

Politechnika Śląska w Gliwicach

Wydział Informatyki, Elektroniki i Informatyki

Temat projektu : Kalendarz

autor	Arkadiusz Pilarski
prowadzący	Marek Kokot
rok akademicki	2015/2016
kierunek	informatyka
rodzaj studiów	SSI
semestr	1
termin laboratorium / ćwiczeń	poniedziałek, 10:00 – 11:30
grupa	1
sekcja	2
termin oddania sprawozdania	2015-11-23
data oddania sprawozdania	2015-11-22

1. Treść zadania

Napisać program generujący kalendarz na zadany rok. Numer roku podawany jest po przełączniku -r. Pierwszy dzień tygodnia jest ustawiany po przełączniku -d jako pojedynczy znak: n – niedziela, o – poniedziałek, w – wtorek, s – środa, c – czwartek, t – piątek, b – sobota. Gotowy kalendarz jest zapisywany do pliku wyjściowego, którego nazwa podana jest po przełączniku -o. Przykładowe wywołanie programu: `program.exe -o kal2013 -d o -r 2013` Wskazówka: Dzień tygodnia dla zadanej daty można wyznaczyć algorytmem Gaussa lub Zeller'a.

2. Analiza zadania.

Zagadnienie przedstawia problem generowania kalendarza w oparciu o dane przekazane z konsoli. Gotowy kalendarz zapisany do pliku o pożądanej nazwie powinien wyświetlać dni tygodnia w kolejności ustalonej poprzez parametr przekazany z konsoli.

2.1 Struktury danych

W programie wykorzystano trzy tablice. Dwie tablice typu string: pierwsza zawiera nazwy miesięcy występujących w roku, druga nazwy dni tygodnia. Trzecią tablicą jest tablica typu int, zawierająca liczby reprezentujące liczbę dni danego miesiąca. Układ elementów w tablicy miesięcy i liczb dni nie jest przypadkowy – elementy o tych samych numerach tworzą parę: miesiąc i liczba dni w tym miesiącu.

2.2 Algorytmy

Program generuje kalendarz w oparciu o działanie pętli w pętli, które wykorzystują podane przez konsolę parametry.

3. Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Należy przekazać do programu trzy parametry:

- 1) po przełączniku „-o” nazwę pliku z rozszerzeniem .txt, który zostanie utworzony z wygenerowanym kalendarzem,
 - 2) po przełączniku „-d” dzień tygodnia, który będzie wyświetlany jako pierwszy,
 - 3) po przełączniku „-r” rok, dla którego chcemy utworzyć kalendarz.
- Parametry można podawać w dowolnej kolejności.

UWAGA: Pierwszy dzień tygodnia jest ustawiany po przełączniku -d jako pojedynczy znak:

n – niedziela, o – poniedziałek, w – wtorek, s – środa, c – czwartek, t – piątek, b – sobota.

Przykład wywołania programu:

```
Kalendarz.exe -r 2013 -o kalendarz2013.txt -d n
```

```
Kalendarz.exe -o kal2020 -d t -r 2020
```

Uruchomienie programu z parametrem -h :

```
Kalendarz.exe -h
```

powoduje wyświetlenie krótkiej pomocy obsługi programu.

Próba uruchomienia programu w sposób inny niż przewidziany przez program (inny niż podany w pomocy) powoduje wyświetlenie odpowiedniego komunikatu, np.:

```
„Niepoprawny znak przełącznika” / „Niepoprawny format danych” /  
„Niepoprawne dane”
```

wraz z krótką pomocą.

Niepowodzenie podczas tworzenia pliku tekstowego powoduje
wyświetlenie powiadomienia:

```
„Proba otwarcia pliku zakończona niepowodzeniem!”
```

4. Specyfikacja wewnętrzna

Program został oparty o działanie pętli.

4.1 Typy zdefiniowane w programie:

Brak typów zdefiniowanych w programie.

4.2 Ogólna struktura programu

I W pierwszej kolejności program „pobiera” parametry przekazane z konsoli i przypisuje odpowiadające im numery odpowiednio do nazw zmiennych „rok”, „nazwa”, „dzień”.

Wykorzystuje w tym celu funkcje if i pętle for.

Dzięki temu powstają dane: argv[rok], argv[nazwa] i argv[dzień].

W razie wystąpienia błędów, wyświetlany jest stosowny komunikat.

II Następnie tworzony jest plik tekstowy o nazwie zawartej w zmiennej argv[nazwa]. Próba otwarcia tego pliku zakończona powodzeniem powoduje przejście do dalszych etapów. Nieudana próba otwarcia wyświetla informujący o tym komunikat.

III W programie zdefiniowane zostają tablice typów, wykorzystane później do tworzenia kalendarza.

IV Program przypisuje dniom tygodnia odpowiednie wartości numeryczne, za pomocą funkcji switch.

V Definiowana jest zmienna „wciecie”, wykorzystywana później w programie.

VI Funkcja wykorzystująca wszystkie dotychczas pobrane dane wyświetla

4.3 Szczegółowy opis implementacji funkcji

1)

```
int zeller(int d, int m, int y)
{
    return ((d += m<3 ? y-- : y - 2, 23 * m / 9 + d + 4 + y / 4 - y / 100 + y / 400) % 7);
}
```

Funkcja wyliczająca dzień tygodnia dla zadanej daty. Przypisuje cyfry od „0” dla niedzieli do „6” dla soboty. Funkcja służy do wyliczenia dnia tygodnia przypadającego na 1. stycznia danego (parametrem w konsoli) roku. (wywołanie: **zeller(1,1,argv[rok])**)

2)

```
for (int i = 0; i < 12; i++)
{
    int ile_dni = ilosc_dni[i];

    if (i == 1)
        if (atoi(argv[rok]) % 4 == 0 && !(atoi(argv[rok]) % 100 == 0) ||
            atoi(argv[rok]) % 400 == 0) ile_dni = 29;
```

Początek pętli wyświetlającej 12 miesięcy kalendarza wraz z nazwami dni i ich numeracją. „ile_dni” ogranicza numerację dni do liczby dni w danym miesiącu (odwołanie do tablicy ilosc_dni). Uwzględniany jest rok przestępny, kiedy luty ma 29 dni.

3)

```
for (int a = 0; a < 7; a++)
{
    if (dzien1 + a < 7) zapis << week[dzien1 + a] << "    ";
    else zapis << week[dzien1 + a - 7] << "    ";
}
zapis << endl;
```

Wypisanie nazw dni tygodnia w odpowiedniej kolejności zdefiniowanej parametrem po przełączniku -d. Dni tygodnia jest 7, więc zostanie

wypisanych 7 nazw, jedna obok drugiej.

4)

```
int licznik_dni = 1;
    for (int a = 0; a < 6; a++)
    {

        if (licznik_dni > ile_dni) break;

        if (a == 0)
        {
            for (int b = 0; b < wciecie; b++) zapis << "    ";
            for (int b = 0; b < 7 - wciecie; b++) zapis << licznik_dni++ << "    ";
            zapis << endl;
            wciecie = 0;
            continue;
        }
    }
```

Funkcja wypisująca numery dni w danym miesiącu. Funkcja ta uwzględnia wyliczoną wcześniej wartość "wciecie". Wykorzystując zmienną "licznik_dni" zaczyna numerację od 1. Wykona się maksymalnie 6 razy, ponieważ 6 to maksymalna liczba wierszy tygodni w miesiącu. Pierwszy wiersz tygodnia wykorzystuje zmienną "wciecie" wyliczoną w oparciu o wzór Zellera. Po wypisaniu pierwszego tygodnia miesiąca wartość "wciecie" jest zerowana, by kolejne wiersze kalendarza były wypisywane od początku do końca wiersza, czyli 7 razy na wiersz.

5)

```
for (int b = 0; b < 7; b++)
{
    if (licznik_dni > ile_dni)
        wciecie = b;
        break;
    }
    zapis << licznik_dni++ << " ";
}
```

Jeśli liczba wypisanych dni zrównała się z liczbą dni danego miesiąca, zapamiętywana jest wartość zmiennej "wciecie" równa b, czyli liczbie dni już wypisanych w ostatnim wierszu danego miesiąca. Umożliwi to rozpoczęcie numeracji dni kolejnego miesiąca od miejsca w wierszu, w którym zakończyła się numeracja poprzedniego miesiąca.

5. Testowanie

Program został przetestowany przy różnego rodzaju parametrach. W przypadku podawania danych niewystarczających do prawidłowego wygenerowania kalendarza, zostają wyświetlane odpowiednie komunikaty. Przy prawidłowych parametrach program działa poprawnie.

6. Wnioski

Program generujący kalendarz może składa się z podstawowych funkcji poznawanych przy okazji podstaw języka C++, lecz dla mnie (niezaznajomionego wcześniej z żadnym językiem programowania) stanowił on nie lada problem. Największą trudność sprawiało mi wymyślenie zasady, na jakiej program mógłby działać. Również wprowadzanie parametrów z konsoli było dla mnie czymś zupełnie obcym. Jednak po zasięgnięciu informacji od ludzi bardziej zaznajomionych z tematem, dałem radę stworzyć kod. Praca nad kodem i zabezpieczanie działania programu przed niewłaściwymi danymi nauczyły mnie jednak najwięcej. Uważam, że wiem o języku C++ więcej, niż przed przystąpieniem do projektu.