

# Instrukcja projektowa; projekt numer 1

---

*Podstawy Programowania 2016/2017, kierunek Informatyka*

autor: Krzysztof Bruniecki<sup>1</sup>

wersja z dnia: 2016-10-20

## Projekt: Tekstowy edytor obrazków

---

### Cel

Zadanie polega na implementacji programu **konsolowego** wyświetlającego tekstowo podgląd i pozwalającego na tekstową edycję obrazka wczytanego z pliku.

### Ogólne wytyczne

Projekt powinien być napisany w języku C z możliwością stosowania podstawowych elementów języka C++ (podobnie jak podczas laboratoriów). Projekt może być pisany w sposób obiektowy, ale całkowicie **zabronione jest użycie biblioteki standardowej C++ (w tym typu string, cin, cout, vector itp.)**.

Obsługa plików powinna być zrealizowana za pomocą funkcji z rodziny f???? (fopen, fread itp.) oraz typu FILE. Nie można w tym celu używać mechanizmów C++ (np. fstream).

Za projekt można uzyskać 0-18 punktów. 15 punktów stanowi tzw. 100% pozostałe 3 punkty to wymagania ponadprogramowe ("z gwiazdką").

Realizacja wymagań obowiązkowych konieczna jest aby projekt był w ogóle oceniany.

Program powinien być napisany z użyciem szablonu dostępnego w materiałach. Szablon umożliwia uzyskanie zaawansowanych możliwości w zakresie obsługi terminala w systemie Windows. **Zabronione jest używanie instrukcji cin/cout/printf/scanf.** W celu pisania na ekran i czytania z klawiatury, należy używać wyłącznie metod dostępnych w szablonie.

---

<sup>1</sup> Uwaga: W razie niejasności lub niejednoznaczności w poniższym opisie proszę kontaktować się z autorem instrukcji; pokój 738EA; konsultacje odbywają się w poniedziałki 12:15-14:00; adres [www: http://pg.edu.pl/9dc69f4a9d\\_krzysztof.bruniecki](http://pg.edu.pl/9dc69f4a9d_krzysztof.bruniecki)

**Uwaga!** Udostępniony szablon działa w systemie Windows i nie ma możliwości łatwego przeniesienia jej do innych systemów operacyjnych. Projekt należy zatem przygotować **wyłącznie dla systemu Windows** z wykorzystaniem szablonu.

Podczas oddawania (aby usprawnić oddawanie) należy przygotować obrazki przykładowe. Student oddający projekt powinien móc wczytać je z pliku po wybraniu klawisza "i".

**Pamiętaj żebyś przygotował się do odbioru - koniecznie sprawdź wcześniej czy potrafisz sprawnie uruchomić swój program na komputerze w laboratorium!**

## Obsługa programu

Program powinien wykorzystywać klawiaturę w następujący sposób (zarówno duże jak i małe litery):

strzałki	przemieszczanie się kursorem po pikselach obrazka oraz przemieszczanie się w menu (parz wymagania z gwiazdką)
esc	wyjście z programu/anulowanie trybu rysowania linii lub prostokąta
i	wczytanie pliku z obrazkiem domyślnym
o	wczytywanie pliku z możliwością podania nazwy przez użytkownika
n	utworzenie nowego pliku z możliwością podania nazwy przez użytkownika
s	zapis pliku z możliwością podania nazwy przez użytkownika - jeśli użytkownik nie zmieni nazwy oryginalna nazwa jest stosowana
l	rysowanie linii
k	rysowanie prostokąta
1234567890qwerty	wybór koloru z palety
ctrl+strzałki	przewijanie ekranu
backspace	cofnij
alt+strzałki	zaznaczenie obszaru do skopiowania
c	kopiuj zaznaczony fragment do schowka
p	wklej fragment ze schowka w miejscu kursora
f	wypełnianie od punktu położenia kursora

## Wymagania obowiązkowe (5 pkt.)

Do wymagań podstawowych należy obsługa obrazków o wymiarach nie przekraczających **50x20** w trybie "2-kolorowym" - strzałki przesuwają kursor, spacja przełącza kolor biały/czarny

Tworzenie nowego obrazka o wpisanych wymiarach. Podczas tworzenia obrazka użytkownik podaje wymiary. Przy tworzeniu nowego obrazka jego zawartość powinna być jednolita (np. czarna).

Możliwość odczytu obrazka z pliku we własnym formacie tekstowym. Student powinien samodzielnie opracować sposób kodowania pliku i umożliwić łatwą zmianę palety (czyli znaków odpowiadających kolorom biały/czarny).

Wyświetlany obrazek nie ma zajmować całego okna, lecz jego większą część. W pozostałym obszarze należy wyświetlać:

- literowe **skróty** do wszystkich zaimplementowanych funkcjonalności,
- **bieżący tryb i stan dostępności funkcji** (np. czy funkcja "undo" jest dostępna w danej chwili, czy w schowku coś jest i jaki ma rozmiar, czy możliwe jest wklejenie w całości w bieżącym położeniu kursora, tryb rysowania linii, tryb rysowania prostokąta, tryb przewijania itp.)
- **nazwa pliku,**
- **pozycja kursora.**

Domyślnie, skróty i pozostałe informacje dodatkowe mają być wyświetlane po lewej stronie ekranu, student może zostać jednak poproszony o szybką zamianę miejscami wybranych fragmentów interfejsu graficznego więc powinno to być dobrze zaprojektowane - np. z użyciem stałych.

## Wymagania nieobowiązkowe (10 pkt.)

Do wymagań dodatkowych należą:

**1pkt.** zapis we własnym formacie

**1pkt.** zapis/odczyt do pliku w formacie BMP

**1pkt.** zapis/odczyt do pliku w formacie XPM2

**0.5pkt.** otwieranie pliku podanego w linii poleceń

**0.5pkt.** rysowanie prostokątów

1. wskazanie punktu początkowego
2. wciśnięcie klawisza "k"
3. wskazanie punktu końcowego
4. ponowne wciśnięcie klawisza "k"

Powinna być możliwość przerywania rysowania za pomocą klawisza esc.

**1pkt.** interaktywne rysowanie linii i prostokątów (wskazanie punktu od, wskazanie punktu do, linia może być ukośna),

Wymaganie polega na dodaniu rysowania linii jak również trybie interaktywnym dla prostokąta i linii (poprawna realizacja tego wymagania oznacza jednocześnie spełnienie poprzedniego wymagania)

Rysowanie powinno odbywać się w trybie interaktywnym, co w przypadku linii oznacza następujące zachowanie programu:

za pomocą kursora użytkownik przemieszcza się w wybrany punkt  $(x,y)$ , klika klawisz "l", program zapamiętuje wówczas współrzędne  $(x,y)$  jako początek linii. Gdy następnie użytkownik przesuwając kursor po obrazku, to na bieżąco wyświetlany jest podgląd linii pomiędzy  $(x,y)$ , a bieżącą pozycją kursora. Dopiero gdy użytkownik wciśnie ponownie "l" wówczas linia nanosi się na obrazek. Innymi słowy mamy podgląd jak rysujemy.

Powinna być możliwość anulowania w takim interaktywnym trybie rysowania za pomocą klawisza esc.

**1pkt.** obsługa edycji w trybie 16 kolorów

**1pkt.** implementacja funkcji cofnij. Bez ograniczenia z góry liczby poziomów (limitowane jedynie dostępną pamięcią alokowaną dynamicznie). Należy zadbać o niezwłoczne zwalnianie tej pamięci gdy nie jest już potrzebna - np. gdy obrazek został zapisany i zamknięty.

**1pkt.** wypełnianie ograniczonego obszaru jednolitego koloru począwszy od wskazanego punktu (obszar ograniczony innym kolorem lub brzegiem obrazka, przyjmujemy założenie że piksele stykające się rogami również stanowią ograniczenie)

**1pkt.** zaznaczenie i kopiowanie fragmentu obrazka do schowka oraz wklejanie fragmentu obrazka ze schowka

**1pkt.** obsługa obrazków o wymiarach przekraczających rozmiar okna (zapewnienie przewijania). W ramach tego wymagania należy również zapewnić sygnalizowanie pozycji widocznego fragmentu obrazka (paski z pozycją przy prawej i dolnej krawędzi -- tzw.: scrollbary, ale bez konieczności obsługi klikania na nich)

## Wymagania z gwiazdką

**1pkt.** Implementacja funkcji zakładki - kilka plików otwartych jednocześnie (bez ograniczenia na liczbę plików - pamięć alokowana dynamicznie), kopiowanie fragmentów pomiędzy plikami

**2pkt.** Implementacja własnego tekstowego okienka dialogowego do zapisu i odczytu plików. Okienko zapewnia wyświetlanie listy plików, katalogów oraz możliwość poruszania się za pomocą strzałek po drzewie katalogów.

## Uwagi

### **Aktualizacja: 20 października 2016**

W związku z pojawieniem się pytań dot. dopuszczalności programowania obiektowego, poniżej drobne uściślenie.

#### **Programowanie obiektowe jest dopuszczalne m.in. w zakresie:**

- implementacji własnych klas, funkcji klas, szablonów,
- używania hermetyzacji, polimorfizmu, dziedziczenia, abstrakcji itp.,
- używania alokacji za pomocą new oraz delete (ew. new[] delete[]).

#### **Kategorycznie nie jest dopuszczalne:**

- użycie kontenerów `std::vector` itp.,
- użycie algorytmów i innych elementów z biblioteki C++, dla których nie ma bezpośrednich odpowiedników w bibliotece standardowej C,
- użycie typu `string`, `cin`, `cout`, `fstream`,
- użycie bibliotek dodatkowych.

Należy pamiętać, że używanie konstrukcji programistycznych wykraczających poza tematykę wykładu nakłada na studenta obowiązek samodzielnego i kompletnego zapoznania się z zasadami działania takich elementów, gdyż mogą one być przedmiotem pytań oceniającego projekt (odpowiedzi w stylu: "znalazłem na stronie X przykład i działa" są zdecydowanie niewystarczające i spowodują obniżenie oceny).