

Politechnika Śląska w Gliwicach  
Wydział Automatyki, Elektroniki i Informatyki



# Podstawy Programowania Komputerów

„Łamacz kodów”

---

autor	Kamil Lewandowski
prowadzący	dr inż. Maciej Długosz
rok akademicki	2016/2017
kierunek	informatyka
rodzaj studiów	SSI
semestr	1
termin laboratorium / ćwiczeń	wtorek, 12:00-13:30
grupa	6
sekcja	1
termin oddania sprawozdania	2016-12-02
data oddania sprawozdania	2016-12-01

---

# 1 Treść zadania

Proponowany program jest grą wzorowaną na grze „Mastermind”. Polega ona na wyborze przez gracza (gracz kodujący) czterech pionków spośród zapasu pionków w sześciu kolorach. Zadaniem przeciwnika (gracza odgadującego) jest określenie ich sekwencji poprzez przedstawienie serii co najwyżej dziesięciu propozycji. Po każdej propozycji odgadujący otrzymuje informację o liczbie pionków, które zostały odgadnięte prawidłowo oraz o liczbie pionków, których kolor został dobrany właściwie, jednak umieszczonych przez odgadującego w niewłaściwych miejscach.

Celem programu jest dostarczenie uproszczonej wersji gry, w której rolę gracza kodującego pełni komputer (wybierający serię pionków w sposób losowy oraz udzielający informacji o liczbie trafionych kolorów). Zadaniem użytkownika jest podjęcie próby odgadnięcia sekwencji komputera.

## 2 Analiza zadania

Zagadnienie przedstawia problemy operowania danymi oraz porównywania.

### 2.1 Struktury danych

W programie wykorzystano strukturę posiadającą dane potrzebne do poprawnego działania programu. Takie jak na przykład dane podane przez użytkownika lub szyfr wylosowany przez program.

### 2.2 Algorytmy

Najbardziej złożony algorytm w programie zaimplementowany jest w funkcji `spr()`. Polega on na porównywaniu dwóch tabel. Jego złożoność wynosi  $O(n^2)$ .

## 3 Specyfikacja zewnętrzna

Program może być uruchamiany z konsoli. Żadne dane nie są wymagane. Cała interakcja z użytkownikiem odbywa się podczas działania programu. W programie zostało wykorzystane proste menu. Za jego pomocą użytkownik może komunikować się z programem. W razie

wprowadzenia błędnych danych program wyświetli odpowiedni komunikat. Dopuszczane znaki do wpisania w konsoli to cyfry od 1 do 9 oraz w(małe bądź duże).

## 4. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji(sprawdzania poprawności wpisanego szyfru).

### 4.1 Typy zdefiniowane w programie

W programie zdefiniowano następujący typ:

```
//-----  
1.  struct GraS  
2.  {  
3.  int stanGry; //zmienna określająca stan gry  
  
4.  int szyfr[4]; //tabela która przechowuje wylosowany szyfr  
  
5.  int Tab[8][6]; //tabela która przechowuje dane podane przez użytkownika  
    oraz punkty pomocnicze  
6.  int runda; //zmienna określająca rundę(wpisanie cyfr składających się na  
    szyfr)  
  
7.  int tura; //zmienna określająca turę(pojedyncze wpisanie cyfry)  
  
8.  int kont; //zmienna pomocnicza określająca czy w pliku są dane  
  
9.  int blad; //zmienna pomocnicza określająca czy zostały wpisane błędne dane  
10. };  
//-----
```

Typ ten przechowuje wartości niezbędne do poprawnego funkcjonowania programu.

### 4.2 Ogólna struktura programu

W głównej funkcji programu jest stworzona pętla ,która obsługuje całą grę. Stan w którym znajduje się gra definiuje pole struktury GraS o nazwie stanGry.

```
//-----  
1. int menu(GraS& game); // metoda wyświetlająca menu i określająca stan gry w  
    zależności od pobranych danych  
//-----
```

Przy pomocy tej funkcji określany jest stan gry. W zależności od niego program kończy prace, tworzy nową rozgrywkę lub wczytuje wcześniej zapisaną.

```
//-----  
1. void xGra(GraS& game, int k); // w zaleznosci od argumentu wczytuje gra badz  
   rozpoczyna na nowo  
//-----
```

W tej funkcji odbywa się cała rozgrywka. Parametr k definiuje czy program tworzy nowa grę czy wczytuje z pliku zapisaną już rozgrywkę.

### 4.3 Szczegółowy opis implementacji funkcji

W programie można znaleźć takie funkcje jak:

```
//-----  
1. void spr(GraS& game); // metoda sprawdzajaca poprawnosc podanego przez  
   uzytkownika kodu  
//-----
```

Funkcja sprawdza odpowiedzi użytkownika. Funkcja nie zwraca wartości, zamiast tego wpisuje wyniki do tabeli utworzonej struktury GraS .

```
//-----  
1. bool sprGra(); //sprawdza czy w pliku jest zapis gry  
//-----
```

Aby nie wczytywać danych z pustego pliku program sprawdza dany plik przy pomocy tej funkcji. Funkcja zwraca wartości „true” bądź „false”.

```
//-----  
1. int sprDane(char d); // sprawdza poprawnosc danych wprowadzonych przez  
   uzytkownika  
//-----
```

Sprawdza dane które podał użytkownik. Zwraca wartość 2 lub 1 gdy dane są poprawne i 0 gdy dane są błędne.

```
//-----  
1. void wczytajGre(GraS& game); // wczytuje gre z pliku  
//-----
```

Funkcja pobiera wartości z pliku i wpisuje je do odpowiednich pól i tablic obiektu typu GraS. Funkcja nic nie zwraca.

```
//-----  
1. void zapiszGre(GraS game); // zapisuje gre do pliku  
//-----
```

Funkcja zapisuje wartości z poszczególnych pól i tablic obiektu typu GraS do pliku. Funkcja nic nie zwraca.

```
//-----  
1. void nowaGra(GraS& game); // tworzy nowa gre
```

```
//-----  
Funkcja przypisuje odpowiednie początkowe wartości do pól i tabel obiektu typu GraS. Funkcja nic nie zwraca.
```

```
//-----  
1. void losuj(GraS& game); //losuje szyfr
```

```
//-----  
Funkcja wywoływana w funkcji nowaGra losuje szyfr. Funkcja nie zwraca żadnej wartości.
```

```
//-----  
1. void ekran(GraS game); // metoda wyswietla ekran rozgrywki
```

```
//-----  
Funkcja wyświetla cały ekran rozgrywki. Funkcja nic nie zwraca.
```

## Testowanie

Do programu wprowadzane były różne dane. Program działał poprawnie wyłącznie dla cyfr lub ciągów cyfr. Po wprowadzeniu odpowiednich zabezpieczeń program działa poprawnie również dla znaków i liter, wypisując że dane są błędne.

Pojawiły się pewne problemy z operacjami na plikach. Program nie potrafił otwierać plików tekstowych. Udało się odnaleźć źródło błędu i je usunąć.

## Wnioski

Do stworzenia gry łamacz kodów nie jest potrzebna wysoka znajomość technologii C++. Jest to program dość prosty w działaniu. Najcięższe okazało się stworzenie sytemu menu. Po wielu poprawkach ostatecznie udało mi się stworzyć w pełni działającą grę z możliwością zapisu stanu gry do pliku. Dzięki temu projektowi lepiej opanowałem operacje na plikach.