

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 muzičkim kompozicijama. Potrebno je obezbediti osnovnu manipulaciju nad kompozicijama, kao i generisanje fajlova različitih formata koji se mogu koristiti u alatima za vizuelizaciju i reprodukciju. Podržani formati fajlova treba da budu MUSICXML, BMP i MIDI 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:

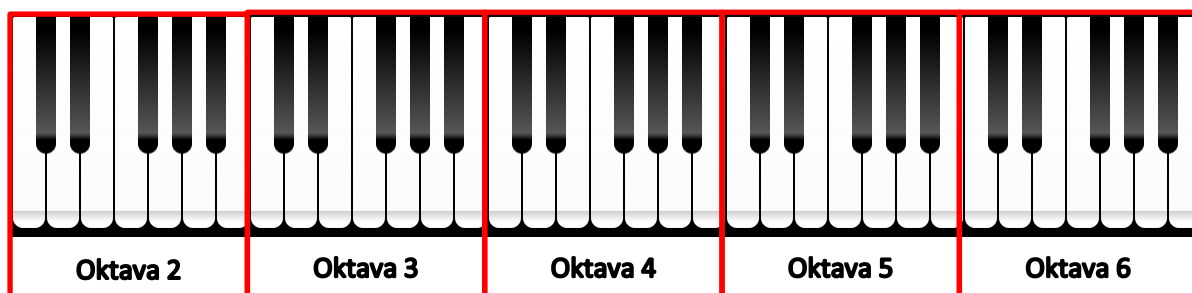
- Interakciju sa korisnikom putem tekstualnog menija ili grafičkog korisničkog interfejsa
- Učitavanje podataka
 - učitavanje podataka o kompoziciji
 - učitavanje podataka o notama
- Osnovnu manipulaciju nad kompozicijom
 - prikaz podataka o kompoziciji
 - iteriranje kroz kompoziciju uz mogućnost izmene muzičkih simbola
 - promena takta kompozicije i pomeranje kompozicije kroz oktave
- Eksportovanje kompozicije
 - formatiranje fajlova po MUSICXML, BMP ili MIDI formatu
- 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 komentaranju programskog koda.

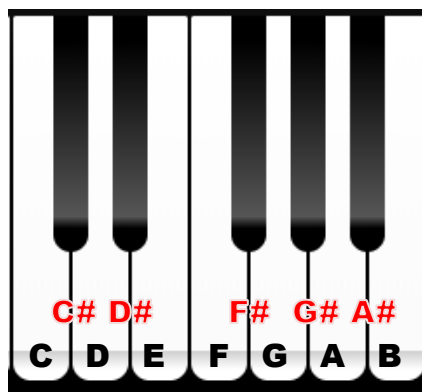
Osnovni pojmovi

Kompozicija se sastoji iz dva linijska sistema. Prvi sadrži deo kompozicije koji se svira levom rukom, a drugi deo kompozicije koji se svira desnom rukom. Svaki od linijskih sistema se sastoji od taktova. Svaki takt se sastoji od muzičkih simbola. Muzički simboli su nota i pauza. Muzički simbol ima svoje trajanje koje se zadaju u vidu razlomka. Nota je opisana oktavom (ceo broj u opsegu 2-6) i visinom (C, D, E, F, G, A, B, poznatije kao: DO, RE, MI, FA, SOL, LA, SI). Izgled uprošćene verzije klavira koja se koristi u ovom projektu sa označenim oktavama dat je na slici 1.



Slika 1. Izgled klavira

Notu je moguće povisiti dodavanjem zanaka # (povisilica), te je povišena nota C u stvari C# (otuda i naziv programskog jezika). Povišene note se na klaviru sviraju na crnim dirkama desno od osnovne note. Složena nota (dvozvuk ili akord) se sastoji od više drugih nota koje se sviraju istovremeno, a sve moraju biti istog trajanja. Takt je opisan trajanjem (vrsta, mera) koje određuje maksimalno trajanje muzičkih simbola u okviru takta. Svi taktovi u okviru kompozicije su jednakog trajanja.



Na slici 2. dat je uvećan prikaz jedne oktave sa označenim tonovima.

Crnim slovima su na dirkama označeni tonovi koje predstavljaju bele dirke, dok su crvenim slovima označeni tonovi koje predstavljaju crne dirke iznad njih, tj. povišeni tonovi (snizilice nećemo razmatrati u ovom projektu). Raspored tonova je isti u svim oktavama, a radi jednostavnosti uzeto je da se klavir sastoji od 5 celih oktava.

Slika 2. Izgled jedne oktave

Na slici 3. prikazani su simboli koji će biti korišćeni u ovom projektu. Simboli 1 i 7 predstavljaju violinski, odnosno bas ključ koji se postavlja na početak zapisa. Simboli 2 i 8 predstavljaju trajanje takta, koje je u ovom primeru 3/4. Simboli 3 i 9 predstavljaju notu čije je trajanje 1/4, dok simboli 5 i 11 predstavljaju notu čije je trajanje 1/8. Ostali simboli predstavljaju pauze, tj. 4 i 10 predstavljaju pauzu čije je trajanje 1/4, dok 6 i 12 predstavljaju pauzu čije je trajanje 1/8. U ovom projektu koriste se samo trajanja 1/4 i 1/8.



Slika 3. Izgled simbola

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. U oba slučaja obezbediti da se interakcija sa korisnikom odvija putem statičkih ekrana. Interakcija u slučaju grafičkog interfejsa može da se vrši putem tastature ili miša. U zavisnosti od izabrane opcije i njenih parametara, program izvršava zadatak 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

Prvi argument komandne linije prilikom pokretanja programa predstavlja putanju do fajla sa potrebnim mapiranjima karaktera na note i MIDI brojeve. U svakom redu datoteke nalaze se podaci za po jedan ulazni karakter i to: karakter koji se opisuje, tekstualni opis note i MIDI broj note, odvojeni znakom zareza, kao u navedenom primeru, koji opisuje karakter 't' koji se mapira na notu C4 čiji je MIDI broj 60:

t,C4,60

Korisnik odabirom ponuđene opcije i unosom putanje do ulaznog fajla može zahtevati učitavanje kompozicije. Ulazni fajl se sastoji od sekvence karaktera koji opisuju muzičke simbole. Može sadržati prelaze u novi red, ali njih treba ignorisati. Dekodovanje ulaznog fajla i pretvaranje karaktera u muzičke simbole se vrši uz pomoć mapiranja koje se učitava po pokretanju programa. Na osnovu mapiranja se određuje visina i oktava u kojoj se nota nalazi. Prilikom parsiranja fajla pauze i trajanja nota se određuju prema sledećim pravilima:

- blanko znak označava pauzu u trajanju 1/8
- znak | između simbola označava pauzu u trajanju 1/4
- trajanje nota navedenih unutar uglastih zagrada sa znakom razmaka između svake note je 1/8
- navođenje nota unutar uglastih zagrada bez znaka razmaka između njih označava da sve note treba odsvirati u istom trenutku i sve note su trajanja 1/4
- trajanje nota navedenih van zagrada je 1/4

U nastavku je dat primer dela ulaznog fajla, a u tabeli ispod i tumačenje svakog od simbola:

[Yu] t / Yu [t u Y]

Ulazni simboli	Simboli koje označavaju	Trajanje simbola
[Yu]	note D#4 i E4	1/4 (sviraju se istovremeno)
_ (blanko znak)	pauza	1/8
t	nota C4	1/4
	pauza	1/4
Yu	note D#4 i E4	1/4 (sviraju se jedna za drugom)
_ (blanko znak)	pauza	1/8
[t u Y]	note C4, E4 i D#4	1/8 (sviraju se jedna za drugom)

Prilikom učitavanja korisnik pored putanje do fajla kompozicije unosi i željenu dužinu svih taktova od kojih će se kompozicija sastojati. Dužina taktova se zadaje u vidu razlomka čiji imenilac mora biti ili 4 ili 8. Postupak parsiranja se zatim svodi na učitavanje muzičkih simbola iz ulazne datoteke i kreiranje taktova kompozicije. Ukoliko je oktava učitaniog simbola veća od 3 simbol se smešta u linijski sistem kompozicije za desnu ruku, dok se u suprotnom smešta u linijski sistem za levu ruku. Ukoliko se radi o sekvenci simbola koji se sviraju odjednom simboli se razvrstavaju na linijske sisteme leve i desne ruke prema svojoj oktavi. Ako je potrebno da simbol svira samo jedna ruka, u linijski sistem za drugu ruku se ubacuje pauza istog trajanja kao i simbol koji se svira. Pauza kao ulazni simbol se smešta u oba linijska sistema.

Problem koji se može pojaviti kod ovakvog kreiranja kompozicije jeste da u trenutku kada je u taktu ostalo mesta za simbol trajanja 1/8 naiđe sledeći simbol koji je potrebno smestiti trajanja 1/4. Takve situacije rešavati tako što se dati simbol podeli na dva jednaka simbola trajanja 1/8 od čega se prvi smesti u tekući takt, a drugi u naredni takt. Primer: korisnik za kompoziciju traži da trajanje takta bude 3/8. Parsira se nota trajanja 1/4. Nota se regularno ubacuje u takt. Parsira se nova nota trajanja 1/4. Tekući takt još uvek nije popunjen, ali dodavanje tekuće note prevazilazi njegovo trajanje. Nota se deli na dve jednake note trajanja 1/8 i prva se smešta u tekući takt, a druga započinje novi takt. Note dobijene na ovaj način potrebno je posebno obeležiti, jer prilikom eksportovanja kompozicije ne treba koristiti dve odvojene note, već ovo posmatrati kao jednu notu čije je trajanje njihov zbir. Detaljniji opis upotrebe ovakvih nota dat je u nastavku dokumenta u poglavljima o manipulaciji i eksportovanju kompozicije. Poslednji takt kompozicije dopuniti pauzama do kraja.

Prilikom učitavanja podataka o kompozicijama i notama **koristiti regularne izraze** za parsiranje datih fajlova. Predvideti način za oporavak od grešaka u slučaju neispravnog formata datoteke ili nepostojanja tražene datoteke.

Osnovna manipulacija nad kompozicijom

Korisniku je po učitavanju kompozicije potrebno omogućiti sledeće:

- Ispisivanje podataka o kompoziciji:
Ispisivanje je potrebno izvršiti prema linijskom sistemu koji je prikazan na slici 3, tj. ispisuju se redom muzički simboli u po dva reda (u gornjem redu simboli koji se sviraju desnom rukom, a u donjem redu simboli koji se sviraju levom rukom). Parovi redova se odvajaju praznim redom. Taktovi u okviru kompozicije se odvajaju znakom |. Takt koji svira leva i takt koji svira desna ruka moraju biti ispisani na istoj širini. Jedan takt se obavezno ispisuje u jednom redu – nije moguće prelamanje takta u više redova. Tekstualni opis note treba da sadrži jednoslovnu visinu i jednocifrenu vrednost oktave. Povišene note iza visine imaju znak #. Note trajanja 1/4 pisati velikim slovima, a note trajanja 1/8 malim slovima. Za sve što nije naznačeno u ovom opisu usvojiti razumnu pretpostavku.
- Iteriranje kroz kompoziciju:
Korisniku je potrebno omogućiti kretanje kroz taktove i note kompozicije. Iteriranje kreće od prvog takta kompozicije. Korisnik prilikom iteriranja kroz taktove ima sledeće mogućnosti:
 - prelazak na sledeći takt
 - povratak na prethodni takt
 - započinjanje iteriranja kroz note takta

Prilikom iteriranja kroz note jednog takta korisnik ima sledeće mogućnosti:

- prelazak na sledeću notu
- povratak na prethodnu notu
- menjanje oktave ili visine note
- dodavanje ili uklanjanje povisilice

U svakom koraku korisniku prikazati tekstualni opis takta/note na kojem/kojoj se nalazi, kao i redni broj takta u kompoziciji/note u taktu.

- Izmena takta kompozicije:
Korisniku je potrebno omogućiti da promeni trajanje taktova kompozicije. Nakon izmene potrebno je sve postojeće muzičke simbole ponovo rasporediti po taktovima prema pravilima koja su važila i prilikom njihovog inicijalnog dodavanja.
- Pomeranje kompozicije:
Korisniku je potrebno omogućiti da celu kompoziciju pomeri za zadati broj oktava naviše ili naniže, što se izvršava tako što se svakoj noti broj oktave uveća ili umanja za dati broj.

Eksportovanje kompozicije

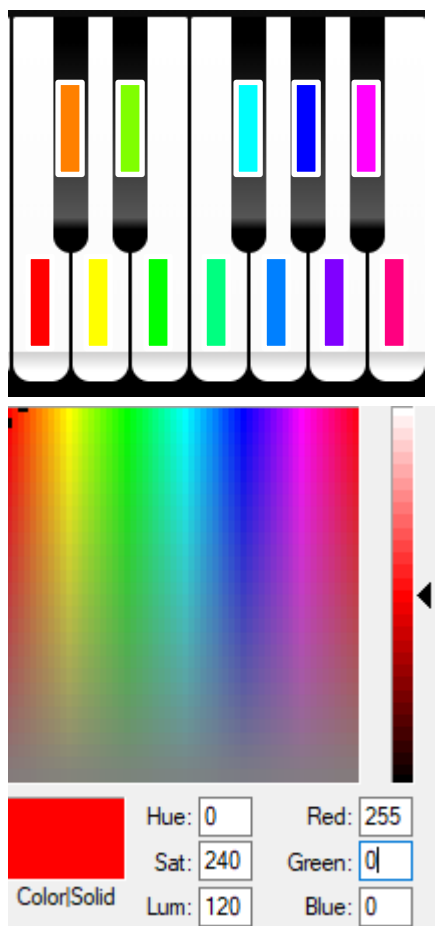
Potrebno je korisniku omogućiti eksportovanje kompozicije u tri različita formata. Prilikom eksportovanja korisnik zadaje željeni format, kao i putanju do fajla u koji je potrebno smestiti rezultat eksportovanja. Podržati sledeće formate:

- MusicXML – standardni XML format za digitalni zapis kompozicija
- MIDI – audio format zapisa kompozicije
- BMP – format zapisa slike

Specifikacije sva tri formata date su u prilogu ovog dokumenta.

Za pisanje MIDI fajlova može se koristiti Midifile biblioteka koju je moguće preuzeti na sledećem linku: <https://github.com/craigsapp/midifile>. Primer upotrebe biblioteke dat je uz specifikaciju formata fajla.

Eksportovanje kompozicije u vidu BMP formata predstavlja njenu konverziju u sliku. Konverzija se vrši mapiranjem muzičkih simbola na odgovarajuće boje u RGB formatu, prema načinu koji je opisan u nastavku.



Slika 4. Mapiranje tonova na boje

Na slici 4. dato je mapiranje tonova na boje. RGB paleta boja je podeljena na 12 jednakih delova, upravo kao što i jedna oktava ima 12 tonova. U tabeli 1. je prikazano dato mapiranje tonova na boje u RGB formatu.

Ton	Boja		Crvena	Zelena	Plava
C4	Crvena		255	0	0
C#4	Narandžasta		255	127	0
D4	Žuta		255	255	0
D#4	Šartruz		127	255	0
E4	Zelena		0	255	0
F4	Prolećnozelena		0	255	127
F#4	Cijan		0	255	255
G4	Azurna		0	127	255
G#4	Plava		0	0	255
A4	Ljubičasta		127	0	255
A#4	Ciklama		255	0	255
B4	Roze		255	0	127

Tabela 1. Mapiranje tonova na boje

Dato mapiranje određuje boju tona u središnjoj oktavi (oktava sa brojem 4). Tonovi u nižim oktavama (oktave sa manjim brojem) koriste tamnije boje, dok tonovi u višim oktavama (oktave sa većim brojem) koriste svetlije boje. U nastavku su date formule na osnovu kojih se računaju RGB komponente boja tona po oktavama:

$$x_2 = x_4 - \frac{x_4}{8} * 6$$

$$x_3 = x_4 - \frac{x_4}{8} * 3$$

$$x_5 = x_4 + \frac{255 - x_4}{8} * 3$$

$$x_6 = x_4 + \frac{255 - x_4}{8} * 6$$

U navedenim formulama x_4 predstavlja vrednost komponente boje u četvrtoj oktavi, koja je poznata iz tabele 1. U tabeli 2. predstavljena su data mapiranja za tonove C i G kroz svih 5 oktava.

	Oktava 2			Oktava 3			Oktava 4			Oktava 5			Oktava 6		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
Ton C	63	0	0	159	0	0	255	0	0	255	95	95	255	191	191
	C2			C3			C4			C5			C6		
Ton G	0	31	63	0	79	159	0	127	255	95	175	255	191	223	255
	G2			G3			G4			G5			G6		

Tabela 2. Boje tonova C i G po oktavama

Pauza se predstavlja belom bojom koja u RGB formatu ima sve tri komponente postavljene na 255.

Kreiranje slike na osnovu kompozicije se na osnovu izloženog vrši tako što se od korisnika traži da unese širinu slike u broju piksela, a zatim se za svaki muzički simbol vrši dodavanje određenog broja piksela s leva u desno, odozgo na dole. Boja piksela se određuje na osnovu vrste muzičkog simbola na način koji je izložen. Za muzičke simbole koji se sviraju istovremeno samo jednom se generiše piksel, a njegova boja se dobija kao kombinacija svih boja muzičkih simbola koji učestvuju u njegovom kreiranju. Kombinacija boja se dobija tako što se uzme aritmetička sredina svih pojedinačnih komponenti. Broj piksela zavisi od trajanja simbola. Za simbole trajanja 1/8 dodaje se 1 piksel, dok se za simbole trajanja 1/4 dodaju dva piksela iste boje. Poslednji red slike dopuniti belim pikselima.

Kraj rada

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

Testiranje rada programa

Test primere za testiranje rada realizovanog softverskog rešenja moguće je naći na sajtu: <https://virtualpiano.net/>. Sajt omogućava sviranje virtuelnog klavira pomoću tastature računara, prema istim pravilima koja važe i u ovom projektu. Za veliki broj kompozicija obezbeđeni su skupovi karaktera koje je potrebno pritiskati, koji ujedno predstavljaju i ulazne fajlove ovog softvera, a moguće ih je pronaći pod opcijom "Music sheets".

Rezultat rada programa, odnosno korektnost generisanih fajlova je moguće proveriti alatima za vizuelizaciju i reprodukciju.

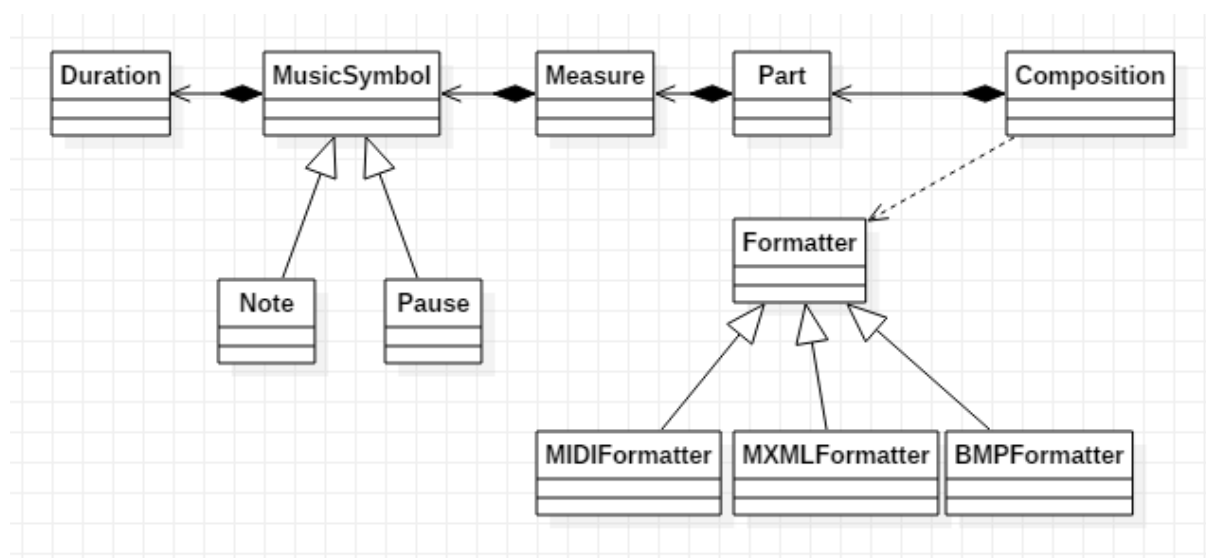
Korektnost MusicXML fajla može se proveriti pomoću online alata za vizuelizaciju ovog formata: <https://www.soundslice.com/musicxml-viewer/>. Potrebno je samo učitati fajl klikom na dugme "Choose file" da bi se dobila vizuelna reprezentacija izabranog fajla. Pomoću ovog alata moguće je čak i reprodukovati fajl.

MIDI fajl može se proveriti pomoću velikog broja audio alata, među kojima je i Windows Media Player, koji dolazi instaliran uz Windows operativni sistem i podržava puštanje midi fajlova.

BMP fajl može se proveriti pomoću alata za obradu slika, među kojima je i Paint, koji, takođe, dolazi uz Windows operativni sistem.

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. Primiti da postoje tri različita formata izlaznih fajlova.

Specifikacija MusicXML formata

MusicXML format predstavlja format za digitalni zapis muzičkih kompozicija. Kompletnu specifikaciju ovog formata moguće je naći na sledećoj adresi: <https://www.musicxml.com/>, a u nastavku su dati opisi delova ovog formata koji su potrebni za izradu projekta, kao i primeri svih simbola.

Svaki MusicXML fajl je sledeće strukture:

```
<?xml version="1.0" encoding="UTF-8"
      standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
      "-//Recordare//DTD MusicXML 3.1 Partwise//EN"
      "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="Right"></score-part>
    <score-part id="Left"></score-part>
  </part-list>
  <part id="Right">
    ...
  </part>
  ...
  <part id="Left">
    ...
  </part>
</score-partwise>
```

U navedenom delu fajla definisana su dva linijska sistema za desnu i levu ruku. Svaki od part tagova je potrebno popuniti taktovima koji se sviraju odgovarajućom rukom. Jedan takt je opisan **measure** tagom. Prvi takt uvek počinje specifikacijom trajanja (vrste, mere) takta i vrste ključa, te će prvi takt desne ruke uvek početi na način koji je prikazan u nastavku sa leve strane, a prvi takt leve ruke na način koji je prikazan sa desne strane:

```
<measure>
  <attributes>
    <divisions>2</divisions>
    <time>
      <beats>3</beats>
      <beat-type>8</beat-type>
    </time>
    <clef>
      <sign>G</sign>
      <line>2</line>
    </clef>
  </attributes>
```

```
<measure>
  <attributes>
    <divisions>2</divisions>
    <time>
      <beats>3</beats>
      <beat-type>8</beat-type>
    </time>
    <clef>
      <sign>F</sign>
      <line>4</line>
    </clef>
  </attributes>
```

Jedino što je potrebno promeniti jeste trajanje takta. U navedenom primeru je 3/8, te je samo tu vrednost potrebno prilagoditi.

Svi muzički simboli koji se dalje nalaze u taktu opisani su tagom **note**. U okviru note taga se mogu nalaziti sledeći tagovi:

- **duration** – određuje trajanje simbola (za 1/8 je vrednost 1, a za 1/4 je vrednost 2)
- **pitch** – određuje oktavu i visinu note (oktava se postavlja u **octave** tagu, a visina u **step** tagu). U slučaju povišene note doda se **alter** tag sa vrednošću 1
- u slučaju da se više nota svira odjednom sve one se navode jedna za drugom u **note** tagovima, a sve sem prve moraju imati i prazan **chord** tag da se naglasi da se radi o akordu.
- za pause se samo postavi prazan **rest** tag u okviru **note** taga.
- u slučaju kada se jedna nota trajanja 1/4 prelomi na 2 note trajanja 1/8 u dva različita takta potrebno je te dve note spojiti lukom. Luk se pravi tako što se u prvoj noti stavi tag **tie** sa atributom **type** postavljenim na **start**, a u drugoj noti isti tag sa vrednošću **type** atributa **end**.

```
<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
<!DOCTYPE score-partwise
PUBLIC
"-//Recordare//DTD
MusicXML 3.1 Partwise//EN"
```



```
"http://www.musicxml.org/dtds/
partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="Right"></score-
part>
    <score-part id="Left"></score-
part>
  </part-list>
  <part id="Right">
    <measure>
      <attributes>
        <divisions>2</divisions>
        <time>
          <beats>3</beats>
          <beat-type>8</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
        </attributes>
        <note>
          <pitch>
            <step>C</step>
            <octave>4</octave>
          </pitch>
          <duration>2</duration>
        </note>
        <note>
          <chord/>
          <pitch>
            <step>E</step>
            <octave>4</octave>
          </pitch>
          <duration>2</duration>
```

```
</note>
<note>
  <pitch>
    <step>E</step>
    <octave>4</octave>
  </pitch>
  <duration>1</duration>
  <tie type="start"/>
</note>
</measure>
<measure>
  <note>
    <pitch>
      <step>E</step>
      <octave>4</octave>
    </pitch>
    <duration>1</duration>
    <tie type="end"/>
  </note>
  <note>
    <pitch>
      <step>F</step>
      <octave>4</octave>
      <alter>1</alter>
    </pitch>
    <duration>2</duration>
  </note>
</measure>
</part>
<part id="Left">
  <measure>
    <attributes>
      <divisions>2</divisions>
      <time>
        <beats>3</beats>
```

```
<beat-type>8</beat-type>
</time>
<clef>
  <sign>F</sign>
  <line>4</line>
</clef>
</attributes>
<note>
  <pitch>
    <step>C</step>
    <octave>3</octave>
  </pitch>
  <duration>2</duration>
</note>
<note>
  <rest/>
  <duration>1</duration>
</note>
</measure>
<measure>
  <note>
    <pitch>
      <step>E</step>
      <octave>3</octave>
    </pitch>
    <duration>1</duration>
  </note>
  <note>
    <rest/>
    <duration>2</duration>
  </note>
</measure>
</part>
</score-partwise>
```

Specifikacija MIDI formata

Kompletnu specifikaciju MIDI formata fajla moguće je pronaći na sledećoj adresi:
<http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html>

Za potrebe projekta potrebno je iskoristiti biblioteku koja se nalazi na sledećoj adresi:
<https://github.com/craigsapp/midifile>

U folderu include datog github repozitorijuma nalaze se potrebni .h fajlovi, a u src folderu potrebni .cpp fajlovi. Primeri upotrebe biblioteke nalaze se u tools folderu. Na sledećoj strani dokumenta je dat kratak primer koji generiše midi fajl za jednu poznatu kompoziciju (preuzeto iz createmidifile.cpp).

Nizovi **melody** i **bass** sadrže MIDI brojeve nota koje je potrebno odsvirati, a koji se dobijaju na osnovu mapiranja koje je opisanu o ovom dokumentu, dok nizovi **mrhythm** i **brhythm** određuju trajanje (vrednost jedan označava trajanje 1/8, a vrednost 2 trajanje 1/4). Pravljenje MIDI fajla se zasniva na dodavanju događaja (**event**), a to se sprovodi sledećim pozivom:

```
outputfile.addEvent(0, actiontime, midievent);
```

U navedenom pozivu prvi argument će označavati da li se radi o levoj ili desnoj ruci (0 – desna, 1 – leva). Drugi argument će označavati trenutak u kome se događaj dešava (umnožak $\frac{tpq}{2}$). Kao treći argument će se prosleđivati vektor koji u sebi sadrži tip događaja (na indeksu 0; koristimo samo 0x90 i 0x80 koji predstavljaju pritisak i otpuštanje dirke klavira za zadatu notu) i broj note koja se svira (na indeksu 1). Note koje se sviraju odjednom će imati isit actionTime. Pauza se ne evidentira kao događaj već se samo uveća actionTime.

U navedenom primeru su napravljene dve odvojene while petlje od kojih jedna evidentira note za desnu (prilikom poziva addEvent uvek kao prvi argument prosleđuje 0), a druga za levu ruku (prilikom poziva addEvent uvek kao prvi argument prosleđuje 1). Preporuka je da se tako realizuje i u projektu, ali je, svakako, moguće realizovati i na drugi način.

```
#include "MidiFile.h"
#include <iostream>

using namespace std;
using namespace smf;

int main(int argc, char** argv) {
    MidiFile outputfile;
    outputfile.absoluteTicks();
    vector<uchar> midievent;
    midievent.resize(3);
    int tpq = 48;
    outputfile.setTicksPerQuarterNote(tpq);
    outputfile.addTrack(1);

    int melody[50] =
{72,72,79,79,81,81,79,77,77,76,76,74,74,72,-1};
    int mrhythm[50] = {1,1,1,1,1,1,2,1,1,1,1,1,1,2,-1};

    int bass[50] =
{48,60,64,60,65,60,64,60,62,59,60,57,53,55,48,-1};
    int brhythm[50]={1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,-1};

    int i=0;
    int actiontime = 0;
    midievent[2] = 64;
    while (melody[i] >= 0) {
        midievent[0] = 0x90;
        midievent[1] = melody[i];
        outputfile.addEvent(0, actiontime, midievent);
        actiontime += tpq/2 * mrhythm[i];
        midievent[0] = 0x80;
        outputfile.addEvent(0, actiontime, midievent);
        i++;
    }
    i = 0;
    actiontime = 0;
    while (bass[i] >= 0) {
        midievent[0] = 0x90;
        midievent[1] = bass[i];
        outputfile.addEvent(1, actiontime, midievent);
        actiontime += tpq/2 * brhythm[i];
        midievent[0] = 0x80;
        outputfile.addEvent(1, actiontime, midievent);
        i++;
    }

    outputfile.sortTracks();
    outputfile.write("twinkle.mid");
    return 0;
}
```

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 8 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 8 B

U odnosu na navedeni primer, prilikom generisanja BMP fajla, 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. Svaki red slike potrebno je dopuniti nulama kako bi ukupan broj bajtova za jedan red slike bio deljiv sa 8.