

Warsztat programisty – Zadanie projektowe.

Etap 2 - wykrywanie błędów

Łukasz Kuszner, Radosław Ziemann

23 grudnia 2025

Ten etap polega głównie na sprawdzeniu poprawności danych. Program powinien sprawdzać, czy wczytane dane wejściowe odpowiadają specyfikacji podanej w opisie etapu 1. Ponadto, ewentualne błędy wykonania nie mogą spowodować, że program zakończy się zgłaszając poprawne wykonanie (z kodem 0).

Uwaga: elementy podane do realizacji na tym etapie projektu nie wyczerpują wszystkich aspektów obsługi błędów. Obowiązkowe jest wykrycie, że wystąpił problem i zgłoszenie tego faktu programowi wywołującemu.

Sprawdzenie poprawności formatu danych

Wczytując dane należy sprawdzić, czy są one zgodne z opisem. W szczególności dla komendy `init`:

- Czy liczba n , jest istotnie liczbą z podanego zakresu?
- Czy kod każdego towaru jest zgodny z opisem?
- Czy nazwa nie jest za długa, albo czy nie zawiera niedozwolonych znaków?
- Czy liczba towarów odpowiada zadeklarowanej?
- Czy w danych dla komendy `init` nie podano dwa razy towaru o tym samym kodzie?

Dla innych komend podobnie.

Sprawdzenie poprawności scenariusza użycia

Zakładamy, że po utworzeniu magazynu komendą `init` w kolejnych wywołaniach obsługujemy sekwencje komend `update` i `print`.

W związku z tym może się zdarzyć, że poprawnie sformatowane dane prowadzą do niepoprawnego scenariusza użycia. Przykładowo

- podano komendę `update` lub `print` bez wcześniejszego utworzenia magazynu komendą `init`

- podano komendę `init` pomimo, że magazyn już istnieje
- dla komendy `update` podano kod towaru nie obsługiwany w magazynie
- wykonanie komendy `update` sprawiłoby, że liczba sztuk towaru stanie się ujemna.

Przykład

Przypuśćmy, że plik `i.txt` zawiera:

```
3
AB235 R12
SD121 R20
RS001 Zestaw baterii
```

a plik `u.txt` zawiera

```
2
AB234 + 8
RS001 + 2
```

Wtedy program uruchomiony w dowolnym trybie ze scenariuszem:

```
init i.txt
print report1211
update u.txt
update u.txt
print report1411
```

Powinien zgłosić błąd, gdyż towar o kodzie `AB234` nie został podany w danych dla komendy `init`.

Spodziewane efekty

- Projekt zawiera testy jednostkowe oraz elementy konfiguracji z etapu 1.
- W przypadku wystąpienia błędu w danych wejściowych (np: niepoprawna komenda, niepoprawny scenariusz) zwracany jest kod różny od zera, a na standardowy strumień błędów (`stderr`) wyświetlana jest informacja o błędzie (im dokładniejsza tym lepiej).
- Może się zdarzyć, że w poprawnym scenariuszu, dla komendy `print` plik nie powstanie. Wtedy program również musi się zakończyć zwracając kod różny od 0. Przykładowe błędy, które należy brać pod uwagę:
 - nie udało się otworzyć/odczytać pliku z danymi wejściowymi,
 - nie udało się zapisać danych do pliku opisującego stan magazynu,
 - wystąpił inny problem w trakcie wykonania, np: `pdflatex` zwrócił błędy, program napisany w C uległ awarii itp.