

Warsztat programisty – Zadanie projektowe.

Etap 3 - rozszerzenie funkcjonalności i generowanie dokumentów

Łukasz Kuszner, Radosław Ziemann

15 grudnia 2025

Ten etap polega na rozszerzeniu funkcjonalności z etapu 2 z wykorzystaniem poznanych narzędzi. Składają się na nie elementy obowiązkowe, nieobowiązkowe i ponadobowiązkowe.

Elementy obowiązkowe (20 pkt.)

- Spodziewane efekty z etapu 1, w tym konfiguracja przy pomocy `cmake`, historia zmian w repozytorium `git`.
- Spodziewane efekty z etapu 2, w tym testy jednostkowe i wykrywanie błędów z uwzględnieniem nowego formatu komend.
- Program obsługuje wiele magazynów.
- Dostępne są dodatkowe komendy `create` i `transfer`.
- Program obsługuje parametry komend oznaczone w opisie jako obowiązkowe.

Elementy nieobowiązkowe (20 pkt.)

Elementy nieobowiązkowe to takie, których można nie wykonać a mimo to zaliczyć projekt. Intencją jest, aby były to elementy niewymagające wiedzy spoza przerobionego materiału w ramach studiów.

Program, który nie implementuje wybranego elementu nieobowiązkowego powinien poprawnie wczytać dane, które go dotyczą, jednak nie musi sprawdzać ich poprawności.

Elementy ponadobowiązkowe (bez limitu punktów)

Za realizację każdego elementu można uzyskać dodatkowe punkty, co pozwala otrzymać ponad 100% punktów przewidzianych za ten etap projektu. W szczególności obsługa podkategorii towarów jest ponadobowiązkowa – żaden element obowiązkowy i nieobowiązkowy nie wymaga ich obsługi poza poprawnym wczytaniem danych.

Towary i magazyny

Na potrzeby realizacji 3 etapu projektu zakładamy, że każdy towar oprócz kodu i nazwy ma dodatkowe parametry:

- C – kategoria towaru (ang. category), jest to liczba całkowita z przedziału $[0, 10]$.
- S – podkategoria towaru (ang. subcategory), jest to liczba całkowita z przedziału $[0, 10]$. Ten parametr jest opcjonalny, to znaczy mogą istnieć towary, które nie należą do żadnej podkategorii.
- F – stopień łatwości, jest to liczba całkowita z przedziału $[0, 10]$. Zero oznacza towary całkowicie niepalne, natomiast 10 oznacza towary ekstremalnie łatwe. Stopień łatwości jest określany dla każdego towaru osobno.

Magazyny również mają swoje kody i nazwy. Dodatkowo każdy magazyn może przechowywać tylko towary o ograniczonym stopniu łatwości z wybranych kategorii i podkategorii.

Każdy magazyn ma również ograniczoną pojemność (ang. capacity) – może przechowywać tylko określoną liczbę towarów z każdej kategorii i podkategorii.

Ponadto, dla każdego magazynu i (pod)kategorii typu towaru w magazynie znana jest minimalna liczba sztuk, która zapewni ciągłość dostaw. To znaczy, jeśli liczba sztuk danej kategorii towaru spadnie poniżej tej wartości, to należy uzupełnić zapasy.

Komenda init

Komenda `init` służy do wczytania parametrów istotnych dla wszystkich magazynów, czyli listy dostępnych towarów.

Dane wejściowe: najpierw liczba całkowita n z przedziału $[1..1000]$, oznaczająca liczbę możliwych towarów, a w kolejnych n liniach dane o towarach w formacie:

```
<C>{.<S>} <F> <kod towaru> <nazwa towaru>
```

Kod towaru to napis złożony z pięciu znaków, z których pierwsze dwa to duże litery alfabetu łacińskiego, a kolejne trzy to cyfry. Nazwa towaru to napis złożony z małych i dużych liter alfabetu łacińskiego, cyfr i spacji zaczynający się od litery (formaty są takie jak na poprzednim etapie projektu).

Przykład

3

4.3 4 AB234 R12

4.5 4 SD121 R20

4 4 RS001 Zestaw baterii

W powyższym przykładzie mamy trzy możliwe towary. Dwa z nich mają ustalone podkategorie, a jeden towar należy bezpośrednio do kategorii 4 i nie ma ustalonej podkategorii. Wszystkie trzy mają stopień łatwopalności 4.

Elementy obowiązkowe: program wczytuje dane i sprawdza zgodność z formatem. Kody towarów nie mogą się powtarzać.

Komenda create

Wywołanie komendy `create` tworzy nowy magazyn wraz z parametrami.

Dane wejściowe: pierwsze dwie linie mają następujący format:

```
<kod magazynu> <całkowita pojemność> <F> <nazwa magazynu>
<n>
```

W pierwszej linii znajduje się kod magazynu (format taki jak dla kodów towarów) jego pojemność, maksymalny poziom łatwopalności przechowanych materiałów i nazwa magazynu (format taki jak dla nazw towarów). W kolejnej linii liczba całkowita: n z przedziału $[1..100]$, oznaczająca liczbę możliwych (pod)kategorii towarów przechowywanych w magazynie. Kolejnych n linii ma następujący format:

```
<C>{.<S>} <pojemnosc> <min>
```

Kolejne elementy to kategoria, i opcjonalnie podkategoria, towarów, pojemność magazynu przeznaczona na towary danej (pod)kategorii i oczekiwany stan minimalny. Warunek konieczny dla prawidłowych danych: $min \leq pojemnosc$.

Przykład

```
MG252 650 6 Nowy super magazyn 1
3
4.3 120 10
4.5 300 10
4 500 30
5 100 10
```

Powyższe dane oznaczają utworzenie magazynu o kodzie MG252, w którym możemy przechowywać towary kategorii 4 i 5. Łącznie nie możemy mieć więcej niż 650 wszystkich towarów, w tym towarów kategorii 4 nie może być więcej niż 500 a towarów kategorii 5 nie może być więcej niż 100. Dodatkowo mamy ograniczenia na podkategorie 4.3 i 4.5 oraz pożądane stany minimalne. Liczba 30 oznacza, że łączna liczba towarów z kategorii 4 i wszystkich podkategorii tej kategorii nie powinna być od niej mniejsza (nie powinna, ale może).

Elementy obowiązkowe: program wczytuje dane i sprawdza zgodność z formatem. Kody magazynów nie mogą się powtarzać. Program sprawdza, czy ograniczenia na pojemność towarów danej

kategori nie jest większa niż pojemność całego magazynu oraz, czy oczekiwane stany minimalne dla wszystkich kategorii łącznie nie przekraczają pojemności magazynu.

Elementy nieobowiązkowe: Program sprawdza, podobne warunki dla podkategorii: czy ograniczenia na pojemność towarów danej podkategorii nie jest większa niż pojemność na towary kategorii do której ona należy oraz czy oczekiwane stany minimalne dla wszystkich podkategorii łącznie nie przekraczają pojemności dla danej kategorii.

Komenda update

Komenda `update` działa tak jak poprzednio, przy czym dodatkowy element danych wejściowych mówi, którego magazynu dotyczy dana komenda. Na przykład:

```
MG252
2
AB234 + 8
RS001 + 2
```

oznacza, że komenda dotyczy magazynu `MG252`.

Elementy obowiązkowe: program wczytuje dane i sprawdza: zgodność z formatem, czy podany magazyn był wcześniej utworzony, ograniczenia na pojemność całego magazynu, zgodność z kategoriami przechowywanych towarów, ograniczenia na łatwopalność i maksymalną pojemność dla danej kategorii.

Elementy nieobowiązkowe: Sprawdzanie zgodności i ograniczeń na maksymalną pojemność dla podkategorii.

Komenda transfer

Komenda `transfer` powoduje przeniesienie towarów pomiędzy magazynami.

W pierwszej linii występują dwa kody magazynów: docelowego i źródłowego, a następnie jedna liczba całkowita n z przedziału $[1..100]$ oznaczająca liczbę towarów do przeniesienia. W kolejnych n liniach podane są: kod towaru i liczba naturalna z przedziału $[1..2000]$ oznaczająca, liczbę sztuk towaru do przeniesienia. Na przykład:

```
MG251 MG252
2
AB234 8
RS001 2
```

oznacza, że przeniesienie odbywa się z magazynu `MG252` do magazynu `MG251` i dotyczy dwóch typów towarów.

Elementy obowiązkowe: program wczytuje dane i sprawdza: zgodność z formatem, czy podane magazyny były wcześniej utworzone, ograniczenia na pojemność całego magazynu, istnienie

towarów do przeniesienia w magazynie źródłowym, zgodność z kategoriami przechowywanych towarów, ograniczenia na łatwopalność i maksymalną pojemność dla danej kategorii w magazynie docelowym.

Elementy nieobowiązkowe: Sprawdzanie zgodności i ograniczeń na maksymalną pojemność dla podkategorii towarów.

Komenda print

Komenda `print` wymaga jedynie parametrów: podstawowej nazwy pliku (podobnie jak w poprzednich etapach), do którego ma być zapisany wynik komendy oraz opcji dodatkowych.

```
print <file> <additional options>
```

Nazwa pliku powinna składać się z 10 znaków: małych liter alfabetu łacińskiego i cyfr zaczynających się od litery.

Opcje dodatkowe dzielimy na opcje sortowania, filtrowania i dodatkowych elementów wydruku. Kolejność ich podanie nie może mieć wpływu na działanie programu. Szczegółowy opis opcji znajduje się poniżej.

W trybie `txt`, dla każdej komendy `print` program generuje plik z nazwą taką jak podano w scenariuszu uruchomienia i doklejonym rozszerzeniem `txt`. W wygenerowanym pliku znajduje się opis stanu magazynów. Dla każdego magazynu najpierw jego kod, a następnie, w kolejnych liniach dla każdego towaru, który ma stan większy niż zero generowany jest wiersz z kodem towaru, liczbą sztuk w magazynie jego nazwą, kategorią i poziomem łatwopalności.

W trybie `pdf` podobnie, tylko plik ma doklejone rozszerzenie `pdf` a w pliku znajdują się tabele z opisem stanu każdego magazynu. W każdym wierszu tabeli, w pierwszej kolumnie znajduje się kod towaru, w drugiej liczba sztuk, a w trzeciej nazwa. Cały dokument podzielony jest na rozdziały (ang. section). Każdy rozdział zawiera tabelę zawierającą stan magazynu, tytułem sekcji jest jego kod.

Dodatkowe elementy wydruku występują tylko w trybie `pdf`. Natomiast opcje sortowania i filtrowania dotyczą obu trybów pracy.

Punktacja

Poniżej znajduje się dokładniejszy opis możliwych rozszerzeń i opcji dodatkowych dla komendy `print` wraz z punktacją.

Elementy nieobowiązkowe (maks. 20 pkt)

Poniższe elementy nie są obowiązkowe, to znaczy można zaliczyć projekt bez ich wykonania. Za każdy podpunkt prawidłowo zaimplementowany można otrzymać dodatkowe punkty, łącznie nie więcej niż 20 pkt.

- RD (+1 pkt) Projekt zawiera plik `README` z opisem programu, sposobem jego użycia i dostępnych opcji.
- L1 (+1 pkt) Projekt korzysta ze stworzonej biblioteki innymi słowy część utworzonego kodu to biblioteka (ang. library) skonfigurowana w systemie `cmake`.
- L2 (+1 pkt) Projekt korzysta przynajmniej z dwóch stworzonych bibliotek, w tym przynajmniej jedna to biblioteka statyczna i przynajmniej jedna dynamiczna (ang. shared).
- TC (+1 pkt) W trybie `pdf` wygenerowany dokument zawiera spis treści. Sugestia rozwiązania: polecenie `tableofcontents` w systemie `LATEX`.
- TP (+2 pkt) W trybie `pdf` na pierwszej stronie wygenerowanego dokumentu znajdują się: tytuł, data utworzenia dokumentu, wersja programu użytego do jego wygenerowania wraz kodem obliczonym na podstawie kodu wykonywalnego programu i jeszcze dwa inne elementy według własnej inwencji, w tym jeden graficzny.
- Sugerowane narzędzie: `sha1` lub inna funkcja skrótu (ang. hash function).
- CA (+2 pkt) Stan każdego magazynu prezentowanego w trybie `pdf` opatrzony jest podpisem/tytułem (ang. caption) pod tabelą. Tytuł powinien zawierać nazwę magazynu podaną w komendzie `create`.
- LT (+2 pkt) Program prawidłowo prezentuje stany magazynów zawierających wiele towarów, których lista nie mieści się na jednej stronie. Sugestia rozwiązania: pakiet `longtable` w systemie `LATEX`.
- SM (+4 pkt) Dokument zawiera rozdział z podsumowaniem, w którym zaprezentowane są statystyki dotyczące wszystkich magazynów, w tym: liczbę magazynów, wraz z ich parametrami, liczbę sztuk towarów łącznie, kompletną listę towarów i liczbę sztuk we wszystkich magazynach łącznie.
- Towary zaprezentowane są w taki sposób, że każda kategoria towarów umieszczona jest w osobnej tabeli.
- SC (+2 pkt) Dla komendy `print` dostępna jest opcja `-SC`, która powoduje posortowanie magazynów według ich kodów. Podobnie towary w obrębie każdej tabeli również są posortowane według kodów.
- Należy użyć bibliotecznej funkcji sortującej, np. `qsort` i funkcji do porównywania napisów, np: `strcmp` – nie należy implementować sortowania, (ta uwaga dotyczy też kolejnego punktu).
- SV (+2 pkt) Dla komendy `print` dostępna jest opcja sortowania `-SV`, która powoduje posortowanie magazynów według liczby towarów w nich przechowywanych. Podobnie towary w magazynach wypisywane są w kolejności zaczynając od tych, których jest najwięcej. Jeśli dostępne są obie opcje: `-SC` i `-SV`, to nie mogą one być użyte jednocześnie. Zastosowanie opcji sortowania powinno mieć wpływ na podsumowanie (o ile zostało zaimplementowane).

PF (+2 pkt) Dla komendy `print` dostępna jest opcja `-PFL<n>`, która spowoduje, że w trybie `pdf` wszystkie towary z kategorią łatwopalności n lub wyższą zostaną wyróżnione w dokumencie innym kolorem.

Elementy ponadobowiązkowe

EC (+2 pkt) Program prawidłowo obsługuje podkategorie. To znaczy określenie podkategorii towaru ma skutek i odpowiednie warunki są sprawdzane w komendach: `create`, `update` i `transfer`.

EF (+2 pkt) Dla komendy `print` dostępna jest opcja filtrowania `-FL`, która powoduje, że wypisywane są stany tylko tych magazynów, w których stan przynajmniej jednej (pod)kategorii towarów jest poniżej oczekiwanego minimum. Podobnie dla każdego magazynu wypisywane są tylko te towary, które należą do takich kategorii. Zastosowanie tej opcji powinno mieć wpływ na tytuł dokumentu i podsumowanie (o ile zostały zaimplementowane). Dla tej opcji wymagane jest obsługiwanie podkategorii.

EE (+2 pkt) Dla komendy `print` dostępna jest opcja filtrowania `-F[<cat1>, ...]`, która powoduje, że wyświetlane są tylko towary z określonych (pod)kategorii podanych na liście. Na przykład podanie parametru `-F[1 2.2 3 4.1 4.2]`. Spowoduje, że wyświetlone zostaną tylko towary z kategorii 1 i 3 oraz z podkategorii 2.2, 4.1 i 4.2. Dla tej opcji wymagane jest obsługiwanie podkategorii.