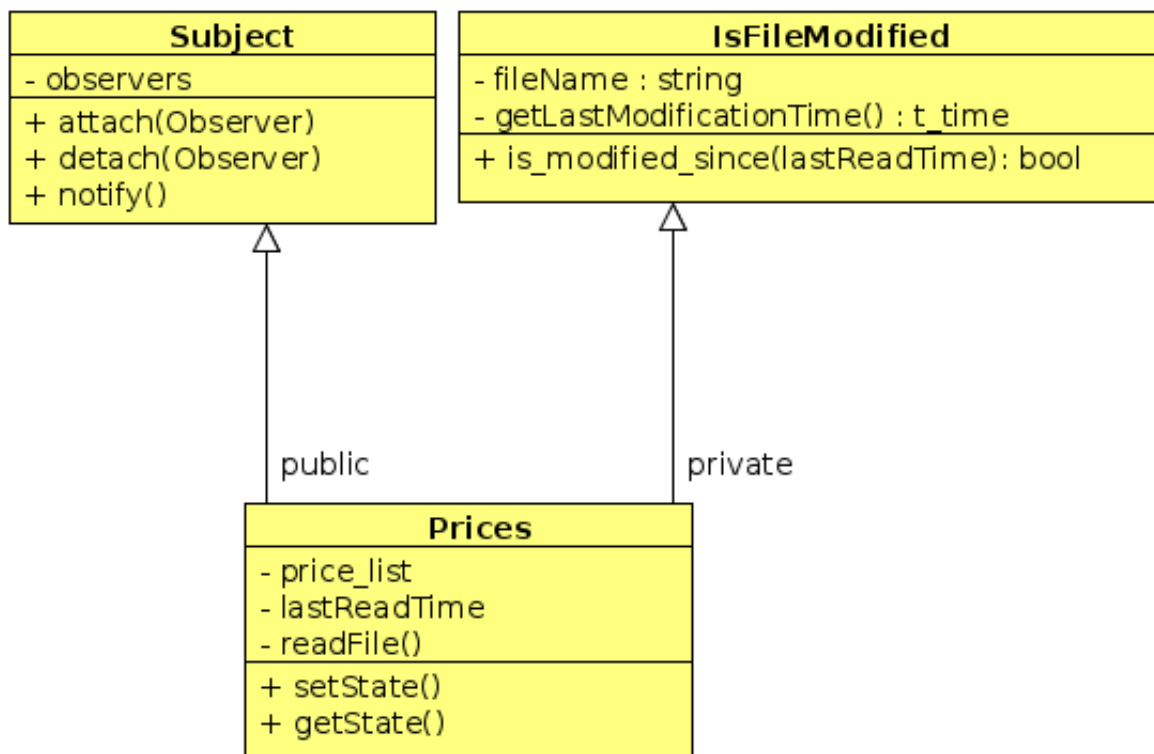
Subjekt je klasa `Prices` koja čita iz datoteke listu proizvoda i cijena i čuva ju interno u nekom spremniku (npr u `std::map`). Glavni program funkcionira tako što u jednoj beskonačnoj petlji periodički poziva `setState()` metodu na `Prices` objektu. Metoda `setState()` poziva `notify()` ako je datoteka u međuvremenu modificirana.

Promatrač `DisplayInfo` ispisuje listu cijena na ekranu. Posljedica Observer patterna je da se poslije svakog editiranja datoteke na ekranu ispisuje nova lista cijena.

Implementacijski detalji

`Prices` mora čuvati vrijeme zadnjeg čitanja datoteke te ga u `setState()` uspoređivati s vremenom zadnje modifikacije. Ako se modifikacija desila iza zadnjeg čitanja stanje Subjekta se ažurira, dok u suprotnom metoda ne radi ništa. Ako je stanje ažurirano metoda poziva `notify()`.

Ispitivanje vremena zadnje modifikacije datoteke treba enkapsulirati u **privatnu** baznu klasu Subjekta.



Nalaženje vremena modifikacije datoteke

Ovdje ćemo implementirati C-kod. Treba uključiti zaglavlja `<ctime>` i `<sys/stat.h>`. Metoda `stat()` ispunjava strukturu `stat` koja sažima informacije o datoteci:

```

struct stat {
    dev_t      st_dev;      /* ID of device containing file */
    ino_t      st_ino;      /* inode number */
    mode_t     st_mode;     /* protection */
    nlink_t    st_nlink;    /* number of hard links */

```

```

uid_t      st_uid;      /* user ID of owner */
gid_t      st_gid;      /* group ID of owner */
dev_t      st_rdev;     /* device ID (if special file) */
off_t      st_size;     /* total size, in bytes */
blksize_t  st_blksize;  /* blocksize for file system I/O */
blkcnt_t   st_blocks;   /* number of 512B blocks allocated */
time_t     st_atime;    /* time of last access */
time_t     st_mtime;    /* time of last modification */
time_t     st_ctime;    /* time of last status change */
};

```

Tražena informacija je u varijabli `st_mtime`.

```

time_t IsFileModified::getLastModificationTime(){
    // C-kod za Unix platformu
    struct stat fileInfo;
    if(stat (m_fileName.c_str(),&fileInfo) != 0){
        throw "stat error!";
    }
    return fileInfo.st_mtime;
}

```

Na Windows platformi treba `stat` zamijeniti sa `_stat`:

```

time_t IsFileModified::getLastModificationTime(){
    // C-kod za Unix platformu
    struct _stat fileInfo;
    if(_stat (m_fileName.c_str(),&fileInfo) != 0){
        throw "stat error!";
    }
    return fileInfo.st_mtime;
}

```

Molim poštovati sljedeća pravila:

- Sve deklaracije klasa, u koje mogu doći samo implementacije jednostavnih metoda, smjestiti u datoteku imena **FileObserver.h**. Implementacije svih drugih metoda staviti u datoteku imena **FileObserver.cpp**.
- Koristiti sljedeći glavni program (**main.cpp**)

```

#include <iostream>
#include <thread>
#include <chrono>
#include <cstdlib>

#include "FileObserver.h"

// Na komandnoj liniji se zadaje ime ulazne datoteke
int main(int argc, char *argv[]) {
    if (argc != 2) {
        std::cout << "Usage: " << argv[0] << " file_name\n";
        return EXIT_FAILURE;
    }
    // konkretni subjekt
    Prices test(argv[1]);
    // konkretni observer
    DisplayInfo display(&test);

    while (true) {
        test.setState();
        // spavaj jednu sekundu
        std::this_thread::sleep_for(std::chrono::milliseconds(1000));
    }
}

```

```
    }  
    return EXIT_SUCCESS;  
}
```

- Primjer ulazne datoteke (**cijene.txt**):

```
banane : 9.99  
bomboni : 8.5  
cokolada : 11.99  
jabuke : 12.1  
salata : 3.13
```