DOKUMENTACJA

25 listopada 2017

Krzysztof Dąbrowski gr. 1I1

Projekt 3. Środa godz. 12.15 – 14.00

# Treść projektu:

Baza ozdobnych guzików dla kolekcjonera.

# Metoda rozwiązania:

Wykorzystanie dynamicznego alokowana pamięci oraz struktur.

# Specyfikacje:

Aplikacja działa w konsoli zarówno na systemie Windows jak i Linux  
Użytkownik wchodzi w interakcje z programem wpisując do konsoli odpowiednie znaki.

# Struktury:

struct Data – Przechowuje dzień, miesiąc i rok daty

enum Material – Materiały z jakich może być wykonany guzik

struct Guzik – Przechowuje dane o pojedynczym guziku

struct Vector – Tablica dynamiczna do przechowywania rekordów bazy danych

# Funkcje:

char \*dataToString(char slowo[], struct Data data) – Zamienie strukturę data na napis w formacie DD-MM-RRRR

const char \*materialToString(enum Material material) – Zamienia typ Matrial na odpowiadający mu ciąg znaków

char \*guzikToString(char nazwa[], struct Guzik guzik) – Zamienia typ Guzik na ciąg znaków opisujący jego cechy

enum Material stringToMaterial(char materialS[]) – Zamienia ciąg znaków opisujący materiał na typ materiał. W przypadku błędnych danych zwraca -1

char \*wczytaj(char \*z, int ile) – Wczytuje daną liczbę znaków lub do końca linii a potem czyści bufor

void czyscBufor(void) – Czyści bufor wejściowy

struct Guzik wczytajGuzik(void) – Wczytuje guzik ze standardowego wejścia

struct Data wczytajDate(void) – Wczytuje datę ze standardowego wejścia

void wyswietlMenuGlowne() – Wyświetla główne menu

void menueCzyszczenia(struct Vector\* baza) – Wyświetla menu czyszczenia bazy danych

void menuEdycji(struct Vector\* baza) – Wyświetla menu edytowania zawartosci bazy danych

void menuZapisuDoPliku(struct Vector\* baza) – Wyświetla menu zapisu do pliku

void menuWczytajZPliku(struct Vector\* baza) – Wyświetla menu wczytania danych z pliku

void menuSortowania(struct Vector\* baza) – Wyświetla menu sortowania bazy danych

void zapiszDoPliku(struct Vector\* vector, char nazwaPliku[]) – Zapisuje bazę danych do pliku

void wczytajZPliku(struct Vector\* vector, char nazwaPliku[]) – Wczytuje bazę danych z pliku

int comparNameI(const void\* p1, const void\* p2) – Porównanie po imieniu rosnąco

int comparNameD(const void\* p1, const void\* p2) – Porównanie po imieniu malejąco

int comparSizeD(const void\* p1, const void\* p2) – Porównanie po rozmiarze rosnąco

int comparSizeI(const void\* p1, const void\* p2) – Porównanie po rozmiarze malejąco

int comparMeterialI(const void\* p1, const void\* p2) – Porównanie po material rosnąco

int comparMeterialD(const void\* p1, const void\* p2) – Porównanie po materiale malejąco

int comparPriceI(const void\* p1, const void\* p2) – Porównanie po cenie rosnąco

int comparPriceD(const void\* p1, const void\* p2) – Porównanie po cenie malejąco

int comparDateI(const void\* p1, const void\* p2) – Porównanie po dacie wprowadzenia rosnąco

int comparDateD(const void\* p1, const void\* p2) – Porównanie po dacie wprowadzenia malejąco

int comparProductionYearI(const void\* p1, const void\* p2) – Porównanie po dacie produkcji rosnąco

int comparProductionYearD(const void\* p1, const void\* p2) – Porównanie po dacie produkcji malejąco

void testVectorDodawanieElementow(void) – Testuje mechanizm dodawania rekordów do bazy danych

void testWypisaniaDanych(void) – Testuje mechanizm wypisywania rekordów

void testSortowan(void) – Testuje wszystkie sortowania

void zainicjalizuj(struct Vector \*vector) – Przygotowanie vectora do pracy

void push\_back(struct Vector \*vector, struct Guzik guzik) – Dodanie element na koniec vectora

void delete(struct Vector \*vector, int index) – Usuniecie wybranego elementu

void empty(struct Vector \*vector) – Wyczyszczenie vectora

void wypisz(struct Vector \*vector) – Wypisanie zawartości vectora

void cleanupAtExit(struct Vector \*vector) – Zwolnienie całej pamięci używanej przez vector. Po wywołaniu tej funkcji należy go ponownie zainicjalizować!

void sort(struct Vector \*vector, enum TypSortowania typSortowania, bool rosnaco) – Posortowanie zawartości vectora według danego kryterium

# Testowanie:

Program zawiera specjalne funkcje do testowania poprawności działania niektórych funkcji.

# Odstępstwa:

Typów materiału, z których może być zrobiony guzik jest 7 zamiast 5.

Bazę sortować można rosnąco i malejąco.