

Praktikum iz objektno orijentisanog programiranja (13S112POOP)

Projektni zadatak – C++

Napisati skup klasa sa odgovarajućim metodama, konstruktorima, operatorima i destruktorkama za realizaciju softverskog sistema za manipulaciju digitalnim slikama. Potrebno je obezbediti osnovnu manipulaciju nad slikama, kao i generisanje fajlova različitih formata. Podržani formati fajlova treba da budu rasterski formati BMP i PAM, a treba predvideti i mogućnost proširenja drugim formatima. Opis formata navedenih fajlova je dat u prilogu ovog dokumenta.

Korisnik (naručilac) softvera, želi da softver pruži sledeće funkcionalnosti:

- Interakciju sa korisnikom putem tekstualnog menija ili grafičkog korisničkog interfejsa
- Učitavanje slike
- Osnovnu manipulaciju nad slikom
 - Rad sa slojevima
 - Definisanje selekcija
 - Operacije nad slikom
- Eksportovanje slike
 - BMP i PAM format
- Snimanje projekta
 - Sopstveni format kojim se serijalizuju relevantni podaci (slojevi, selekcije, kompozitne operacije)
- Kraj rada

Za uspešno rešenje zadatka potrebno je izvršiti analizu zahteva. Kao rezultat analize, potrebno je dopuniti i precizirati funkcionalnu specifikaciju softverskog alata. Na osnovu specifikacije, potrebno je napisati sistem klasa u jeziku C++ koje realizuju traženi softver. U nastavku su navedeni neki elementi specifikacije. Od studenata se očekuje da dopune one stavke koje nisu dovoljno precizno formulisane, odnosno dodaju nove stavke (tamo gde to ima smisla) ukoliko uoče prostor za unapređenje. Izmene i dopune specifikacije mogu da donekle odudaraju od zahteva naručioca softvera u onoj meri u kojoj to neće narušiti traženu funkcionalnost. Takođe, priloženi UML dijagram koji opisuje zahtevani softver se ne mora obavezno poštovati, već samo predstavlja skicu potencijalnog rešenja. Prilikom izrade specifikacije voditi računa o potencijalnom unapređenju softvera na osnovu naknadnih zahteva.

Prilikom izrade rešenja, od studenata se očekuje intenzivno korišćenje svih onih mogućnosti koje pružaju specifikacija jezika C++ i biblioteka STL, kao što su šablonske funkcije, kolekcije, algoritmi, regularni izrazi, iteratori, lambda izrazi i sl. **Rešenja koja ne vode računa o ovom aspektu neće moći da dobiju maksimalan broj poena.** Takođe, voditi računa o **objektno orijentisanom dizajnu rešenja**, čistoći, čitkosti i komentarisanju programskog koda.

Funkcionalna specifikacija

U nastavku je zadat deo korisničkih zahteva koje treba razraditi i, po potrebi, dopuniti tako da se dobije funkcionalna aplikacija.

Interakcija sa korisnikom

Korisnik može da interaguje sa programom bilo izborom odgovarajućih opcija iz tekstualnog menija putem tastature ili izborom u datom trenutku dostupnih opcija putem grafičkog korisničkog interfejsa. Nije potrebno realizovati oba načina. Interakcija u slučaju grafičkog interfejsa može da se vrši putem tastature ili miša. U zavisnosti od izabrane opcije i njenih parametara, program izvršava zadatu opciju ili ispisuje poruku greške. Poruka greške treba da bude što je moguće detaljnija da bi korisniku pomogla da grešku otkloni. Sve eventualne parametre koji su potrebni prilikom rada aplikacije je potrebno zatražiti od korisnika. Ukoliko korisnik ne zada ništa, koristiti vrednosti fiksirane u programu.

Učitavanje podataka

Potrebno je podržati učitavanje slika u BMP i PAM formatu. Pored toga, omogućiti učitavanje slike i iz sopstvenog formata (projekta) u kome su pored same slike sačuvane i informacije o slojevima, selekcijama i kompozitnim operacijama (videti u nastavku dokumenta). Sopstveni format projekta koji se uvodi mora biti **tekstualnog tipa** i dozvoljeno je da čuva reference (putanje u fajl sistemu) ka izvornim slikama. Prilikom učitavanja podataka koristiti regularne izraze za parsiranje sopstvenog formata. Predvideti način za oporavak od grešaka u slučaju neispravnog formata datoteke ili nepostojanja tražene datoteke. Prilikom definisanja sopstvenog formata voditi računa da će isti format projekta biti korišćen i u Java projektu, te da ga je potrebno pažljivo izabrati. Preporuka je da se koristi XML format. Više o ovom formatu može se pronaći na sledećem linku: <https://www.w3schools.com/xml/default.asp>. Za manipulaciju ovim formatom (parsiranje, kreiranje, menjanje) moguće je naći gotove C++ biblioteke.

Osnovna manipulacija nad slikom

Korisniku je potrebno omogućiti sledeće:

- **Rad sa slojevima**

Slojevi se ponašaju kao prozirne folije poredane po dubini, od 1 do N, gde je folija 1 na vrhu. Deo slike na sloju X se vidi u onoj meri u kojoj delovi slike koji potiču od slojeva 1, 2, ... X-1 to dozvoljavaju svojom prozirnošću. Prozirnost sloja se izražava u opsegu od 0 do 100 (0 – potpuno proziran, 100 – neproziran). Korisnik može da bira aktivne slojeve na koje se odnose operacije koje se sprovode nad slikom. Takođe, korisnik može da bira vidljive slojeve, odnosno slojeve koji učestvuju u formiranju slike. Svi slojevi su uvek istih dimenzija i prostiru se preko cele slike. Prilikom kreiranja sloja moguće je odrediti koja slika će se nalaziti na tom sloju (zadavanjem putanje do te slike koja mora biti u BMP ili PAM formatu). Ukoliko su dimenzije slike manje od tekuće dimenzije slojeva koji su kreirani, vrši se popunjavanje sloja

providnim pikselima, dok se u slučaju da su dimenzije slike veće vrši povećavanje dimenzija svih postojećih slojeva (popunjavanjem providnim pikselima) do dimenzija novokreiranog sloja. Prilikom kreiranja prvog sloja korisnik može izabrati sliku koja se na njemu nalazi i u tom slučaju je dimenzija sloja jednaka dimenziji učitane slike, dok u slučaju kada korisnik ne želi da specificira sliku moraju biti zadate dimenzije prvog sloja. Sloj je moguće obrisati.

- **Rad sa selekcijama**

Selekcija predstavlja uniju jednog ili više pravougaonih delova slike. Korisnik bira selekciju po imenu koje zadaje prilikom stvaranja. Prilikom stvaranja selekcije zadaje se i niz pravougaonih oblika od kojih je svaki opisan pozicijom levog gornjeg ugla, kao i širinom i visinom. Selekcija može biti aktivna ili neaktivna. U izvođenju operacija nad slikom, koje su definisane u nastavku, učestvuju svi aktivni slojevi, uzimajući u obzir selekciju. Obezbediti mogućnost popunjavanja selekcije zadatom bojom. Selekciju je moguće obrisati.

- **Operacije nad slikom**

U slučaju da postoji aktivna selekcija, operacije se primenjuju samo nad selektovanim pikselima. U suprotnom, operacije se primenjuju nad celom slikom. Za dati piksel operacija se primenjuje nezavisno na svaku od tri osnovne komponente (R, G i B), osim ako nije drugačije naznačeno. Vrednosti komponentata predstavljaju se celim brojem u opsegu od 0 do 255. Rezultat bilo koje operacije može biti van ovog opsega, ali konačna vrednost, nakon završenog računanja, mora biti u ovom opsegu. Operacije nad datim pikselom su sledeće:

- Osnovne aritmetičke operacije, gde je jedan operand tekuća vrednost piksela, a drugi zadata konstanta: sabiranje, oduzimanje (oduzimanje konstante od tekuće vrednosti), inverzno oduzimanje (oduzimanje tekuće vrednosti od konstante), množenje, deljenje i inverzno deljenje.
- Funkcije: power (podizanje tekuće vrednosti na stepen zadat konstantom), log, abs, min (tekuća vrednost i konstanta – sve komponente veće od zadate konstante se postavljaju na zadatu konstantu), max (sve komponente manje od zadate konstante se postavljaju na zadatu konstantu).
- Pravljenje proizvoljne imenovane kompozitne funkcije ulančavanjem osnovnih aritmetičkih operacija i drugih funkcija (prostih ili složenih).
- Predvideti sledeće predefinisane funkcije:
 - Inverzija: rezultujuća boja se dobija tako što se tekuća vrednost oduzme od maksimalne.
 - Pretvaranje u sliku u nijansama sive: komponente rezultujuće boje imaju istu vrednost izračunatu kao aritmetička sredina komponenti tekuće boje.
 - Pretvaranje u crno-belu sliku: boja piksela biće crna (0, 0, 0) ukoliko je aritmetička sredina R, G i B komponente niža od 127, dok će u suprotnom biti bela (255, 255, 255).
 - Medijana: rezultujuća boja se dobija kao medijana boja datog i susednih piksela.

- **Operacije sa kompozitnim funkcijama**

Kompozitnu funkciju je moguće eksportovati u proizvoljnom tekstualnom formatu u fajl sa ekstenzijom *.fun*. Preporuka je da se koristi isti format koji se koristi i prilikom čuvanja slike u

sopstvenom formatu. U bilo kom trenutku moguće je učitati kompozitnu funkciju iz datoteke čija se putanja zadaje.

Eksportovanje

Prilikom eksportovanja korisnik zadaje željeni format, kao i putanju do fajla u koji je potrebno smestiti rezultat eksportovanja. Potrebno je omogućiti eksportovanje slike u dva različita formata, kao i eksportovanje celog projekta u sopstvenom formatu. Podržati sledeće formate slike:

- PAM (Portable Arbitrary Map)
- BMP (Bitmap Image File)

Specifikacije ova dva formata date su u prilogu ovog dokumenta.

Prilikom eksportovanja u sopstvenom formatu potrebno je sačuvati sve informacije o svim slojevima, selekcijama i kompozitnim funkcijama.

Kraj rada

Korisnik može da zahteva kraj rada. Od korisnika se traži potvrda za napuštanje programa, uz upozorenje ukoliko program nakon poslednje izmene slike nije generisao neki fajl. Korisniku treba ponuditi mogućnost da eksportuje sliku pre napuštanja programa, ukoliko to želi.

Testiranje rada programa

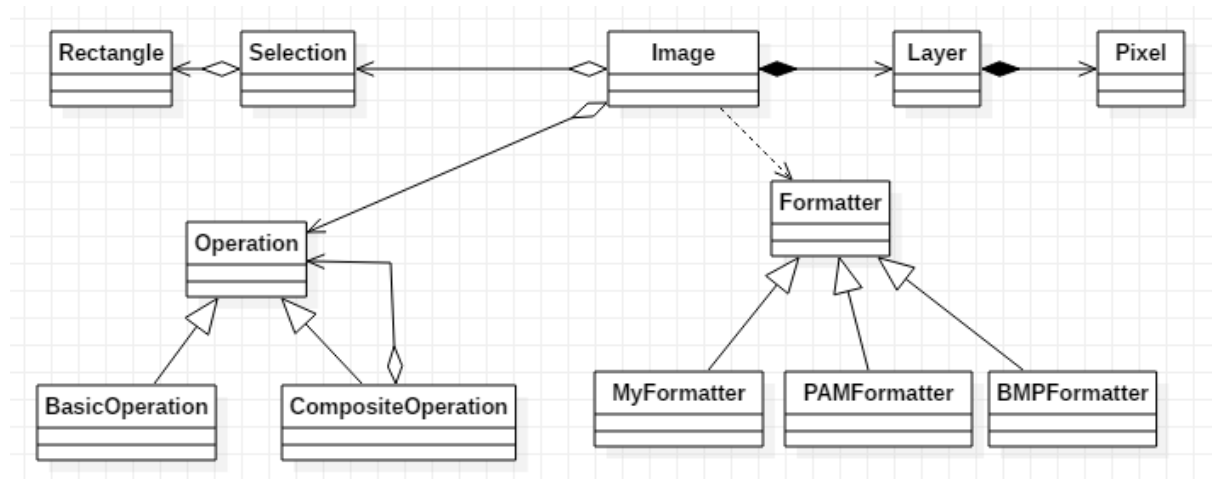
Program je moguće pokrenuti na dva načina:

1. Zadavanjem putanje do datoteke u kojoj se nalazi slika u proizvoljnom formatu i putanje do datoteke u kojoj se nalazi kompozitna funkcija. Prilikom pokretanja programa na ovaj način učitava se slika iz zadate datoteke, primenjuje se učitana kompozitna funkcija, izmenjena slika se čuva u istoj datoteci iz koje je učitana i program se završava. Ovaj režim rada programa biće korišćen u drugom delu projekta koji se odnosi na programski jezik Java. Sve transformacije slike obavljaće se izvršavanjem C++ programa.
2. Bez zadatih argumenata. Po pokretanju programa korisniku se prikazuje meni koji ga dalje vodi kroz program.

Prilikom testiranja rada programa moguće je koristiti programe za prikazivanje slika. BMP fajl može se proveriti pomoću alata za obradu slika, među kojima je i Paint, koji dolazi uz Windows operativni sistem. U nedostatku prikazivača PAM fajlova moguće je online konvertovati PAM fajl u neki drugi format. Konverzija u PNG format je dostupna na sledećem linku: <https://convertio.co/pam-png/>.

Dijagram klasa

Na osnovu prethodne funkcionalne specifikacije formiran je sledeći dijagram klasa. Dijagram klasa nije detaljan, te ga treba tumačiti kao skicu koja načelno ukazuje na arhitekturu softvera. Studenti mogu da koriste ovaj dijagram kao referencu i, po potrebi, prošire ga da bi ga usaglasili sa eventualnim dopunama specifikacije.



Prilikom implementacije rešenja, obratiti pažnju na objektno orijentisani dizajn i intenzivno koristiti kolekcije i algoritme standardne biblioteke jezika C++ i lambda funkcije gde god je to moguće. Primititi da postoje tri različita formata izlaznih fajlova.

Specifikacija BMP formata

BMP je rasterski format zapisa digitalnih slika. Kompletnu specifikaciju ovog formata moguće je pronaći na sledećem linku: https://en.wikipedia.org/wiki/BMP_file_format. Na navedenom linku su dati i primeri bmp fajlova, od kojih je jedan predstavljen i u nastavku ovog dokumenta.

Offset	Veličina	Hex vrednost	Vrednost	Opis
BMP zaglavlje				
0h	2	42 4D	"BM"	ID polje (42h, 4Dh)
2h	4	46 00 00 00	70 B (54+16)	Veličina BMP fajla
6h	2	00 00	Ne koristi se	-
8h	2	00 00	Ne koristi se	-
Ah	4	36 00 00 00	54 B (14+40)	Offset na kome počinje niz piksela
DIB (device independent bitmap) zaglavlje				
Eh	4	28 00 00 00	40B	Broj bajtova u DIB zaglavlju (od ove tačke) (fiksno)
12h	4	02 00 00 00	2 piksela	Širina slike
16h	4	02 00 00 00	2 piksela	Visina slike
1Ah	2	01 00	1	Jedna karata boja (fiksno)
1Ch	2	18 00	24 bits	Broj bita po pikselu (fiksno)
1Eh	4	00 00 00 00	0	BI_RGB (fiksno)
22h	4	10 00 00 00	16 B	Veličina bitmape u bajtovima
26h	4	13 0B 00 00	2835 pixels/metre	Ostaviti fiksirano
2Ah	4	13 0B 00 00	2835 pixels/metre	
2Eh	4	00 00 00 00	0	Ostaviti fiksirano
32h	4	00 00 00 00	0	Ostaviti fiksirano
Početak niza piksela				
36h	3	00 00 FF	0 0 255	Crveni piksel
39h	3	FF FF FF	255 255 255	Beli piksel
3Ch	2	00 00	0 0	Dopuna do 4 B
3Eh	3	FF 00 00	255 0 0	Plavi piksel
41h	3	00 FF 00	0 255 0	Zeleni piksel
44h	2	00 00	0 0	Dopuna do 4 B

U odnosu na navedeni primer, potrebno je promeniti ukupnu veličinu BMP fajla, širinu i visinu slike, ukupnu veličinu fajla koja se odnosi na niz piksela i, naravno, same piksele koji se nalaze od offseta 36h. Za svaki piksel se navode u obrnutom redosledu, kao što se vidi iz primera. Svaki red slike potrebno je dopuniti nulama kako bi ukupan broj bajtova za jedan red slike bio deljiv sa 4. Za potrebe ovog projekta biće potrebno pored R, G i B kanala dodati i A (alpha) kanal kako bi se predstavila prozirnost. Na linku koji je dat moguće je naći primer i sa poluprozirnim pikselima.

Specifikacija PAM formata

Pam je rasterski format zapisa digitalnih slika i deo je Netpbm open-source paketa. Kompletanu specifikaciju ovog formata moguće je pronaći na sledećem linku: https://en.wikipedia.org/wiki/Netpbm#PAM_graphics_format. Svaki PAM fajl počinje sledećim zaglavljem:

P7 # oznaka PAM fajla, magic number P7
WIDTH w # w – širina slike
HEIGHT h # h – visina slike
DEPTH d # d – broj kanala (1 – greyscale, 3 – RGB, 4 – RGBA)
MAXVAL m # m – maksimalna vrednost (255 za RGB)
TUPLTYPE t # t – tip torke (dat u tabeli ispod)
ENDHDR # oznaka kraja zaglavlja

t	m	d	Objasnjenje	Prozirnost (alpha)
BLACKANDWHITE	1	1	Crno-bela slika	Ne
GRAYSCALE	2...65535	1	Slika u nijansama sive	Ne
RGB	1...65535	3	Slika u RGB formatu	Ne
BLACKANDWHITE_ALPHA	1	2	Crno-bela slika	Da
GRAYSCALE_ALPHA	2...65535	2	Slika u nijansama sive	Da
RGB_ALPHA	1...65535	4	Slika u RGB formatu	Da

Tabela 1 Tipovi torke u PAM fajlu

Nakon ASCII zaglavlja slede binarno zapisani podaci o pikselima u zavisnosti od izabaranog tipa torke (t). Podaci o pikselima jednog reda se zapisuju jedni za drugima bez separatora. Između piksela različitih redova potrebno je ubaciti ASCII kod znaka za prelaz u novi red (0x0A).

Za kreiranje, prikaz i menjanje binarnih datoteka moguće je koristiti alat Hex Editor Neo. U nastavku su prikazani primer PAM fajla i slika koju taj fajl opisuje.

