

Zadanie polega na napisaniu skryptu implementującego prosty organizer, który umożliwia zarządzanie zadaniami do wykonania. Skrypt uruchamia się poleceniem "todo" z parametrami opisanymi poniżej. Wszystkie potrzebne dane skrypt pobiera i przechowuje w katalogu ".todo" w katalogu domowym użytkownika, który skrypt wywołał.

Zadaniem systemu TODO jest zarządzanie wydarzeniami, z których każde jest opisane przez swój unikalny numer identyfikacyjny (id), datę, treść i priorytet (liczba naturalna), na przykład:

13: 2005-10-30, napisać maila do Agaty, 20

14: 2005-10-23, iść na wybory, 100

Skrypt dopuszcza wiele różnych wywołań, które są omówione poniżej. W każdym wypadku należy po wykonaniu polecenia wypisać odpowiedni przyjemny dla oka komunikat informujący z grubsza co się stało.

ADD

./todo add _data _treść [_priorytet]

Dodaj nowe wydarzenie na zadany dzień. Ostatni parametr jest opcjonalny, jeżeli nie będzie podany przydzielamy wydarzeniu priorytet 0. Jeżeli data wypada w przeszłości lub dalej niż rok naprzód należy wyświetlić odp. komunikat i odmówić wykonania operacji. Automatycznie należy dokonać przydziału nowego (unikalnego) numeru id dla wydarzenia.

Przykład:

```
./todo add 2005-11-11 "kup koniecznie ciacho" 10
```

Dodano na 2005-11-11 z id=321.

LIST

./todo list [_data]

====

Wyświetl wszystkie zadania zaplanowane na daną datę. Parametr jest opcjonalny, w razie jego braku chodzi o zadania na dzisiaj. Należy wyświetlić id zadania, treść i priorytet, posortowane malejąco po priorytecie. Mogłby się przydać jakiś nagłówek.

Przykład:

```
./todo list 2005-10-30
```

ID: TREŚĆ, PRIORYTET

13: napisać maila do Agaty, 20

76: wymienić zasłonkę, 3

DEL

./todo del _id

Usuń wydarzenie o zadanym identyfikatorze.

Przykład:

```
./todo del 13
```

Usunięto id=13 na 2005-10-30: napisać maila do Agaty.

MOVE

./todo move _id _new_date

Przesuń zadanie o numerze _id z jego obecnej daty na dzień _new_date. Poza tym nic nie zmieniaj.

Przykład:

```
./todo move 13 2006-05-11
```

Przesunięto: napisać maila do Agaty z 2005-10-30 na 2006-05-13.

OLD

`./todo old move [_inc] / del / list [p/d]`

Zrób coś ze wszystkimi przeterminowanymi zadaniami: albo usuń (del), albo przesuń je do zadań dzisiejszych (move), albo po prostu wypisz (list). W przypadku opcji move drugi parametr _inc mówi o ile należy zwiększyć priorytet każdego przenoszonego zadania. W przypadku opcji list drugi parametr mówi czy zadania należy sortować po priorytecie (p), czy po dacie (od najstarszych) i dopiero potem po priorytecie (d).

Przykład:

```
./todo old move 1
```

Przeniesiono na dzisiaj 2 zadań.

```
./todo old del
```

Usunięto 0 przeterminowanych zadań.

STAT

`./todo stat [_end_date]`

Wypisz statystykę zadań zaplanowanych na przyszłość od dziś aż do daty końcowej _end_date (opcjonalnej, jeżeli jej nie ma to do końca świata), oczywiście w kalendarzowej kolejności dni. Należy dla każdego dnia podać liczbę zadań zaplanowanych na ten dzień oraz ich średni priorytet (zapaleńcy statystyki mogą też policzyć wariancję :)).

Przykład:

```
./todo stat 2005-12-25
```

```
DATA: ILE, ŚR_PRI
```

```
-----
```

```
2005-12-06: 1, 23
```

```
2005-12-24: 2, 112
```

IMPORT

`./todo import _user [n]`

Zaimportuj zawartość bazy zadań użytkownika _user i dodaj je do swoich wydarzeń.

Jeżeli podano opcjonalną opcję n, to importujemy tylko wydarzenia nieprzeterminowane.

W razie braku jakiegokolwiek bazy u tego użytkownika, braku dostępu do niej (patrz niżej) etc. należy wypisać odpowiednie komunikaty. (Uwaga! nie można skopiować całkiem "na pałę" - dlaczego?)

Przykład:

```
./todo import zuzia324
```

Zaimportowano 23 zadań od zuzia324.

```
./todo import olek12 n
```

Brak uprawnień do odczytania bazy zadań użytkownika olek12.

BLOCK

`./todo block / unblock`

Zablokuj (block) lub odblokuj (unblock) dostęp do swojej bazy dla wszystkich poza sobą, poprzez modyfikację uprawnień do odpowiednich plików i/lub katalogów (nie przejmujemy się i nie ruszamy uprawnień do katalogu domowego - jeżeli inni nie mogą nawet przez niego przejść, to trudno, uznajemy, że to jest ustawienie ponad nami).

Przykład:

```
./todo block
```

Zablokowano bazę.

Uwagi:

Należy sprawdzać, czy liczba parametrów obowiązkowych jest taka jak powinna być.

W razie błędu (zły parametr, etc.) należy wypisać stosowny komunikat. Można założyć, że tam gdzie jako parametr ma wystąpić liczba, faktycznie będzie liczba itp.

Przy pierwszym uruchomieniu należy założyć potrzebne struktury w katalogu .todo. Nie kładziemy bardzo dużego nacisku na efektywność (no chyba, że sposób zaimplementowania jakiejś operacji będzie drastycznie wolny a można to bardzo łatwo zrobić szybciej np. przeglądanie całej listy wydarzeń w celu wybrania kolejnego unikalnego id).

Format daty jest do wyboru przez rozwiązującego. Wewnętrzna organizacja zawartości katalogu .todo podobnie.

[Dodaj pracę](#)

Status przesłanego zadania

Status przesłanego zadania	No submissions have been made yet
Stan oceniania	Nieocenione



Platforma obsługiwana przez:
[Centrum e-Learningu AGH](#)
[Centrum Rozwiązań Informatycznych AGH](#)

Pobierz aplikację mobilną



Wybierz język

