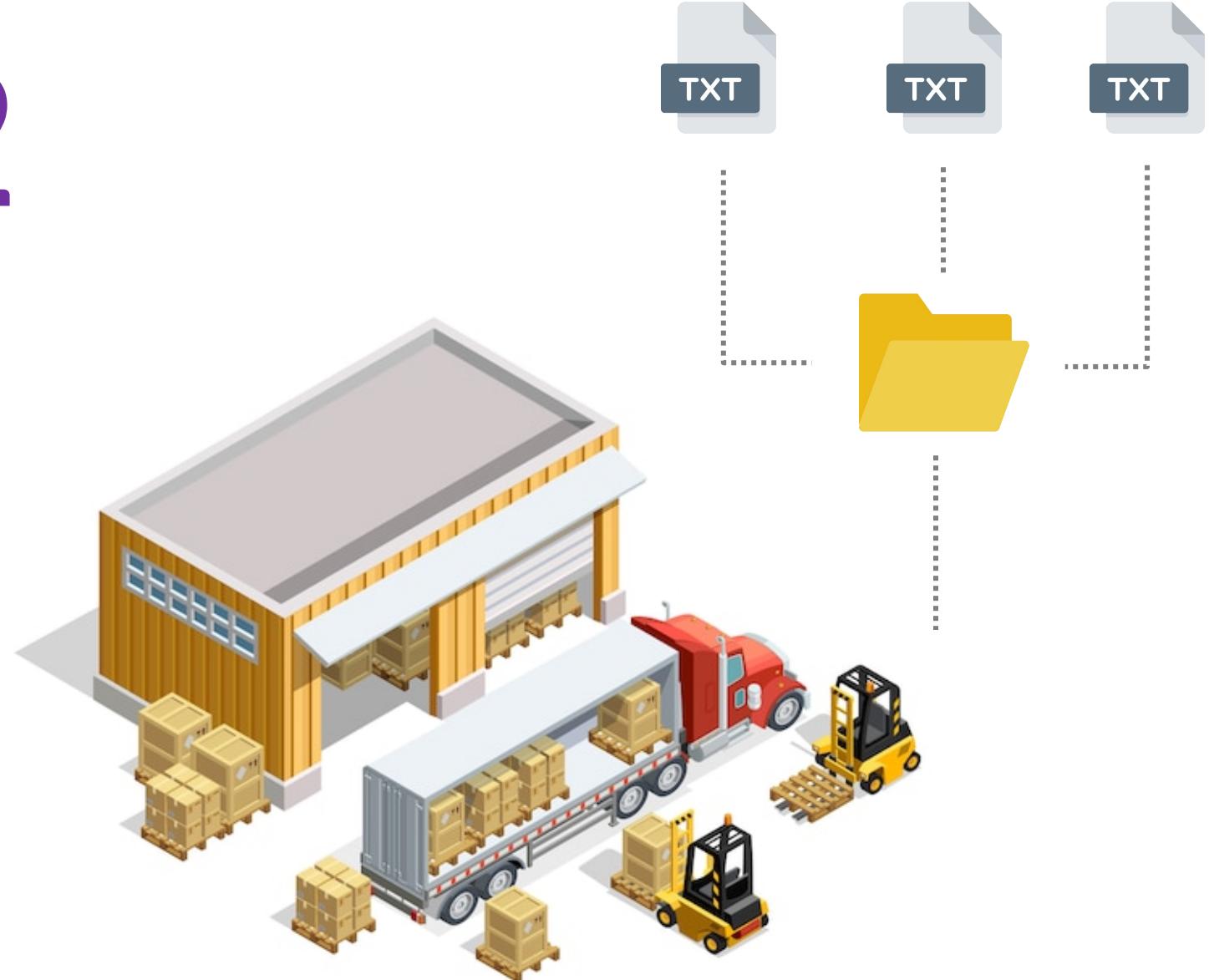


PROG2

Zadanie 5

Pavol Marák

19. 4. 2022



Obsah 1/2

- Podmienky
- Úloha
- Projekt
- Textové súbory Nové
- Štruktúry
- CMD argumenty
- Chybové situácie

Obsah 2/2

- Vstup
- Fázy činnosti programu
- Testovacie scenáre
- Live ukážka

Podmienky

- **Deadline:** 29. apr 2022, 23:59:59
- 10 bodov

Odvzdávanie

- www.prog2.dev
- kontrola anti-plagiátorským systémom

Penalizácia pri odovzdávaní zadania 5

1. pokus max. 10 b
2. pokus max. 9 b
3. pokus max. 7 b
4. pokus max. 5 b
5. pokus max. 3 b
6. pokus max. 1 b

Zadanie 5

- Nadväzuje na zadanie 4.
- Napíšte konzolovú aplikáciu v jazyku C na získavanie informácií o tovaroch a skladoch, ktoré sú uložené **v textových súboroch**.



Zadanie 5

- Aplikácia bude riadiť svoju činnosť pomocou CMD prepínačov.
- Informácie o skladoch sa načítajú z hlavného textového súboru.
- Informácie o tovaroch na jednotlivých skladoch sa načítajú z pomocných textových súborov v definovanom priečinku.



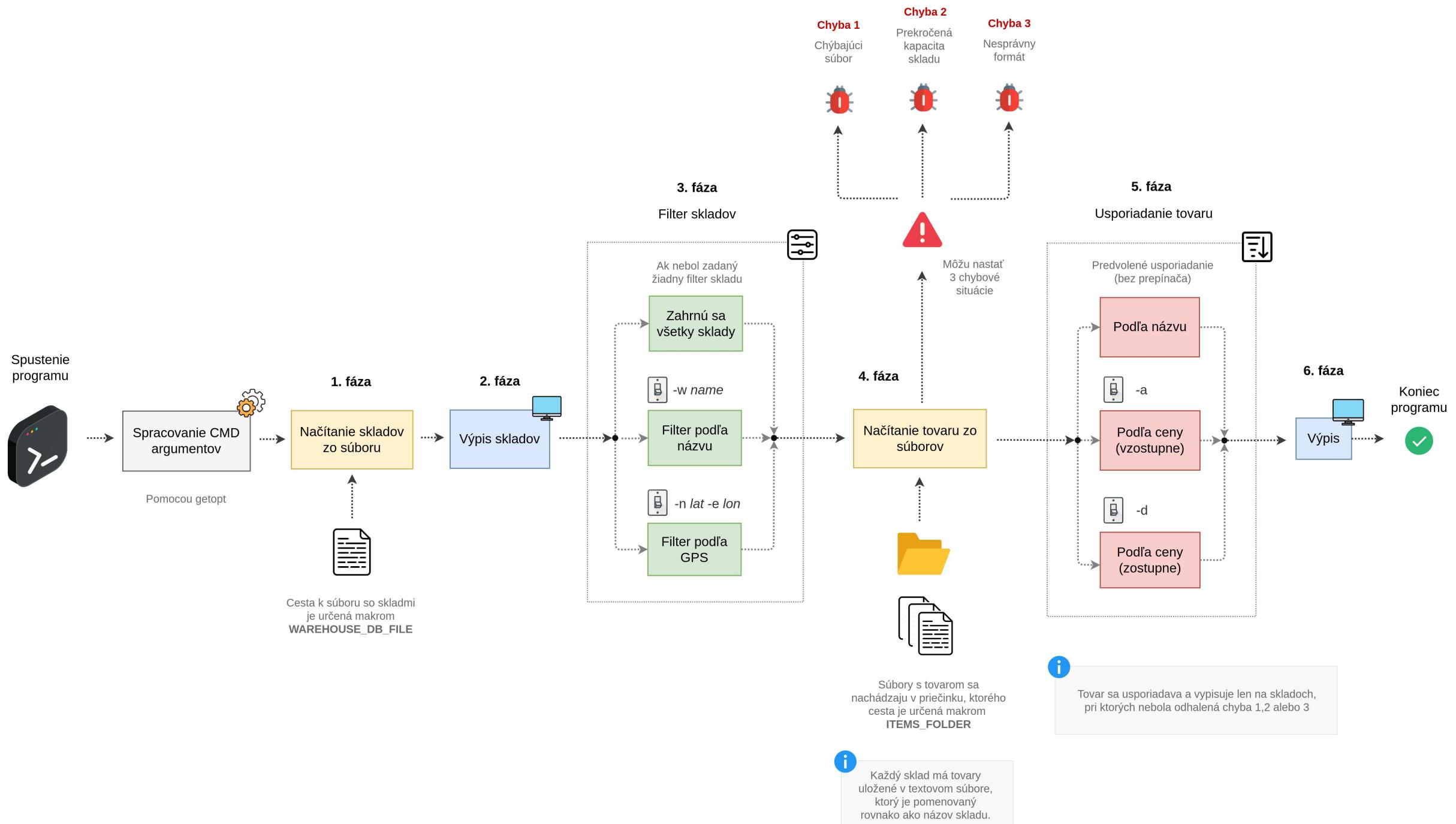
Zadanie 5

- Aplikácia funguje v 6 fázach v pevne danom poradí.
- Používateľ pomocou CMD prepínačov ovláda spôsob vykonania 2 fáz.
- Počas behu programu sa môžu vyskytnúť chybové situácie týkajúce sa načítavania súborov.
- Po odhalení chybovej situácie aplikácia vypíše hlásenie na *stderr*.
- Zmyslom zadania je precvičiť si prácu s dynamickou pamäťou a súbormi.

Úloha

Fázy činnosti aplikácie

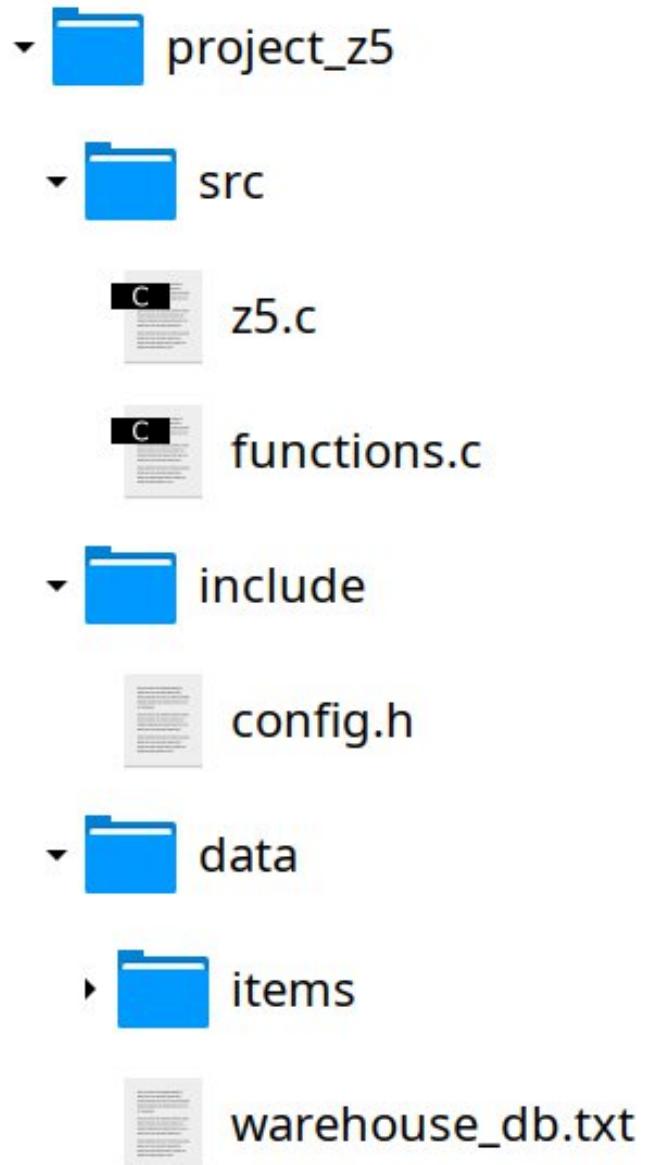
1. Načítanie skladov z hlavného textového súboru.
2. Výpis načítaných skladov.
3. Filter skladov.
4. Načítanie tovaru zo súborov.
5. Usporiadanie tovaru.
6. Výpis.



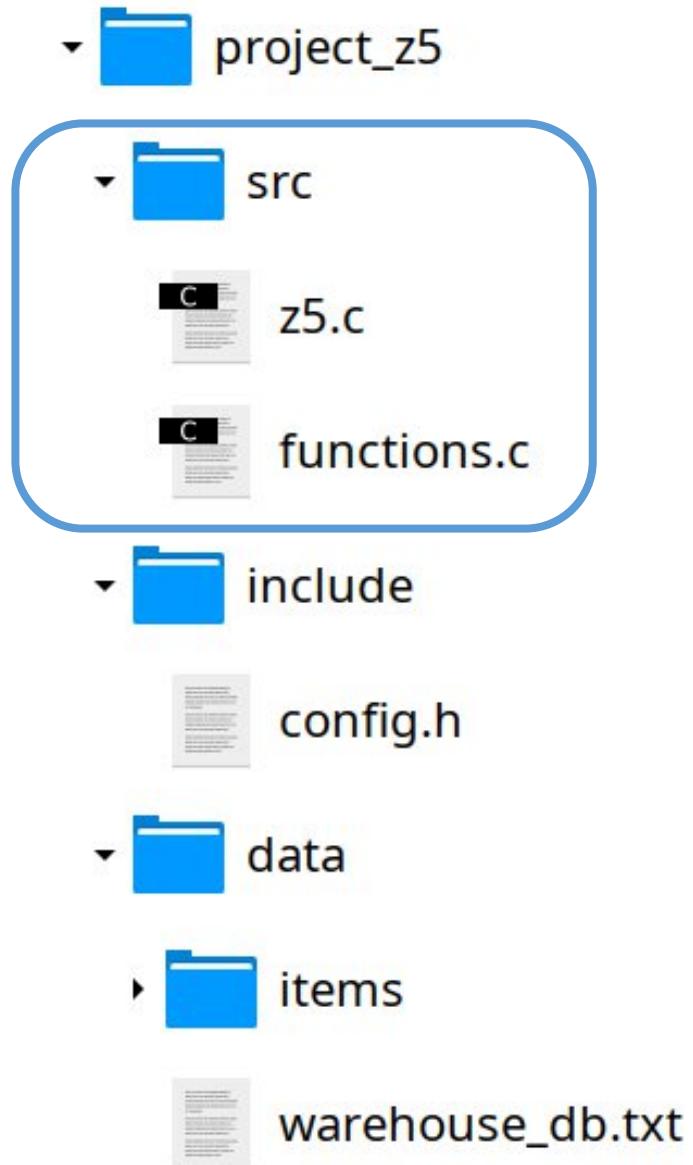
Projekt

- Projekt pozostáva z **3 súborov** (.c a .h).
- Na úspešné skompilovanie si treba uvedené súbory vložiť do projektu vo vývojovom prostredí (odporúčame CLion).
- Projekt obsahuje adresár ***data***, v ktorom sú uložené textové súbory.

Projekt

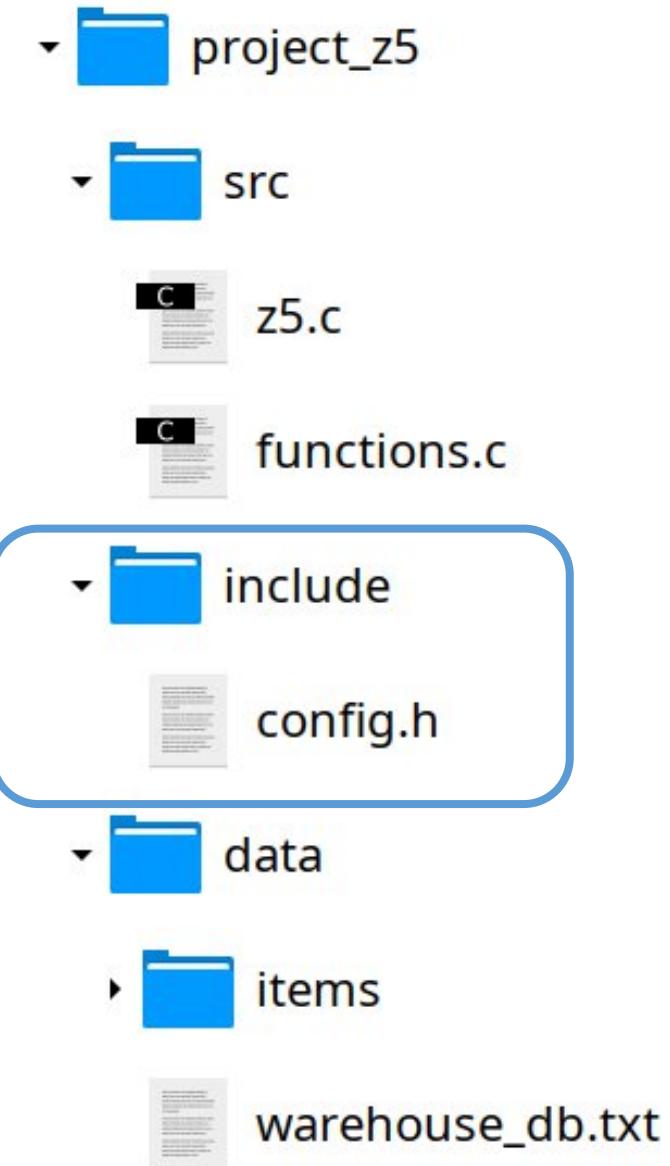


Projekt



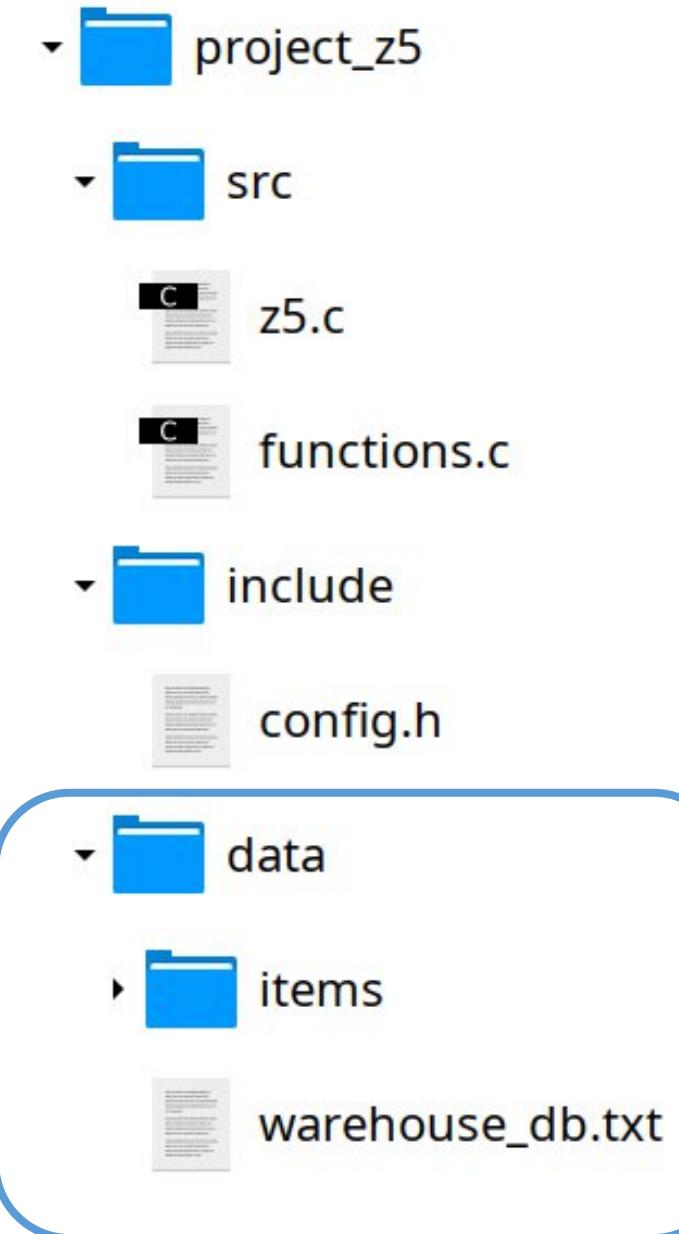
Zdrojové súbory

Projekt



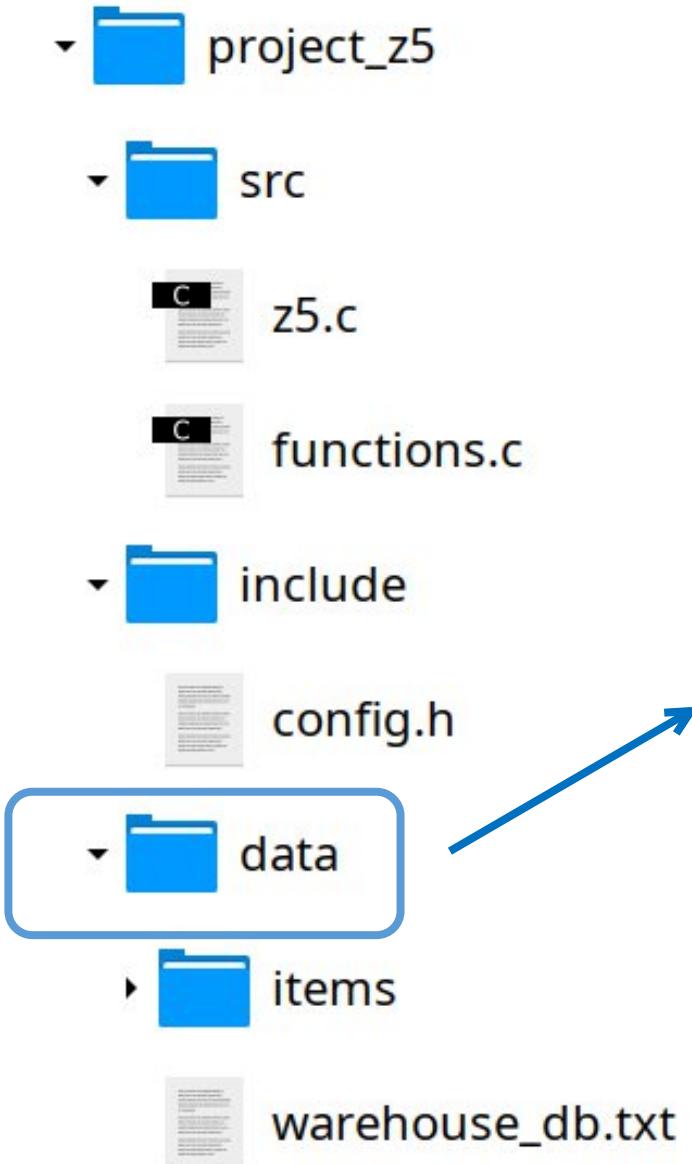
Hlavičkový súbor

Projekt



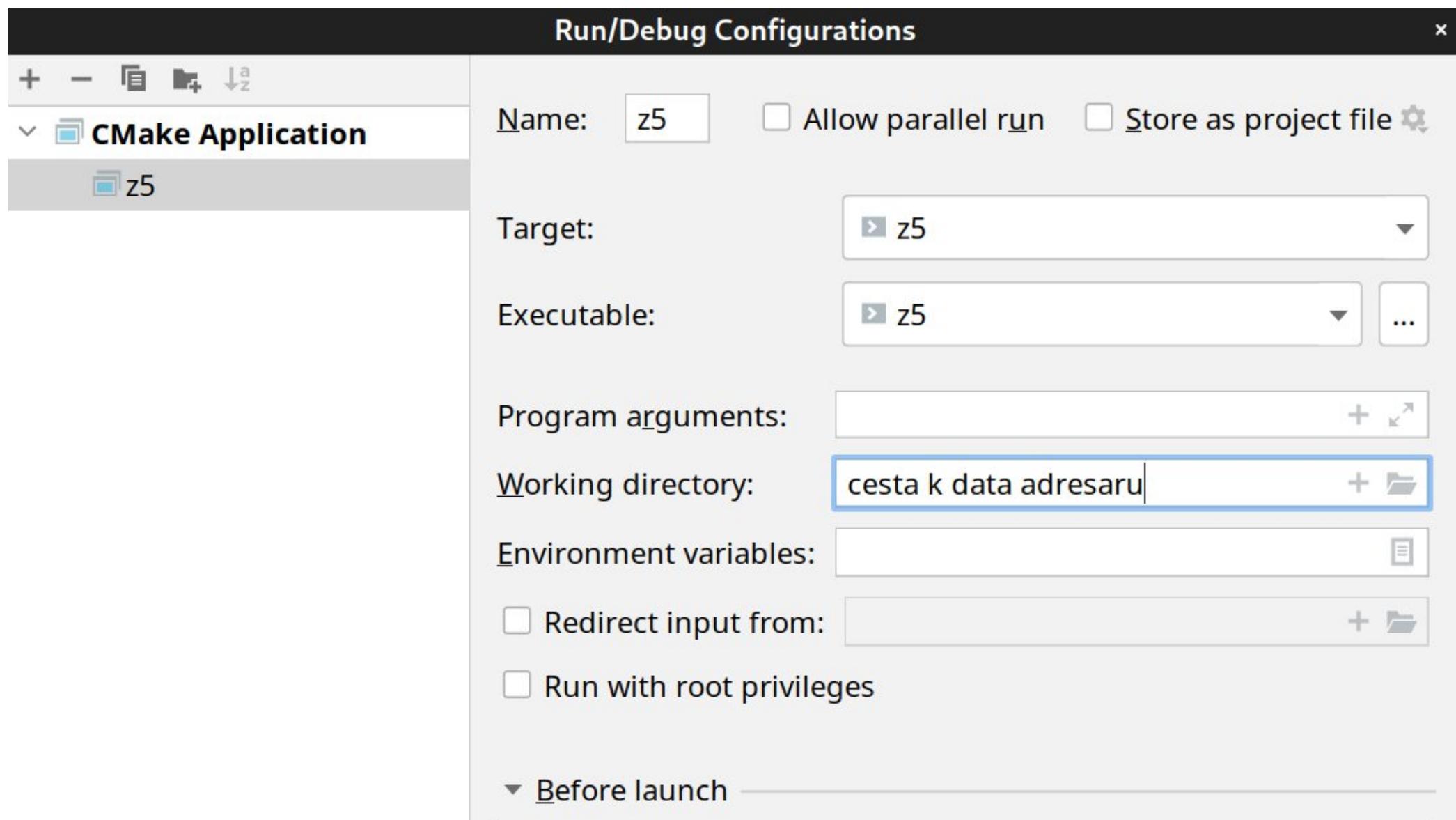
Databáza skladov a tovarov

Projekt



Adresár *data* treba nastaviť
ako working directory vo
vašom vývojovom prostredí.

Nastavanie working directory v CLione



config.h

Hlavičkový súbor, ktorý obsahuje:

- makrá
- konštanty
- definície dátových typov (štruktúry)
- deklarácia pomocnej funkcie
- ~~deklaráciu databázy skladov~~

config.h

Hlavičkový súbor, ktorý obsahuje:

- makrá
- konštanty
- definície dátových typov (štruktúry)
- deklarácia pomocnej funkcie
- ~~deklaráciu databázy skladov~~



Databáza skladov a tovaru je v textových súboroch.

config.h

```
// MAKRA
#define WAREHOUSE_DB_FILE "warehouse_db.txt"
#define ITEMS_FOLDER "items"
#define MAX_NAME 100 // max. dlzka pre nazvy
#define RADIUS_EARTH_KM 6372 // polomer zeme
#define deg2rad(X) ((X)*M_PI/180) // makro na konverziu stupnov na radiany
```

config.h

```
// MAKRA
#define WAREHOUSE_DB_FILE "warehouse_db.txt"
#define ITEMS_FOLDER "items"
#define MAX_NAME 100 // max. dlzka pre nazvy
#define RADIUS_EARTH_KM 6372 // polomer zeme
#define deg2rad(X) ((X)*M_PI/180) // makro na konverziu stupnov na radiany
```

Cesta k hlavnému súboru
so zoznamom skladov.

config.h

```
// MAKRA
#define WAREHOUSE_DB_FILE "warehouse_db.txt"
#define ITEMS_FOLDER "items"
#define MAX_NAME 100 // max. dlzka pre nazvy
#define RADIUS_EARTH_KM 6372 // polomer zeme
#define deg2rad(X) ((X)*M_PI/180) // makro na konverziu stupnov na radiany
```

Cesta k adresáru, ktorý obsahuje textové súbory so zoznamom tovaru pre jednotlivé sklady.

config.h

```
// SEPARATOR ADRESAROV
#ifndef _WIN32
#define PATH_SEPARATOR "\\"
#else
#define PATH_SEPARATOR "/"
#endif
```

config.h

```
// GPS poloha
typedef struct gps {
    double lat; // latitude
    double lon; // longitude
} GPS;

// polozka na sklade (tovar)
typedef struct item {
    char name[MAX_NAME + 1]; // nazov
    int price; // cena
} ITEM;

// sklad
typedef struct warehouse {
    char name[MAX_NAME + 1]; // nazov
    GPS gps; // poloha
    int capacity; // kapacita skladu
    int n; // aktualny pocet poloziiek na sklage
    ITEM *items; // databaza poloziiek na sklage (dynamicke pole)
} WAREHOUSE;
```

config.h

```
// DEKLARACIE FUNKCII  
double distance(GPS gps1, GPS gps2);
```

functions.c

```
#include "config.h"

// Funkcia na vypocet vzdialnosti dvoch GPS suradnic.
double distance(const GPS gps1, const GPS gps2) {
    double d, z, u, v;

    // konverzia na radiany
    double phi1 = deg2rad(gps1.lat);
    double phi2 = deg2rad(gps2.lat);
    double lambda1 = deg2rad(gps1.lon);
    double lambda2 = deg2rad(gps2.lon);

    // predvypocet sinusov, ktore sa potom umocnia na druhu
    u = sin((phi2 - phi1) / 2.0);
    v = sin((lambda2 - lambda1) / 2.0);

    // vypocet vnutornej casti vzorca
    z = sqrt(u * u + cos(phi1) * cos(phi2) * v * v);
    // finalizacia
    d = 2.0 * RADIUS_EARTH_KM * asin(z);

    return d;
}
```

z5.c

```
#include "config.h"
// TODO
// Vase vlastne pomocne makra, datove typy a funkcie.
int main(int argc, char *argv[]){
    // TODO
    // Implementacia hlavnej logiky programu.
    return 0;
}
```

Textové súbory

- Všetky informácie o skladoch a tovare sú uložené v TXT súboroch.
- Databáza TXT súborov je k dispozícii na stiahnutie z webstránky spolu so zdrojovými/hlavičkovými súbormi (v podpriečinku *data*).

Hlavný súbor so zoznamom skladov

- Obsahuje základné informácie o skladoch.
- Neobsahuje informácie o tovare na skladoch.
- Cesta k súboru je určená makrom `WAREHOUSE_DB_FILE`, ktoré je definované v `config.h`.

```
#define WAREHOUSE_DB_FILE "warehouse_db.txt"
```

Hlavný súbor so zoznamom skladov

Ukážka

Secovce • 48.700 • 21.650 • 4 L_F

Dobsina • 48.821 • 20.367 • 15 L_F

Zlate_Moravce • 48.378 • 18.396 • 16 L_F

Zarnovica • 48.483 • 18.717 • 16 L_F

Hlohovec • 48.431 • 17.803 • 4 L_F

Hlavný súbor so zoznamom skladov

```
Secovce · 48.700 · 21.650 · 4 LF
Dobsina · 48.821 · 20.367 · 15 LF
Zlate_Moravce · 48.378 · 18.396 · 16 LF
Zarnovica · 48.483 · 18.717 · 16 LF
Hlohovec · 48.431 · 17.803 · 4 LF
```



Názov skladu (reťazec bez medzier) s
max. dĺžkou MAX_NAME.

Hlavný súbor so zoznamom skladov

Secovce • 48.700 • 21.650 • 4 L_F

Dobsina • 48.821 • 20.367 • 15 L_F

Zlate_Moravce • 48.378 • 18.396 • 16 L_F

Zarnovica • 48.483 • 18.717 • 16 L_F

Hlohovec • 48.431 • 17.803 • 4 L_F



Zemepisná šírka
(3 desatinné miesta)

Hlavný súbor so zoznamom skladov

Secovce • 48.700 • 21.650 • 4 L_F

Dobsina • 48.821 • 20.367 • 15 L_F

Zlate_Moravce • 48.378 • 18.396 • 16 L_F

Zarnovica • 48.483 • 18.717 • 16 L_F

Hlohovec • 48.431 • 17.803 • 4 L_F



Zemepisná dĺžka
(3 desatinné miesta)

Hlavný súbor so zoznamom skladov

Secovce • 48.700 • 21.650 • 4 L_F

Dobsina • 48.821 • 20.367 • 15 L_F

Zlate_Moravce • 48.378 • 18.396 • 16 L_F

Zarnovica • 48.483 • 18.717 • 16 L_F

Hlohovec • 48.431 • 17.803 • 4 L_F



Kapacita skladu

Hlavný súbor so zoznamom skladov

Secovce • 48.700 • 21.650 • 4 L_F

Dobsina • 48.821 • 20.367 • 15 L_F

Zlate_Moravce • 48.378 • 18.396 • 16 L_F

Zarnovica • 48.483 • 18.717 • 16 L_F

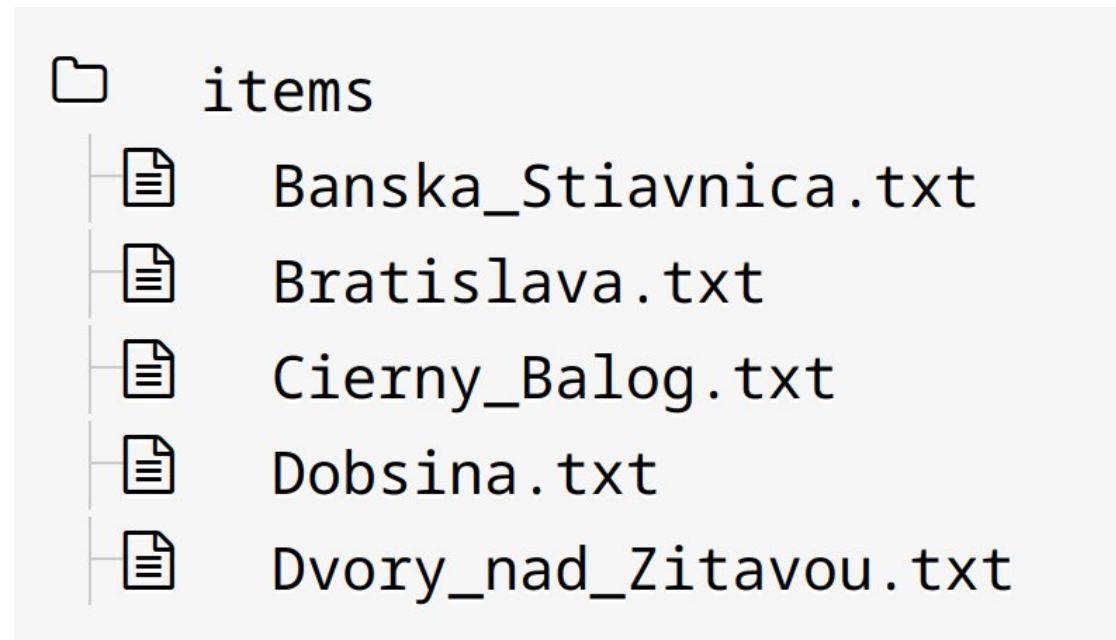
Hlohovec • 48.431 • 17.803 • 4 L_F

Každý riadok končí
znakom \n

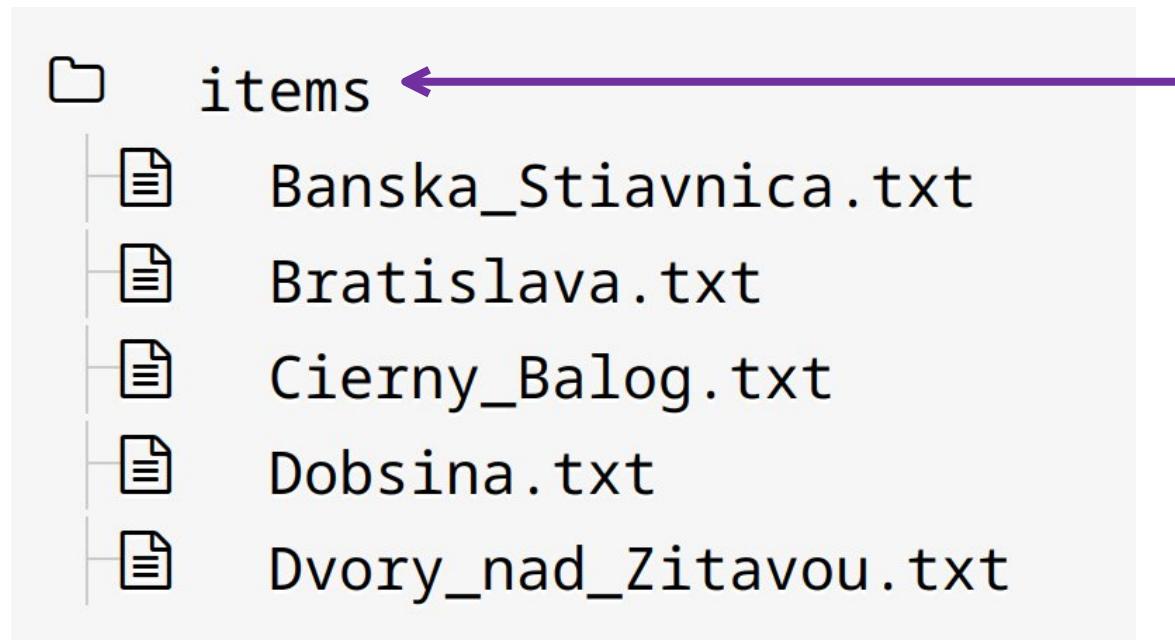
Súbory so zoznamom tovaru

- Sklad má svoj zoznam tovaru uložený v TXT súbore v podpriečinku, ktorého cesta je určená makrom **ITEMS_FOLDER**.
- Súbor so zoznamom tovaru má rovnaký názov ako sklad.
- Napr. zoznam tovaru pre sklad s názvom *Zvolen* sa nachádza v TXT súbore s názvom *Zvolen.txt*, ktorý sa nachádza v adresári s cestou **ITEMS_FOLDER**).

Súbory so zoznamom tovaru



Súbory so zoznamom tovaru



Adresár určený
makrom
ITEMS_FOLDER.

Súbory so zoznamom tovaru

Príklad súboru so
zoznamom tovaru

salad • 10 L_F

pasta • 6 L_F

cake • 17 L_F

melon • 5 L_F

sushi • 89 L_F

Súbory so zoznamom tovaru

```
salad · 10 ℅  
pasta · 6 ℅  
cake · 17 ℅  
melon · 5 ℅  
sushi · 89 ℅
```



Názov tovaru

(reťazec bez medzier s max. dĺžkou MAX_NAME)

Súbory so zoznamom tovaru

salad • 10 ₽

pasta • 6 ₽

cake • 17 ₽

melon • 5 ₽

sushi • 89 ₽



Cena tovaru

Súbory so zoznamom tovaru

salad • 10 ₠

pasta • 6 ₠

cake • 17 ₠

melon • 5 ₠

sushi • 89 ₠ ←

Každý riadok končí znakom \n

Štruktúry

Pomocou štruktúr reprezentujeme:

- GPS súradnice
- Tovar na sklade
- Sklad

GPS súradnice

```
// GPS poloha
typedef struct gps {
    double lat; // latitude
    double lon; // longitude
} GPS;
```



Tovar na sklade

```
// polozka na sklade (tovar)
typedef struct item {
    char name[MAX_NAME + 1]; // nazov
    int price; // cena
} ITEM;
```



Sklad

Nové

```
// sklad
typedef struct warehouse {
    char name[MAX_NAME + 1]; // nazov
    GPS gps; // poloha
    int capacity; // kapacita skladu
    int n; // aktualny pocet položiek na skla
    ITEM *items; // databaza položiek na sklade (dynamické pole)
} WAREHOUSE;
```



Sklad

Nové

Max. počet položiek na sklade

```
// sklad
typedef struct warehouse {
    char name[MAX_NAME + 1]; // nazov
    GPS gps; // pozicia
    int capacity; // kapacita skladu
    int n; // aktualny pocet položiek na skla
    ITEM *items; // databaza položiek na skrade (dynamické pole)
} WAREHOUSE;
```



Sklad

Nové

```
// sklad
typedef struct warehouse {
    char name[MAX_NAME + 1]; // nazov
    GPS gps; // poloha
    int capacity; // kapacita skladu
    int n; // aktualny pocet položiek na skla
    ITEM *items; // databaza položiek na sklade (dynamicke pole)
} WAREHOUSE;
```

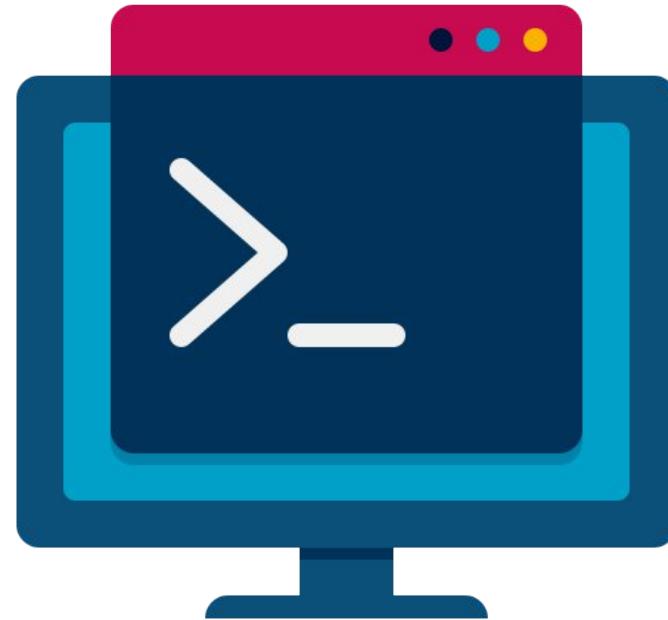


Dynamické pole položiek na sklade.

CMD argumenty

Filter skladov:

1. Predvolený filter.
2. Podľa názvu skladu.
3. Podľa polohy.
4. Podľa názvu tovaru na sklade



CMD argumenty | Filter skladov

Predvolený filter

- Do výsledkov hľadania sa zahrnú všetky skladov.
- Nepoužije sa žiadny prepínač slúžiaci na filtrovanie skladov.

CMD argumenty | Filter skladov

Podľa názvu skladu 

- Prepínač *-w name*

CMD argumenty | Filter skladov

Podľa polohy 

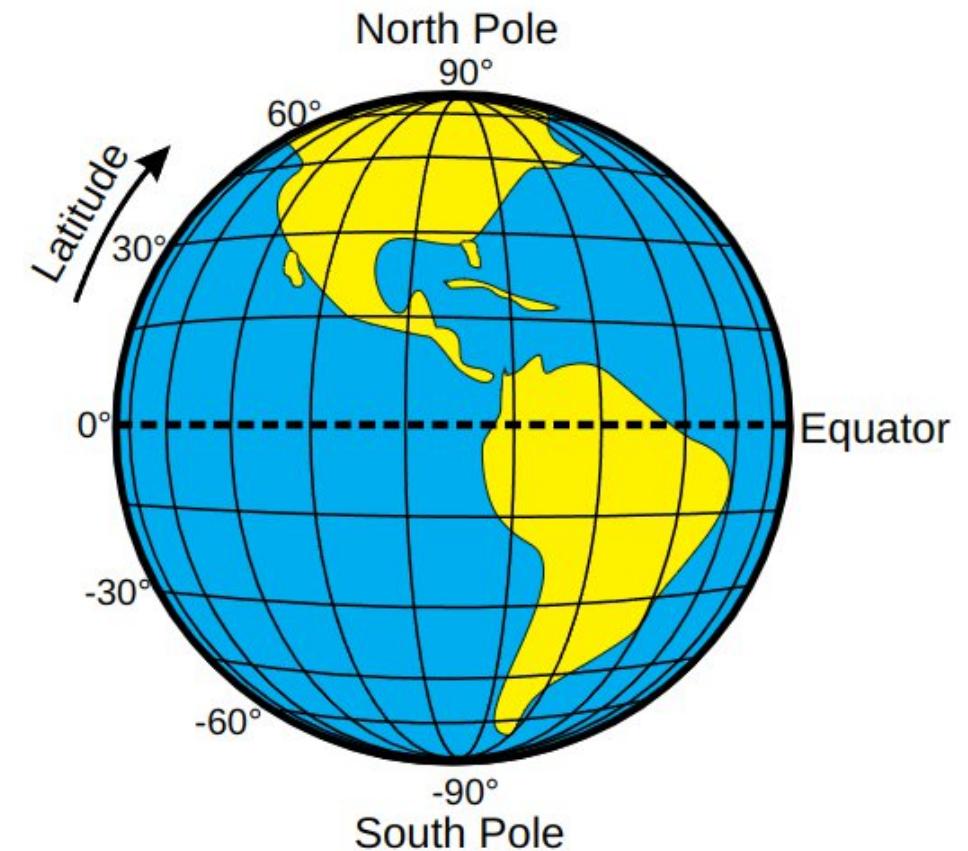
- Prepínače *-n lat -e lon*

CMD argumenty | Filter skladov

Podľa polohy

- Prepínače *-n lat -e lon*

zemepisná šírka
↑
lat

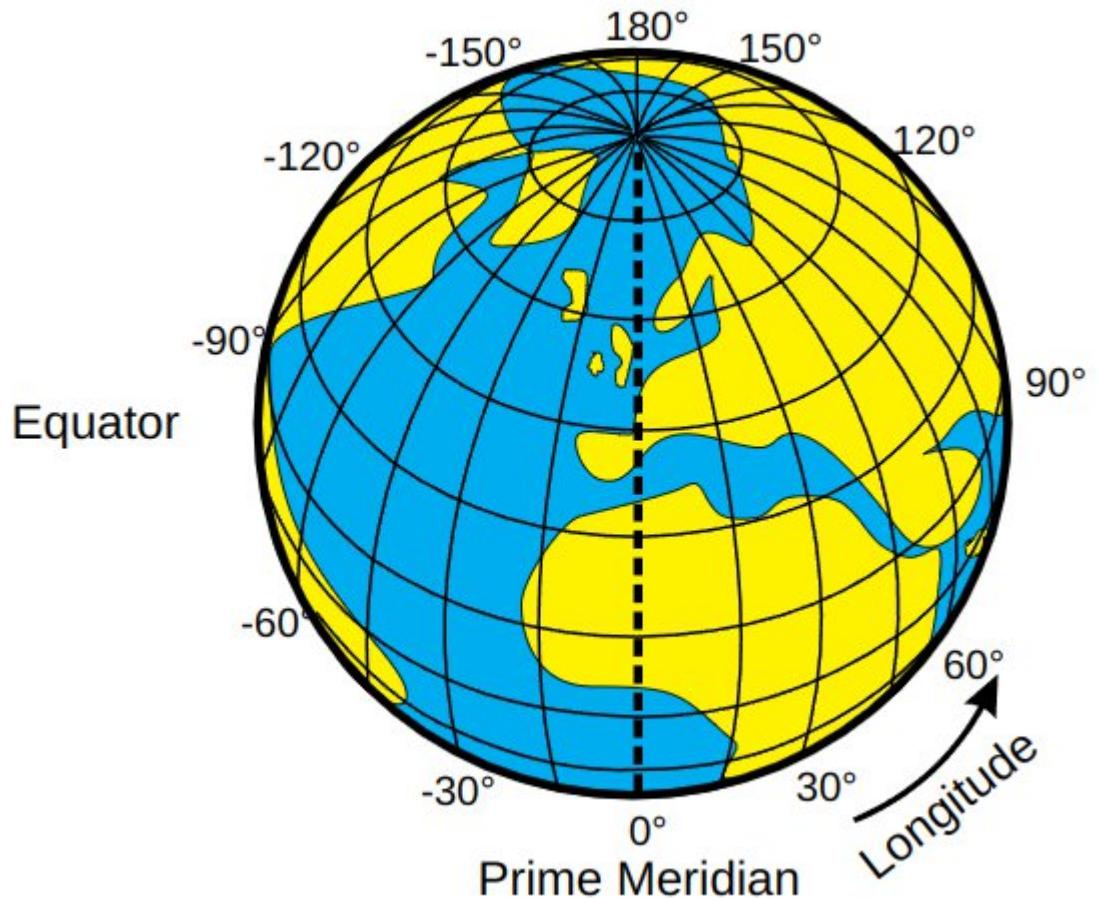


CMD argumenty | Filter skladov

Podľa polohy

- Prepínače *-n lat -e lon*

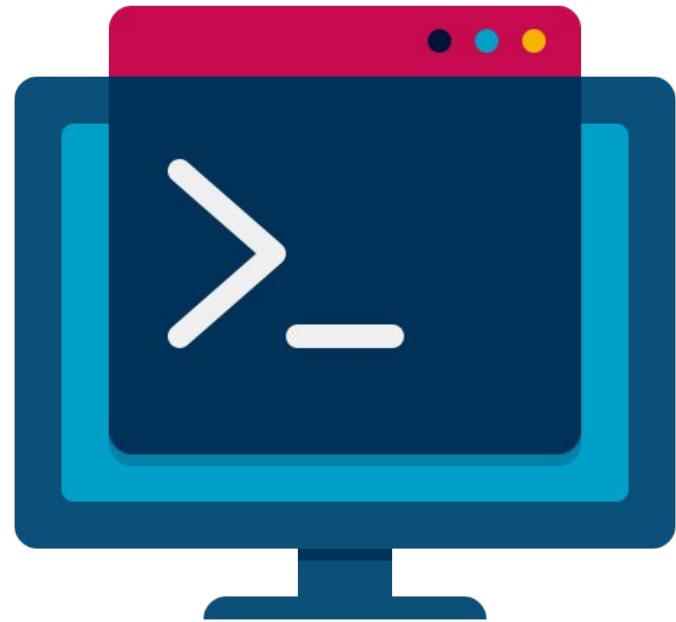
zemepisná dĺžka
↑
lat



CMD argumenty

Usporiadanie tovaru na sklade:

1. Predvolené usporiadanie.
2. Usporiadanie podľa ceny vzostupne.
3. Usporiadanie podľa ceny zostupne.



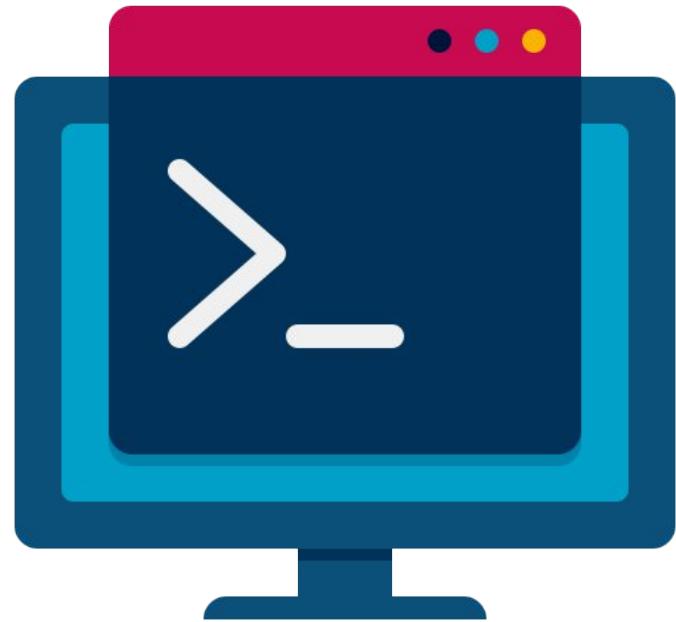
CMD argumenty

Usporiadanie tovaru na sklade:

1. Predvolené usporiadanie.
2. Usporiadanie podľa ceny vzostupne.
3. Usporiadanie podľa ceny zostupne.



Usporiadanie tovaru sa aplikuje len na tie sklady, ktoré vyhovujú filtrovaniu a pri ktorých sa nevyskytla žiadna z chybových situácií.



CMD argumenty | Usporiadanie tovaru

Predvolené usporiadanie

- Tovar na skladoch sa usporiada podľa názvu vzostupne.
- Nepoužije sa žiadny prepínač slúžiaci na usporiadanie tovaru.

CMD argumenty | Usporiadanie tovaru

Predvolené usporiadanie

- Tovar na skladoch sa usporiada podľa názvu vzostupne.
- Nepoužije sa žiadny prepínač slúžiaci na usporiadanie tovaru.

Podľa ceny vzostupne

- Prepínač `-a`

CMD argumenty | Usporiadanie tovaru

Predvolené usporiadanie

- Tovar na skladoch sa usporiada podľa názvu vzostupne.
- Nepoužije sa žiadny prepínač slúžiaci na usporiadanie tovaru.

Podľa ceny vzostupne 

- Prepínač *-a*

Podľa ceny zostupne 

- Prepínač *-d*

Chybové situácie

- V zadaní rozlišujeme 3 chybové situácie súvisiace s načítavaním textových súborov.



Chybové situácie

- V zadaní rozlišujeme 3 chybové situácie súvisiace s načítavaním textových súborov.
- Všetky predpísané hlásenia treba vypísať na štandardný chybový výstup (stderr).



Chybové situácie

- V zadaní rozlišujeme **3 chybové situácie** súvisiace s načítavaním textových súborov.
- Všetky predpísané hlásenia treba vypísať na **štandardný chybový výstup (stderr)**.
- ~~Pri chybovej situácii program skončí bez výstupu s predpísanou návratovou hodnotou.~~



Chybová situácia 1

- Nastane ak pre niektorý sklad neexistuje textový súbor so zoznamom jeho tovaru.



Chybová situácia 1

- Nastane ak pre niektorý sklad neexistuje textový súbor so zoznamom jeho tovaru.
- Program vypíše na *stderr* hlásenie:



```
FILE_ERROR · name.txtL_F
```

Chybová situácia 1

- Nastane ak pre niektorý sklad neexistuje textový súbor so zoznamom jeho tovaru.
- Program vypíše na *stderr* hlásenie:



```
FILE_ERROR · name.txt ←
```

Namiesto *name* sa vypíše názov skladu,
pri ktorom nastale chyba.

Chybová situácia 2

- Nastane ak pri niektorom sklade zistíme, že textový súbor so zoznamom jeho tovaru obsahuje viac položiek ako je kapacita daného skladu.



Chybová situácia 2

- Nastane ak pri niektorom sklade zistíme, že textový súbor so zoznamom jeho tovaru obsahuje viac položiek ako je kapacita daného skladu.
- Program vypíše na *stderr* hlásenie:



```
CAPACITY_ERROR · name . txt ↵
```

Chybová situácia 2

- Nastane ak pri niektorom sklade zistíme, že textový súbor so zoznamom jeho tovaru obsahuje viac položiek ako je kapacita daného skladu.
- Program vypíše na *stderr* hlásenie:



```
CAPACITY_ERROR · name . txt ←
```

Namiesto *name* sa vypíše názov skladu, pri ktorom nastale chyba.

Chybová situácia 3

- Nastane ak pri niektorom sklade zistíme, že textový súbor so zoznamom jeho tovaru obsahuje údaje, ktoré nie sú v požadovanom formáte.



Chybová situácia 3

- Nastane ak pri niektorom sklade zistíme, že textový súbor so zoznamom jeho tovaru obsahuje údaje, ktoré nie sú v požadovanom formáte.
- Program vypíše na *stderr* hlásenie:



```
FORMAT_ERROR · name . txtL_F
```

Chybová situácia 3

- Nastane ak pri niektorom sklade zistíme, že textový súbor so zoznamom jeho tovaru obsahuje údaje, ktoré nie sú v požadovanom formáte.
- Program vypíše na *stderr* hlásenie:



```
FORMAT_ERROR · name . txt ←
```

Namiesto *name* sa vypíše názov skladu,
pri ktorom nastale chyba.

Chybová situácia 3

Príklad správneho
formátu

salad	• 10	L _F
pasta	• 6	L _F
cake	• 17	L _F
melon	• 5	L _F
sushi	• 89	L _F



Chybová situácia 3

Príklad **správneho**
formátu

salad	• 10	L _F
pasta	• 6	L _F
cake	• 17	L _F
melon	• 5	L _F
sushi	• 89	L _F

Príklad **nesprávneho**
formátu

orange	abc
banana	
123	



Chybová situácia 3

Príklad správneho
formátu

salad	• 10	L _F
pasta	• 6	L _F
cake	• 17	L _F
melon	• 5	L _F
sushi	• 89	L _F

Príklad nesprávneho
formátu

orange	abc
banana	
123	



Cena musí byť
numerická hodnota.

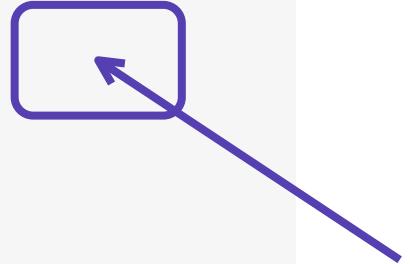
Chybová situácia 3

Príklad **správneho**
formátu

salad • 10 L_F
pasta • 6 L_F
cake • 17 L_F
melon • 5 L_F
sushi • 89 L_F

Príklad **nesprávneho**
formátu

orange abc
banana
123



Chýba cena.



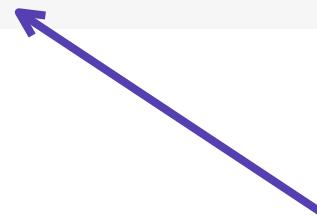
Chybová situácia 3

Príklad **správneho**
formátu

salad • 10^{L_F}
pasta • 6^{L_F}
cake • 17^{L_F}
melon • 5^{L_F}
sushi • 89^{L_F}

Príklad **nesprávneho**
formátu

orange abc
banana
123



Je interpretované ako názov
tovaru, chýba cena.



Vstup

- Program nenačítava z klávesnice žiadne údaje.
- Všetky vstupné informácie sú zadané v termináli ako CMD argumenty a načítané z TXT súborov.



Fázy činnosti programu

1. Načítanie skladov z hlavného textového súboru.
2. Výpis načítaných skladov.
3. Filter skladov.
4. Načítanie tovaru zo súborov.
5. Usporiadanie tovaru.
6. Výpis.

1. Načítanie skladov

- Sklady sa načítajú z TXT súboru, ktorého cesta je určená makrom v *config.h*.
- Treba si vytvoriť a správne alokovať dynamické pole skladov.
- Na čítanie riadkov textového súboru odporúčame využiť niektorú knižničnú funkciu určenú na formátované čítanie ako napr. *fscanf*.

```
#define WAREHOUSE_DB_FILE "warehouse_db.txt"
```

2. Výpis načítaných skladov

- Po načítaní skladov z hlavného textového súboru nasleduje ich výpis na **štandardný výstup** (stdout).
- Sklady sa vypíšu v rovnakom formáte, v akom sú uložené v súbore.

3. Filter skladov

Po načítaní a výpise skladov nasleduje ich filtrovanie:

- Predvolený filter.
- Podľa názvu skladu (*-w name*).
- Podľa polohy (*-n lat -e lon*).

4. Načítanie tovaru zo súborov

- Po prefiltrovaní skladov sa načítajú tovary na zostávajúcich skladoch.
- Sklad má svoj zoznam tovaru uložený v TXT súbore, ktorého názov sa zhoduje s názvom skladu.
- Všetky TXT súbory so zoznamom tovaru sú v adresári s cestou určenou makrom v *config.h*.

```
#define ITEMS_FOLDER "items"
```

4. Načítanie tovaru zo súborov

- Každý sklad má atribút *items*, ktorý predstavuje dynamické pole tovarov.
- Pre toto pole treba alokovať dostatok pamäte na uchovanie všetkých položiek v zozname tovaru.
- Po načítaní tovarov je potrebné nastaviť atribút skladu *n* na hodnotu, ktorá predstavuje počet načítaných položiek tovaru.

4. Načítanie tovaru zo súborov

- V procese načítavania tovarov pre sklad môžu nastať **3 typy chybových situácií**, ktoré sú opísané vyššie.
- Po odhalení chybovej situácie program vypíše predpísané chybové hlásenie na ***stderr*** a pokračuje v činnosti.
- Sklad, pri ktorom nastala niektorá z chýb je v ďalších fázach programu ignorovaný, t.j. neusporiadava sa jeho tovar ani sa nezúčastňuje koncového výpisu.

5. Usporiadanie tovaru

- Po načítaní tovarov v jednotlivých skladoch, ktoré vyhovujú filtrovaniu skladov nasleduje usporiadanie položiek tovaru.
- Rozlišujeme tieto typy usporiadania:
 - Predvolené usporiadanie tovaru (podľa názvu tovaru vzostupne).
 - Podľa ceny vzostupne (**-a**).
 - Podľa ceny zostupne (**-d**).

5. Usporiadanie tovaru

- Na usporiadanie je potrebné napísať si vlastnú funkciu.
- Algoritmus usporiadania tovaru musí byť **stabilný**, t.j. relatívne poradie rovnakých položiek musí zostať po usporiadaní nezmenené.
- Nedá sa použiť knižničná funkcia *qsort* (negarantuje stabilitu).
- Inšpirovať sa môžete napr. BubbleSort algoritmom.

5. Usporiadanie tovaru

Príklad stabilného usporiadania podľa názvu vzostupne.

filet 2

eggs 14

ketchup 2

eggs 3

curd 98

Pred

curd 98

eggs 14

eggs 3

filet 2

ketchup 2

Po

5. Usporiadanie tovaru

Príklad stabilného usporiadania podľa ceny vzostupne.

filet 2
eggs 14
ketchup 2
eggs 3
curd 98

Pred

filet 2
ketchup 2
eggs 3
eggs 14
curd 98

Po

5. Usporiadanie tovaru

Príklad stabilného usporiadania podľa ceny zostupne.

filet 2
eggs 14
ketchup 2
eggs 3
curd 98

curd 98
eggs 14
eggs 3
filet 2
ketchup 2

Pred

Po

6. Výpis

- Program počas svojej činnosti produkuje dva druhy výpisov:
 - *stdout* (bežné výpisy)
 - *stderr* (chybové hlásenia)

stdout

- Výpis na *stdout* sa skladá z 2 častí:
 1. Výpis načítaných skladov.
 2. Výpis usporiadанého tovaru na zostávajúcich skladoch.

Príklad stdout výpisu

Databáza s
3 náhodnými skladmi.

```
Secovce · 48.700 · 21.650 · 4 L_F
Poprad · 49.050 · 20.300 · 11 L_F
Dvory_nad_Zitavou · 47.993 · 18.264 · 7 L_F
Secovce · 48.700 · 21.650 · 4 · : L_F
1. · biscuit · 97 L_F
2. · muffin · 55 L_F
3. · poutine · 12 L_F
Poprad · 49.050 · 20.300 · 11 · : L_F
1. · batter · 78 L_F
2. · coffee · 27 L_F
3. · queso · 21 L_F
4. · skillet · 63 L_F
5. · yeast · 5 L_F
Dvory_nad_Zitavou · 47.993 · 18.264 · 7 · : L_F
1. · aroma · 98 L_F
2. · butter · 75 L_F
3. · drink · 87 L_F
4. · enchilada · 28 L_F
5. · skillet · 31 L_F
6. · tartar · 19 L_F
7. · taste · 87 L_F
```

Príklad stdout výpisu

Databáza s
3 náhodnými skladmi.

Výpis načítaných skladov

Secovce · 48.700 · 21.650 · 4 L_F

Poprad · 49.050 · 20.300 · 11 L_F

Dvory_nad_Zitavou · 47.993 · 18.264 · 7 L_F

Secovce · 48.700 · 21.650 · 4 · : L_F

1. · biscuit · 97 L_F

2. · muffin · 55 L_F

3. · poutine · 12 L_F

Poprad · 49.050 · 20.300 · 11 · : L_F

1. · batter · 78 L_F

2. · coffee · 27 L_F

3. · queso · 21 L_F

4. · skillet · 63 L_F

5. · yeast · 5 L_F

Dvory_nad_Zitavou · 47.993 · 18.264 · 7 · : L_F

1. · aroma · 98 L_F

2. · butter · 75 L_F

3. · drink · 87 L_F

4. · enchilada · 28 L_F

5. · skillet · 31 L_F

6. · tartar · 19 L_F

7. · taste · 87 L_F

Príklad stdout výpisu

Databáza s
3 náhodnými skladmi.

Výpis usporiadaného tovaru
na zostávajúcich skladoch

Príklad s predvoleným
usporiadaním

Secovce · 48.700 · 21.650 · 4 L_F

Poprad · 49.050 · 20.300 · 11 L_F

Dvory_nad_Zitavou · 47.993 · 18.264 · 7 L_F

Secovce · 48.700 · 21.650 · 4 · : L_F

1. · biscuit · 97 L_F

2. · muffin · 55 L_F

3. · poutine · 12 L_F

Poprad · 49.050 · 20.300 · 11 · : L_F

1. · batter · 78 L_F

2. · coffee · 27 L_F

3. · queso · 21 L_F

4. · skillet · 63 L_F

5. · yeast · 5 L_F

Dvory_nad_Zitavou · 47.993 · 18.264 · 7 · : L_F

1. · aroma · 98 L_F

2. · butter · 75 L_F

3. · drink · 87 L_F

4. · enchilada · 28 L_F

5. · skillet · 31 L_F

6. · tartar · 19 L_F

7. · taste · 87 L_F

stderr

- Na *stderr* vypisujeme len chybové hlásenia v prípade chybových situácií 1-3.

Príklad stderr výpisu

```
FORMAT_ERROR Zlate_Moravce.txt
CAPACITY_ERROR Zarnovica.txt
FILE_ERROR Hlohovec.txt
FILE_ERROR Sala.txt
FORMAT_ERROR Velky_Meder.txt
FILE_ERROR Komarno.txt
CAPACITY_ERROR Oscadnica.txt
FILE_ERROR Bardejov.txt
FILE_ERROR Krupina.txt
FILE_ERROR Chorvatsky_Grob.txt
FILE_ERROR Snina.txt
CAPACITY_ERROR Piestany.txt
CAPACITY_ERROR Ruzomberok.txt
CAPACITY_ERROR Velke_Kapusany.txt|
CAPACITY_ERROR Surany.txt
```

Príklad stderr výpisu

V CLione sa *stderr*
výstup zobrazuje
červeným písmom.

```
FORMAT_ERROR Zlate_Moravce.txt
CAPACITY_ERROR Zarnovica.txt
FILE_ERROR Hlohovec.txt
FILE_ERROR Sala.txt
FORMAT_ERROR Velky_Meder.txt
FILE_ERROR Komarno.txt
CAPACITY_ERROR Oscadnica.txt
FILE_ERROR Bardejov.txt
FILE_ERROR Krupina.txt
FILE_ERROR Chorvatsky_Grob.txt
FILE_ERROR Snina.txt
CAPACITY_ERROR Piestany.txt
CAPACITY_ERROR Ruzomberok.txt
CAPACITY_ERROR Velke_Kapusany.txt|
CAPACITY_ERROR Surany.txt
```

Testovacie scenáre

Testovanie chybových situácií
(kontroluje sa len *stderr*).

Testovacie scenáre		
Scenár 1	Chybová situácia 1	1 b
Scenár 2	Chybová situácia 2	1 b
Scenár 3	Chybová situácia 3	1 b

Testovacie scenáre

Filter **-w** + usporiadanie tovaru.

Scenár 4	Prepínač -w	1 b
Scenár 5	Kombinácia prepínačov -w a -a	0.5 b
Scenár 6	Kombinácia prepínačov -w a -d	0.5 b

Testovacie scenáre

Filter `-n lat -e lon` + usporiadanie tovaru.

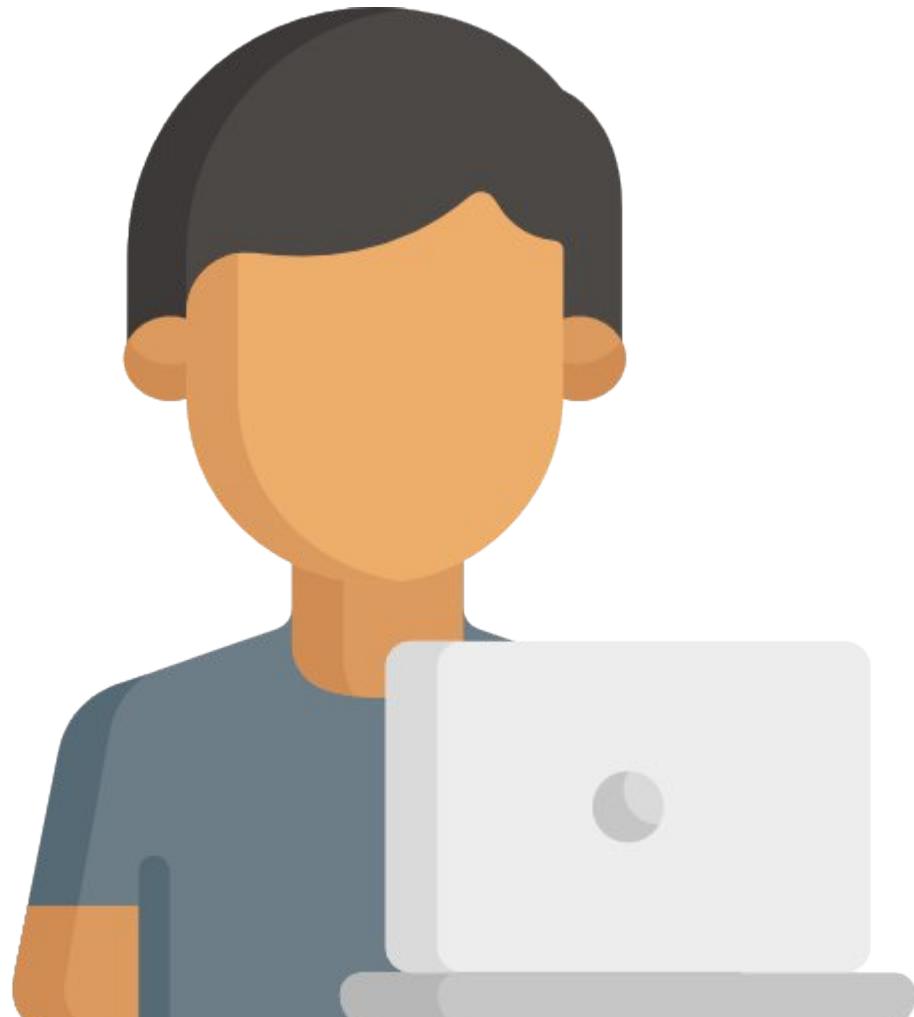
Scenár 7	Prepínače <code>-n</code> a <code>-e</code>	1 b
Scenár 8	Kombinácia prepínačov (<code>-n</code> , <code>-e</code>) a <code>-a</code>	0.5 b
Scenár 9	Kombinácia prepínačov (<code>-n</code> , <code>-e</code>) a <code>-d</code>	0.5 b

Testovacie scenáre

Predvolený filter + usporiadanie tovaru.

Scenár 10	Bez prepínačov	1 b
Scenár 11	Prepínač - a	1 b
Scenár 12	Prepínač - d	1 b
	Súčet	10 b

Live ukázka



Zdroje

- <https://www.flaticon.com/>
- <https://www.freepik.com/>