

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2029

**Oblikovanje i implementacija programskog
produkta za prikaz elektroencefalografskih
signala**

Lea Suć

Zagreb, lipanj 2011.

Sadržaj

1. Uvod	1
2. Snimanje i pohrana EEG-a.....	3
2.1 Elektroencefalografija.....	3
2.2 Međunarodni sustav 10-20	4
2.3 EDF format [7]	6
2.3.1 Specifikacija EDF formata	7
2.3.2 Specifikacija EDF+ formata [8]	9
3. Zahtjevi postavljeni na programski produkt.....	11
4. Arhitektura i izgradnja programskog produkta	15
4.1 Rukovanje podacima iz EDF datoteke.....	15
4.2 Implementacija grafičkog korisničkog sučelja.....	17
4.3 Vizualizacija signala.....	22
4.3.1 Određivanje veličine prostora za iscrtavanje	22
4.3.2 Iscrtavanje signala u ovisnosti o parametrima	23
4.3.3 Iscrtavanje vrijednosti s pozicijom neovisnom o pomicanju	24
5. Opis i primjeri korištenja programskog produkta	25
5.1 Opcije za prikaz signala	26
5.2 Dodatne opcije	30
5.2.1 Ispis uzoraka u tekstualnu datoteku	30
5.2.2 Prikaz bitnih informacija iz zaglavlja	32
6. Zaključak.....	34
7. Literatura.....	35
Oblikovanje i implementacija programskog produkta za prikaz elektroencefalografskih signala	36
Sažetak.....	36
Ključne riječi.....	36
Design and implementation of application for displaying electroencephalographic signals	37
Abstract.....	37
Keywords	37

1. Uvod

Ljudski mozak je vrlo složen i sastoji se od ogromnog broja živčanih stanica (neurona), njih oko 10^{11} , pri čemu je svaki od njih povezan s oko 10^4 drugih [1].

Prijenos informacija kroz mozak je elektrokemijske prirode. Električni potencijal koji stvara pojedinačni neuron ima izrazito malu amplitudu, no sinkroniziranim djelovanjem čitave mreže neurona stvara se potencijal dovoljno velik da se može mjeriti elektrodama na površini glave [2].

Neinvazivna metoda kojom se pomoću elektroda smještenih na glavi registrira moždana električna aktivnost naziva se elektroencefalografija. Razlika u naponu između dvije elektrode može se izmjeriti voltmetrom, a zapis napona tijekom vremena rezultira elektroencefalogramom (EEG) [1].

Promjene električnog potencijala membrane neurona moguće su zbog ionske propusnosti membrane. Djelovanje podražaja na neuron najprije ima utjecaj na otvaranje natrijevih kanala i ulaska naglo pozitivno nabijenih Na^+ iona u neuron, pri čemu dolazi do promjene električnog naboja i stvaranja nove ravnoteže unutar milisekunde, odnosno depolariziranja neurona. U trenu dolazi do porasta transmembranskog napona za 100 mV, zbog čega se diferencijalni napon na unutarnjoj strani stanice prebaci od negativnog (otprilike -70 mV) na pozitivnu vrijednost (otprilike +30 mV). Zbog naglog porasta otvaraju se kalijevi kanali, što dovodi do naglog izlaska K^+ iona, pa se napon uskoro vraća na izvorno negativnu vrijednost s unutarnje strane membrane. Kompletan proces polarizacije i depolarizacije naziva se akcijski potencijal. Živčana vlakna su poput vodiča, te akcijski potencijal koji je stvoren na jednom mjestu stvara gradijent napona između aktivnog dijela membrane i dijelova membrane u mirovanju. Pritom vrijedi princip "sve ili ništa", odnosno veličina akcijskog potencijala će uvijek biti jednaka, jedina je značajka da će se on ili pojaviti (ako je podražaj veći od praga podražljivosti), ili neće uopće. Na opisani način akcijski potencijal putuje aksonom (dugačkim vlaknom koje

prenosi električne signale) od jednog neurona do drugog, koji ga prima pomoću dendrita i zatim pomoću aksona opet šalje prema drugim neuronima, tako da neuroni djeluju kao prijenosnici informacija [3].

Elektroencefalogram je nekad bio iscrtavan isključivo na papiru, dok je danas češće zastupljen u obliku digitalnog prikaza korištenjem računala. Kod dijagnoze moždane aktivnosti na temelju elektroencefalograma javlja se potreba za detaljnijom vizualizacijom zabilježenih signala.

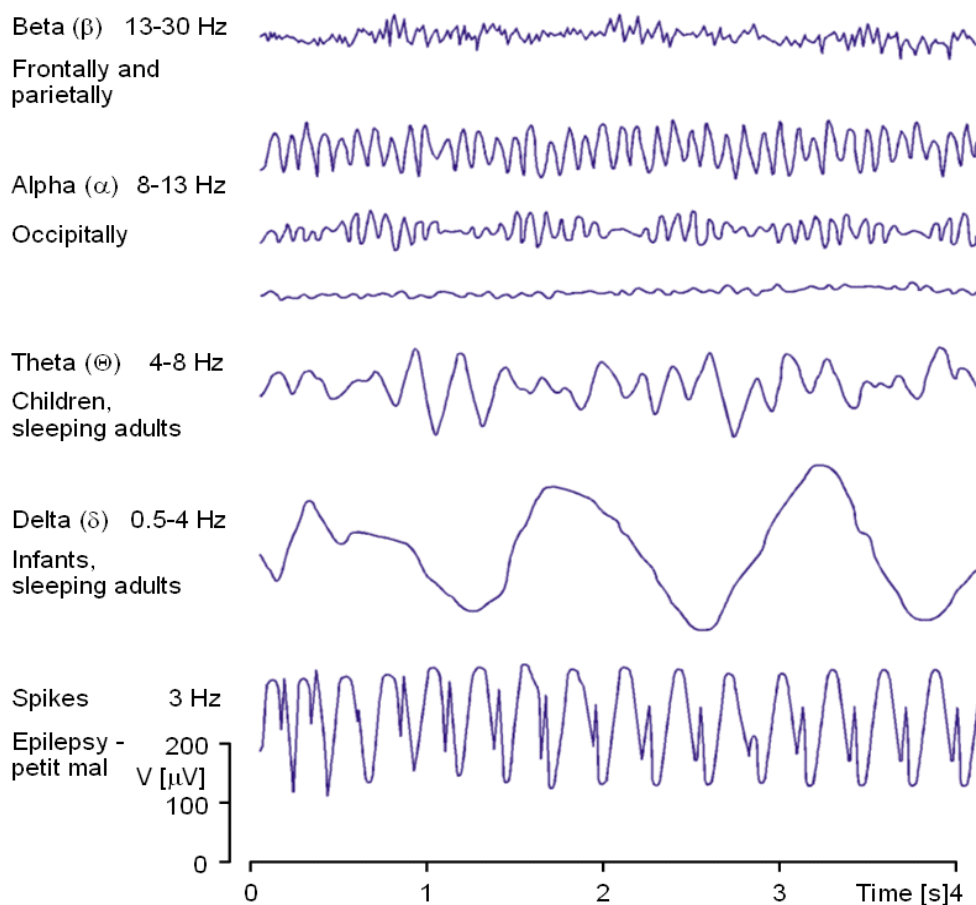
Cilj ovog rada je izrada programskog produkta koji omogućava učitavanje i detaljniji prikaz EEG signala u ovisnosti o različitim parametrima, kako bi se omogućila njihova lakša obrada i usporedba. U nastavku su opisane osnove elektroencefalografije i njezin značaj, karakteristični signali dobiveni mjerenjem moždane električne aktivnosti, međunarodni sustav 10-20 kojim je uređen raspored elektroda, a čitatelj može naučiti ponešto i o građi i radu ljudskog mozga, a isto tako i o programskom jeziku Java. Također, detaljno je objašnjen normirani format za pohranu EEG signala, a ukratko je opisana i naprednija inačica tog formata (odnosno razlike među njima).

2. Snimanje i pohrana EEG-a

2.1 Elektroencefalografija

Za prvo snimanje električnog polja ljudskog mozga i naziv elektroencefalogram zaslužan je njemački psihijatar Hans Berger 1924. Od 1929. do 1938. objavio je 20 znanstvenih radova o EEG-u pod nazivom "Über das Elektroenkephalogramm des Menschen" [4].

Analiza moždanih signala je vrlo zahtjevna zbog velike količine informacija prikupljenih pomoću elektroda na površini glave, zbog čega je ključno odrediti koji su signali bitni za analizu. EEG signale možemo razvrstati u nekoliko grupa prema različitim kriterijima, primjerice prema frekvencijskim područjima ili obliku. Na slici 2.1 prikazani su neki karakteristični signali koji se mogu uočiti u EEG-u – alpha (α), beta (β), delta (δ), i theta (Θ), kao i šiljci povezani s epilepsijom [2].



Slika 2.1. Primjeri EEG valova [2]

Alpha valovi imaju frekvenciju između 8 i 13 Hz i mogu se izmjeriti u zatiljnom režnju osobe u budnom stanju sa zatvorenim očima. Beta valovi imaju frekvenciju između 13 i 30 Hz i mjerljivi su u čeonom i tjemenom režnju. Delta valovi imaju raspon frekvencije od 0.5 do 4 Hz, a mogu se izmjeriti kod male djece i odraslih osoba u stanju sna. Theta valovi imaju raspon 4-8 Hz i mjere se kod djece i odraslih osoba u stanju sna.

Kod mjerenja spontane aktivnosti (aktivnost koja se kontinuirano odvija u mozgu pojedinca), amplituda EEG-a je oko 100 mikrovolta kad se mjeri na glavi, a oko 1-2 mV na površini mozga, a raspon frekvencije od 1 Hz do otprilike 50 Hz. Na EEG-u se mogu promatrati i evocirani potencijali, koji nastaju kao odgovor mozga na određeni podražaj (primjerice vidni, slušni).

Kako mozak nikad ne miruje, aktivnost neurona u mozgu je moguće konstantno pratiti. Elektroencefalografija je metoda koja se koristi dugi niz godina, tako da su obilježja EEG-a vezana uz određene fiziološke obrasce ponašanja dobro poznata. Snimanje EEG-a može otkriti abnormalnosti u funkcioniranju mozga pa je stoga EEG važan alat u dijagnostici brojnih bolesti [5].

Naime, pojedina normalna (budnost, spavanje, drijemanje) i patološka stanja (epilepsija, encefalopatija, moždani udar i drugo) imaju svoj specifičan EEG zapis. Dakle, za svaku razinu svijesti i za svaki poremećaj postoje karakteristični EEG signali. Zbog tog je jedna od najčešćih primjena elektroencefalografije u dijagnosticiranju epilepsije i moždane smrti, ali je isto tako koristan kod tumora, potresa, povreda mozga, gubitka pamćenja, glavobolja, vrtoglavica i poremećaja spavanja. EEG se koristi za ispitivanje kako funkcionira mozak zdravih osoba pri određenim aktivnostima, poput meditacije ili učenja, i mnogih drugih istraživanja (primjerice razlike među spolovima).

2.2 Međunarodni sustav 10-20

Današnje moderne tehnike za snimanje EEG signala prikupljaju informacije o električnom potencijalu na površini glave, te ih digitaliziraju i spremaju za daljnju obradu. Elektrode snimaju napone veličine nekoliko mikrovolti, koje propuštaju kroz pojačalo. Korištenje ove tehnologije ovisi strogo o poziciji postavljenih elektroda i

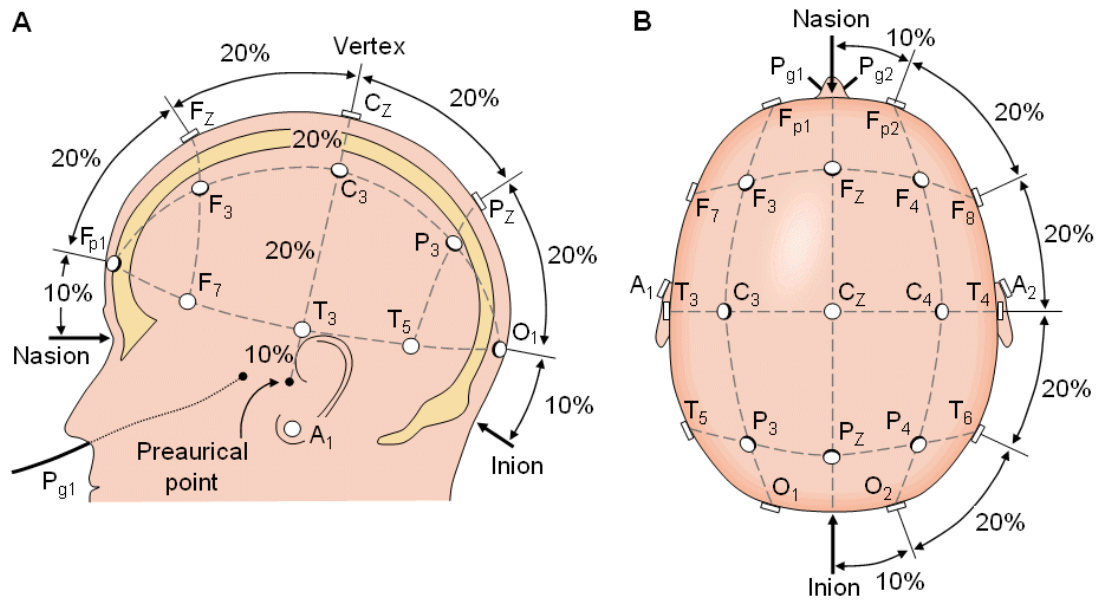
njihovom kontaktu. Za vrijeme ili nakon mjerenja potrebno je signal obraditi. Smetnje se mogu izbjeći ili filtrirati [5].

Da bi se omogućila usporedba dobivenih rezultata između različitih laboratorija potrebno je dobro definirati pozicije prema kojima su elektrode razmještene na glavi [2]. Pritom se raspored postavljanja i nomenklatura elektroda najčešće odabiru prema normi 10-20.

Sustav 10-20 sastoji se od 21 elektrode razmještene po unaprijed definiranim pozicijama, pri čemu se "10" i "20" odnose na razmak elektroda, koji je ili 10% ili 20% ukupne udaljenosti lubanje, uzimajući na taj način u obzir varijacije u veličini glave (Slika 2.2) [6].

Sustav koristi specifičnu nomenklaturu za označavanje pozicija pojedinih elektroda s obzirom na građu mozga. Mozak je podijeljen na dvije polovice, odnosno lijevu i desnu hemisferu. Svaka od njih je podijeljena na čeonu (engl. *frontal*), tjemeni (engl. *parietal*), sljepoočni (engl. *temporal*) i zatiljni (engl. *occipital*) režanj [2].

Za svaku elektrodu postoji slovo koje identificira režanj i broj koji određuje lokaciju na polutki. Slova F, T, C, P i O se odnose na čeonu, sljepoočnu, „središnju“ (engl. *central*), tjemeni i zatiljni režanj, respektivno. Pritom je potrebno naglasiti da središnji režanj zapravo ne postoji, već se slovo „C“ koristi samo u identifikacijske svrhe. Oznaka „z“ (engl. *zero*) odnosi se na elektrode smještene na središnjoj liniji. Parni brojevi (2, 4, 6, 8) se odnose na pozicije elektroda na desnoj hemisferi, dok se neparni brojevi (1, 3, 5, 7) odnose na one na lijevoj hemisferi mozga.



Slika 2.2. Razmještaj 21 elektrode prema sustavu 10-20 s lijeve strane (A) i odozgo (B) [2]

EEG signali se mogu mjeriti monopolarno ili bipolarno [5]. Kod monopolarnog snimanja postoji aktivna elektroda koja se postavlja na područje od interesa i referentna elektroda koja se postavlja na neko područje za koje se pretpostavlja da nije zahvaćeno aktivnošću povezanom sa zadatkom. Kod bipolarnog snimanja snimljeni signal predstavlja razliku signala snimljenih na dvije aktivne elektrode. Tako primjerice „Fp1-F3“ predstavlja razliku u potencijalu između elektrode Fp1 i elektrode F3. Sljedeći kanal „F3-C3“ predstavlja razliku u potencijalu između F3 i C3, i analogno tome se označava za sve elektrode.

2.3 EDF format [7]

EDF format (engl. *European Data Format*) je uobičajeni format za pohranu i razmjenu višekanalnih bioloških i fizikalnih signala. Signali pritom mogu imati bilo koje, pa čak i različite fizikalne veličine, kao i frekvencije uzorkovanja. Razvilo ga je nekoliko europskih inženjera medicine koji su se prvi put sastali 1987., kad se održavao tzv. *Sleep congress* u Kopenhagenu. Inženjeri su od 1989. do 1992. radili na projektu tzv. *Sleep-Wake* analize te su željeli primijeniti svoje algoritme za analizu spavanja na međusobne podatke i usporediti rezultate analiza. U ožujku

1990. su se u Leidenu dogovorili oko vrlo jednostavnog zajedničkog formata podataka, koji je postao poznat kao EDF.

Opis EDF formata je objavljen 1992. u časopisu „Electroencephalography and Clinical Neurophysiology“ [7]. Specifikacija se od tad nije mijenjala i još uvijek ima verziju 0. Danas se široko koristi u različitim istraživačkim projektima.

2.3.1 Specifikacija EDF formata

Jedna EDF datoteka sadrži jedan neprekinuto digitalizirani poligrafski zapis. Sastoji se od zaglavlja (engl. *header*) iza kojeg slijede zapisi podataka (engl. *data record*). Zapis u zaglavlju, koje je promjenjive duljine, sadrži informacije o pacijentu i specificira tehničke karakteristike snimljenih signala. Zapisi podataka sadrže uzastopne epohe određenog trajanja poligrafskih snimanja.

Prvih 256 bajtova zaglavlja specificira broj verzije formata, identifikaciju pacijenta i snimanja, informaciju o vremenu i datumu početka snimanja, broj i trajanje zapisa podataka, te broj signala (n_s) u svakom zapisu podataka (Slika 2.3). Zatim slijedi 256 bajtova za svaki signal, koji specificiraju relevantne podatke o pojedinom signalu, kao što su tip signala (e.g. EEG, body temperature, etc.), podaci o amplitudi, podaci o broju uzoraka u zapisu podataka, iz čega se može izvesti frekvencija uzorkovanja, jer je poznato trajanje svakog zapisa podataka. Na ovaj način EDF format dozvoljava različite frekvencije uzorkovanja i različite oznake fizikalne veličine za svaki signal. Prema tome, zapisa zaglavlja sadrži $256 + (n_s * 256)$ bajtova. U nastavku je opisano nekoliko pravila vezanih uz podatke u zaglavlju.

8 ascii version of this data format (0)
80 ascii local patient identification
80 ascii local recording identification
8 ascii startdate of recording (dd.mm.yy)
8 ascii starttime of recording (hh.mm.ss)
8 ascii number of bytes in header record
44 ascii reserved
8 ascii number of data records (-1 if unknown)
8 ascii duration of a data record, in seconds
4 ascii number of signals (ns) in data record
ns * 16 ascii ns * label (e.g. EEG FpzCz or Body temp)
ns * 80 ascii ns * transducer type (e.g. AgAgCl electrode)
ns * 8 ascii ns * physical dimension(e.g. μ V or degree C)
ns * 8 ascii ns * physical minimum (e.g. -500 or 34)
ns * 8 ascii ns * physical maximum (e.g. 500 or 40)
ns * 8 ascii ns * digital minimum (e.g. -2048)
ns * 8 ascii ns * digital maximum (e.g. 2047)
ns * 80 ascii ns * prefiltering (e.g. HP:0.1Hz LP:75Hz)
ns * 8 ascii ns * nr of samples in each data record
ns * 32 ascii ns * reserved

Slika 2.3. Detaljan digitalni format zapisa zaglavlja (isključivo ascii znakovi) – svaki od n_s signala posebno je specificiran u zaglavlju [7]

Ponoć se označava s 00:00:00. Preporuča se da trajanje svakog zapisa podataka bude cijeli broj sekundi, a njegova veličina u bajtovima ne bi trebala prijeći 61440. U slučaju da jedna sekunda zapisa podataka prekorači ovaj broj bajtova, trajanje svakog zapisa bi ipak trebalo biti kraće od jedne sekunde.

Digitalni minimum i maksimum svakog signala trebali bi specificirati ekstremne vrijednosti koje se mogu pojaviti u zapisima podataka. To su često ekstremne izlazne vrijednosti A/D pretvornika. Fizikalni minimum i maksimum ovih signala trebali bi odgovarati ovim digitalnim ekstremima i biti izraženi u određenim fizikalnim veličinama signala. Ekstremi specificiraju pomak i pojačanje signala.

Zaglavlje sadrži trajanje svakog zapisa podataka u sekundama, odnosno trajanje n_s signala u svakom zapisu u sekundama, pri čemu je svaki signal određen u zaglavlju specificiranim brojem uzoraka po jednom zapisu podataka.

U cilju smanjenja veličine podataka i prilagodbe često korištenom softveru za nabavljanje, obradu i grafički prikaz poligrafskih signala, vrijednost svakog uzorka reprezentirana je kao dvobajtni integer u formatu dvojnog komplementa (Slika 2.4).

nr of samples[1] * integer first signal in the data record
nr of samples[2] * integer second signal
...
nr of samples[ns] * integer last signal

Slika 2.4. Prikaz formata zapisa podataka (isključivo cjelobrojni podaci) [7]

2.3.2 Specifikacija EDF+ formata [8]

EDF+ je mnogo detaljnija specifikacija EDF-a, objavljena 2003. Formati su međusobno kompatibilni, tako da postojeći EDF preglednici ipak mogu prikazati EDF+ signale. EDF+, ipak, podržava i isprekidane zapise kao što su tekstualne bilješke, podražaji i događaji u UTF-8 formatu. Zahvaljujući tome, EDF+ može pohraniti bilo kakve medicinske zapise, kao što su evocirani potencijali ili faze sna. Specifikacija je dosta stroža od osnovnog EDF-a, a riješeni su i neki veći i manji propusti. U nastavku je prikazana osnovna usporedba s EDF formatom.

EDF+ format se također sastoji od zaglavlja iza kojeg slijede zapisi podataka. EDF+ zaglavlje se od osnovnog razlikuje samo u polju rezervirana 44 znaka (tzv. *reserved* polje), koje mora početi s oznakom '*EDF+C*' ukoliko je zapis podataka neprekinut, odnosno vrijeme početka svakog zapisa podataka se podudara s krajem prethodnog (kraj prethodnog jednak je zbroju početka i trajanja prethodnog). U tom slučaju datoteka je EDF kompatibilna i snimanje završava brojem zapisa podataka * trajanje zapisa podataka sekundi od početnog vremena. Ako zapis podataka nije neprekinut, polje *reserved* mora započeti s oznakom '*EDF+D*'. Polje *version* u EDF+ datoteci mora sadržavati '0', kao i u običnoj EDF datoteci, kako bi postojeći EDF preglednici mogli raditi i s EDF+ datotekama, bile one kontinuirane ili isprekidane, i prikazivati ih kao EDF datoteke.

EDF+ zapisi podataka također sadrže niz dvobajtnih uzoraka, koji su uzimani u jednakim vremenskim intervalima, osim toga zapisi podataka smiju bezuvjetno biti kraći od jedne sekunde, i naravno podaci smiju biti neprekinuti. Međutim, kao i u EDF-u slijed zapisa podataka u datoteci mora očuvati njihov vremenski slijed.

Zahvaljujući proširenju EDF+ formata, svi signali, bilješke i događaji koji su zabilježeni u jednom snimanju mogu se sigurno čuvati zajedno u jednoj datoteci, a moguća je i pohrana samo bilješki i događaja primjerice, bez signala. Zato je ovaj format puno fleksibilniji i omogućava korisniku odabir optimalne kombinacije.

3. Zahtjevi postavljeni na programski produkt

Programski produkt treba biti pisan u programskom jeziku Java, s grafičkim korisničkim sučeljem. Potrebno je omogućiti učitavanje EEG signala iz EDF datoteke nastale korištenjem sustava 10-20 za raspored elektroda i nomenklaturu, te prikazivanje više signala odjednom na različitim vremenskim skalama.

Konkretni funkcionalni zahtjevi koje je potrebno zadovoljiti, organizirani po funkcionalnim područjima su sljedeći:

1. Učitavanje

- Programski produkt treba imati mogućnost učitavanja datoteka u EDF i EDF+ formatu

2. Vizualizacija

- Treba biti moguće prikazati više signala odjednom po jednom prikazu
- Treba postojati mogućnost odabira i prikaza signala na različitim vremenskim skalama (opcije su 1 s, 5 s, 10 s, 30 s, 1 min, 30 min, 1 h, čitavo snimanje)
- Treba postojati mogućnost odabira i prikaza različite amplitude u mikrovoltima (opcije su 1, 10, 20, 50, 100, 200, 500, 1000)
- Programski produkt treba imati funkcionalnost za primjenu različitih faktora skaliranja po y osi (1, 2, 3, 4, 5, 10, 15, 20)
- Kod prikazivanja signala, trebaju biti prikazane i vertikalne crte koje označavaju određene vremenske trenutke
- Trebaju biti prikazane deblje crvene vertikalne crte koje označavaju vremenske trenutke određene odabirom vremenske skale
- Trebaju biti prikazane horizontalne crte koje predstavljaju minimalnu i maksimalnu odabranu amplitudu, te horizontalna crta koja predstavlja sredinu između njih, osim u slučaju odabira standardne opcije s amplitudom 50 μ V i razmakom između vertikalnih crta 0.2 s

- Treba biti prikazana oznaka pojedinog signala, koja je u svakom trenutku vidljiva, bez obzira na korištenje trake za pomicanje (engl. *scrollbar*)
- Treba biti prikazana odabrana amplituda zajedno s odgovarajućom fizikalnom veličinom, koja je u svakom trenutku vidljiva, bez obzira na korištenje trake za pomicanje
- Treba biti omogućen prikaz vremenske oznake na nekim vertikalnim crtama (zbog preglednosti ne na svima), koji je uvijek vidljiv, bez obzira na korištenje trake za pomicanje

3. Pomicanje

- Treba postojati mogućnost horizontalnog pomicanja trenutno prikazanog odsječka signala, s time da se horizontalna traka za pomicanje pojavljuje u slučaju da je duljina prikaza signala veća od širine prozora u kojem se prikazuju
- Treba postojati mogućnost vertikalnog pomicanja, s time da se vertikalna traka za pomicanje pojavljuje u slučaju da se svi signali odabrani za signal ne mogu prikazati u istom prozoru, osim ako korisnik nije odabrao klasičnu opciju s amplitudom 50 μV i razmakom između vertikalnih crta 0.2 s, pri čemu su svi signali prikazani u istom prozoru.

4. Dodavanje novih i brisanje signala za prikaz

- Treba biti omogućen jednostavan odabir signala za prikazivanje, kao i mogućnost dodavanja novih signala u prikaz, bez ponovnog označavanja već prikazanih
- Dijalog omogućava označavanje i brisanje određenih ili svih signala koji su prikazani
- Opcija „*clear all*“ omogućuje jednostavno brisanje svih prikazanih signala bez označavanja

5. Spremanje uzoraka odabranih signala u tekstualnu datoteku

- Programski produkt treba imati sljedeće mogućnosti ispisa signala:

- ispis svih signala iz izvorne EDF datoteke
- ispis određenih signala iz izvorne EDF datoteke
- ispis svih signala odabranih za prikaz
- ispis određenih signala odabranih za prikaz
- Treba postojati mogućnost odabira oblika za ispis (samo uzorci, uzorci s proteklim vremenom, uzorci s vremenom i datumom proteklim od početnog vremena i datuma snimanja)
- Treba postojati mogućnost odabira vremenskog odsječka za koji se ispisuju uzorci, pri čemu korisnik može definirati početno i završno vrijeme u formatu *hh:mm:ss.mmm*
- Omogućiti odabir naziva izlazne datoteke
- Omogućiti odabir mjesta na koje se sprema izlazna datoteka

6. Prikaz svojstava EDF datoteke i signala

- Treba postojati mogućnost prikaza relevantnih podataka koji se nalaze u zaglavlju
- Treba postojati mogućnost odabira prikaza svojstava nekog od signala iz izvorne EDF datoteke
- Treba postojati mogućnost odabira prikaza svojstava nekog od prikazanih signala

Nefunkcionalni zahtjevi su sljedeći:

Zahtjevi kvalitete:

- Programski produkt mora biti jednostavan za korištenje, s intuitivnim i razumljivim grafičkim sučeljem
- Čim se odabere određena opcija u izborniku, potrebno je osvježiti prikaz, bez potrebe za tipkom „refresh“
- Potrebno je što preciznije prikazati signale, što znači da su u algoritmu za iscrtavanje dozvoljene samo manja odstupanja

- Potrebno je omogućiti bar jedan scenarij u kojem je signal potpuno vidljiv na ekranu (bez da dio bude „odrezan“), ovisno o odabranim opcijama

Ograničenja:

- Problematican rad s velikim EDF datotekama (sporije učitavanje i iscrtavanje u slučaju velikog broja uzoraka)
- Moguće je prikazati odjednom samo signale koji se nalaze u istoj EDF datoteci, te se u slučaju potrebe za usporedbom prikaza signala iz različitih datoteke svaka datoteka mora zasebno ispočetka učitati
- Nemogućnost odabira svih faktora za skaliranje, ovisno o broju prikazanih signala
- Zbog preglednosti je moguće prikazati samo 10 signala na istom ekranu (osim u slučaju odabira klasičnog prikaza 50 μ V x 0.2 s). U slučaju da broj odabranih signala prijeđe taj broj, potrebno je koristiti vertikalnu traku za pomicanje kako bi se pregledali svi signali
- Zbog pikseliziranosti ekrana na kojem se prikazuju signali iscrtavanje nije potpuno precizno
- EDF datoteka mora biti potpuno u skladu sa specifikacijom, bez odstupanja
- Ulazna datoteka mora biti isključivo u EDF ili EDF+ formatu, s *.edf*, odnosno *.EDF* ekstenzijom
- EDF+ datoteka se prikazuje kao kontinuirana EDF datoteka, što znači da ne postoji mogućnost prikaza bilješki i ostalih isprekidanih zapisa

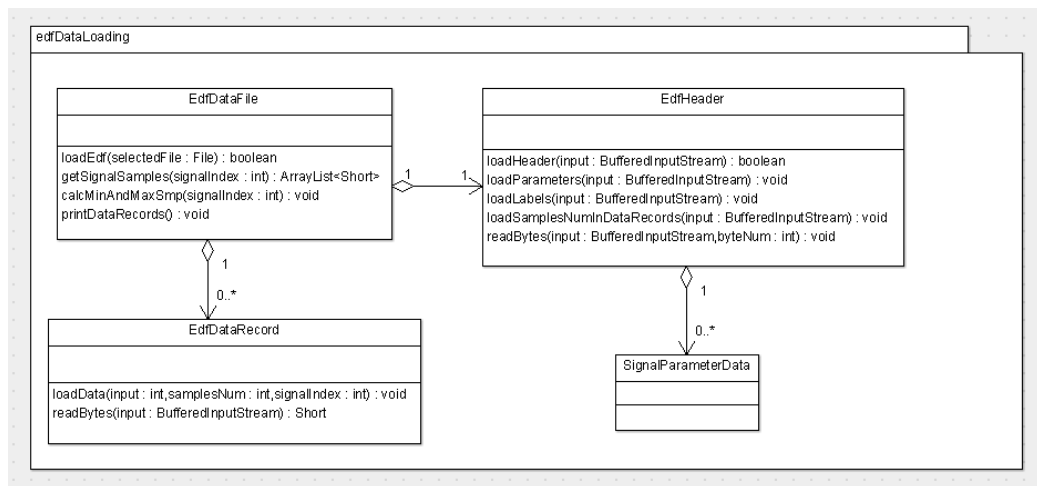
4. Arhitektura i izgradnja programskog produkta

Programski produkt je izrađen u programskom jeziku Java zbog prenosivosti, smanjenja složenosti koje omogućuje objektno usmjerena arhitektura i olakšane izrade grafičkog korisničkog sučelja. U izradi je korišteno integrirano razvojno okruženje Eclipse IDE za Javu.

Sustav je jednostavne arhitekture, jer ga čini samo jedna glavna izvršna komponenta, ostvarena kao programski produkt u Javi. Ostvaren je kroz dva programska modula. Prvi je ostvaren kao standardna Java biblioteka u kojoj je ugrađena funkcionalnost posvećena rukovanju podacima u EDF formatu, dok drugi modul predstavlja grafičko sučelje programskog produkta.

4.1 Rukovanje podacima iz EDF datoteke

Za učitavanje, osnovnu obradu i pohranjivanje podataka iz ulazne datoteke zaduženi su razredi iz paketa *edfDataLoading*. Njihov odnos prikazan je dijagramom razreda na slici 4.1:



Slika 4.1 Dijagram prikazuje odnos između razreda paketa *edfLoading*. Zbog preglednosti su navedene samo neke javne metode. Razred *SignalParameterData* služi isključivo za pohranu parametara signala, te stoga sadrži samo privatne varijable koje nisu prikazane dijagramom.

Konkretni razredi iz ovog paketa navedeni su i detaljno opisani u nastavku.

1. *EdfDataRecord*

Ovaj razred predstavlja jedan zapis podataka iz ulazne datoteke. Služi za pohranu uzoraka svih signala unutar zapisa. Sadrži metodu *readBytes*, koja čita dva bajta u formatu little-endian.

2. *SignalParameterData*

Razred pohranjuje sve važne parametre za jedan signal, kao što su oznaka signala, fizikalni minimum i maksimum, digitalni minimum i maksimum, te broj uzoraka u pojedinom zapisu podataka, ali osim podataka dobivenih učitavanjem EDF datoteke sadrži i podatak o frekvenciji uzorkovanja signala, te najvećem i najmanjem uzorku. Pritom su, za svaki pojedini parametar koji je potrebno učitati, kao konstante eksplicitno navedeni brojevi bajtova koje parametar zauzima u ulaznoj datoteci, prema specifikaciji EDF formata.

3. *EdfHeader*

Ovaj razred predstavlja zaglavlje u EDF datoteci. Sadrži članske varijable koje pohranjuju važne podatke o verziji formata (*versionOfDataFormat*), informaciju o pacijentu, snimanju, vremenu i datumu (*localPatientIdentification*, *localRecordingIdentification*, *recordingStartDate*, *recordingStartTime*), broju i trajanju zapisa podataka (*dataRecordsNum*, *dataRecordDuration*), broju signala u svakom zapisu (*signalsNum*), kao i listu objekata tipa *SignalParameterData*, koja sadrži važne podatke o svim signalima u datoteci. Slično kao u klasi *SignalParameterData*, ovaj razred pomoću konstanti specificira broj bajtova za svaki podatak koji se učitava. Sadrži metodu *loadHeader*, čijim pozivom se učitavaju svi podaci iz zaglavlja ulazne datoteke.

4. *EdfDataFile*

Predstavlja čitavu EDF datoteku, odnosno pohranjuje sve podatke iz zaglavlja i iz zapisa podataka. U tu svrhu sadrži člansku varijablu tipa *EdfHeader* i listu objekata tipa *EdfDataRecord*, u koju se spremaju svi zapisi podataka. Ovaj razred otvara odabranu datoteku za čitanje, obavlja popunjavanje zaglavlja i podataka pozivanjem metoda *loadHeader* i *loadData*, te računa uzorak najmanje i najveće amplitude, što će biti potrebno kasnije kod skaliranja i iscrtavanja signala.

4.2 Implementacija grafičkog korisničkog sučelja

Grafičko sučelje ostvareno je korištenjem ugrađene podrške za GUI u Javi, korištenjem paketa *javax.swing.** (u nastavku Swing) i *java.awt.** (u nastavku AWT) [9]. Dok je AWT (engl. *abstract window toolkit*) stariji paket u kojem izgled i ponašanje programa ovise o platformi na kojoj se izvodi, jer radi s grafičkim komponentama koje postoje na operacijskom sustavu, Swing ne koristi grafičke rutine platforme na kojoj se program izvršava i stoga rezultira konzistentnim izgledom programskog sučelja na različitim platformama. Osim toga ima znatno bogatiji izbor grafičkih komponenti, te ispravlja neke nedostatke AWT-a. Iz tog razloga korišten je uglavnom Swing, no kako on nije potpuna zamjena za AWT, korištena su oba paketa u kombinaciji.

Unutar paketa *gui* nalazi se čitav niz klasa koje omogućavaju učitavanje i vizualizaciju signala i korištenje dodatnih opcija na jednostavan i intuitivan način:

1. *EEGViewerMain*

Ovo je razred koji sadrži metodu *main*, te stvara instancu glavnog prozora i postavlja ga vidljivim.

2. *MainFrame*

Instanca ovog razreda je glavni okvir grafičkog sučelja. Nasljeđuje ugrađeni razred *JFrame* iz Swing paketa. Osnovne komponente koje sadrži su izbornik i ploču (engl. *panel*) koja služi za vizualizaciju signala. Odabirom opcije *File* → *Open...*

stvara se prozor koji služi za odabir i otvaranje EDF datoteke, te pohranjivanje podataka u instancu razreda *EdfDataFile* pozivom metode *loadEdf*. Ovaj razred je zadužen za stvaranje izbornika i osnovnog panela za prikazivanje signala (instanca razreda *SignalDrawing*), dijaloga za odabir signala (instanca razreda *SignalChooserWindow*), dijaloga koji služi za zapis signala u tekstualnu datoteku (instanca razreda *ExportSamplesWindow*), dijaloga za odabir prikaza svojstava zaglavlja i signala (instance razreda *HeaderPropertiesWindow* i *PropertiesSignalChooser*), te stvaranje instance razreda *DrawingParameters* i postavljanje njegovih članskih varijabli u ovisnosti o odabranoj opciji u izborniku.

3. *SignalDrawing*

Najvažniji razred koji služi za iscrtavanje i prikaz signala. Nasljeđuje ugrađeni razred *JPanel* iz *Swing* paketa i sadrži metodu *protected void paintComponent(Graphics g)*, u kojoj je ostvareno osnovno iscrtavanje signala. Sadrži instancu razreda *DrawingParameters* i članske varijable koje postavlja ovisno o vrijednostima zadanim u njoj, kao što su broj sekundi prikazanih na jednom ekranu (*double secondsPerDisplay*), amplituda (*int amplitude*), broj signala koji su odabrani za prikaz (*int numSignalsToDraw*) i drugi. Sadrži i metodu *reset* koja služi za brisanje svih iscrtanih signala. Valja spomenuti i metodu *setConstantPositionValues*, koja postavlja vrijednosti čija se pozicija ne mijenja u ovisnosti o pomicanju horizontalne ili vertikalne pomične trake. Način iscrtavanja je detaljnije opisan u cjelini „4.3 Vizualizacija signala“.

4. *DrawingParameters*

Razred koji služi za pohranu parametara u ovisnosti o kojima se iscrtavaju signali, kao što su indeksi signala koje treba iscrtati, amplituda, broj sekundi koje se prikazuju na ekranu, faktor za skaliranje po visini i drugi.

5. *SignalChooserWindow*

Ovaj razred nasljeđuje ugrađeni razred *JDialog* iz paketa *Swing* i predstavlja dijalog za odabir signala za prikaz ili za brisanje.

6. *ExportSamplesWindow*

Dijalog koji služi za odabir signala čiji uzorci se ispisuju u tekstualnu datoteku. Sadrži metode za formatiranje vremena i datuma koji će biti ispisani pored uzoraka.

7. *PropertiesSignalChooser*

Jednostavan dijalog koji sadrži signale čija svojstva se mogu odabrati za prikaz.

8. *SignalPropertiesWindow*

Dijalog koji prikazuje svojstva odabranog signala.

9. *HeaderPropertiesWindow*

Dijalog koji prikazuje svojstva EDF datoteke.

10. *HorizontalConstPosValues*

Razred služi za pohranu podataka o vrijednostima koje je potrebno prikazati, a koje imaju nepromjenjivu poziciju s obzirom na horizontalno pomicanje, kao što su oznaka i fizikalna dimenzija signala.

11. *VerticalConstPosValues*

Razred služi za pohranu podataka o vrijednostima koje je potrebno prikazati, a koje imaju nepromjenjivu poziciju s obzirom na vertikalno pomicanje, kao što su vremenske oznake na dnu.

12. *ConstPosValuesDrawing*

Ovaj razred obavlja konkretno iscrtavanje i prikaz svih vrijednosti čija se pozicija ne mijenja, bilo u odnosu na horizontalno, bilo u odnosu na vertikalno pomicanje. Postupak iscrtavanja i suradnje s drugim klasama detaljnije je opisan u cjelini „4.3.3 *Iscrtavanje vrijednosti s pozicijom neovisnom o pomicanju*“.

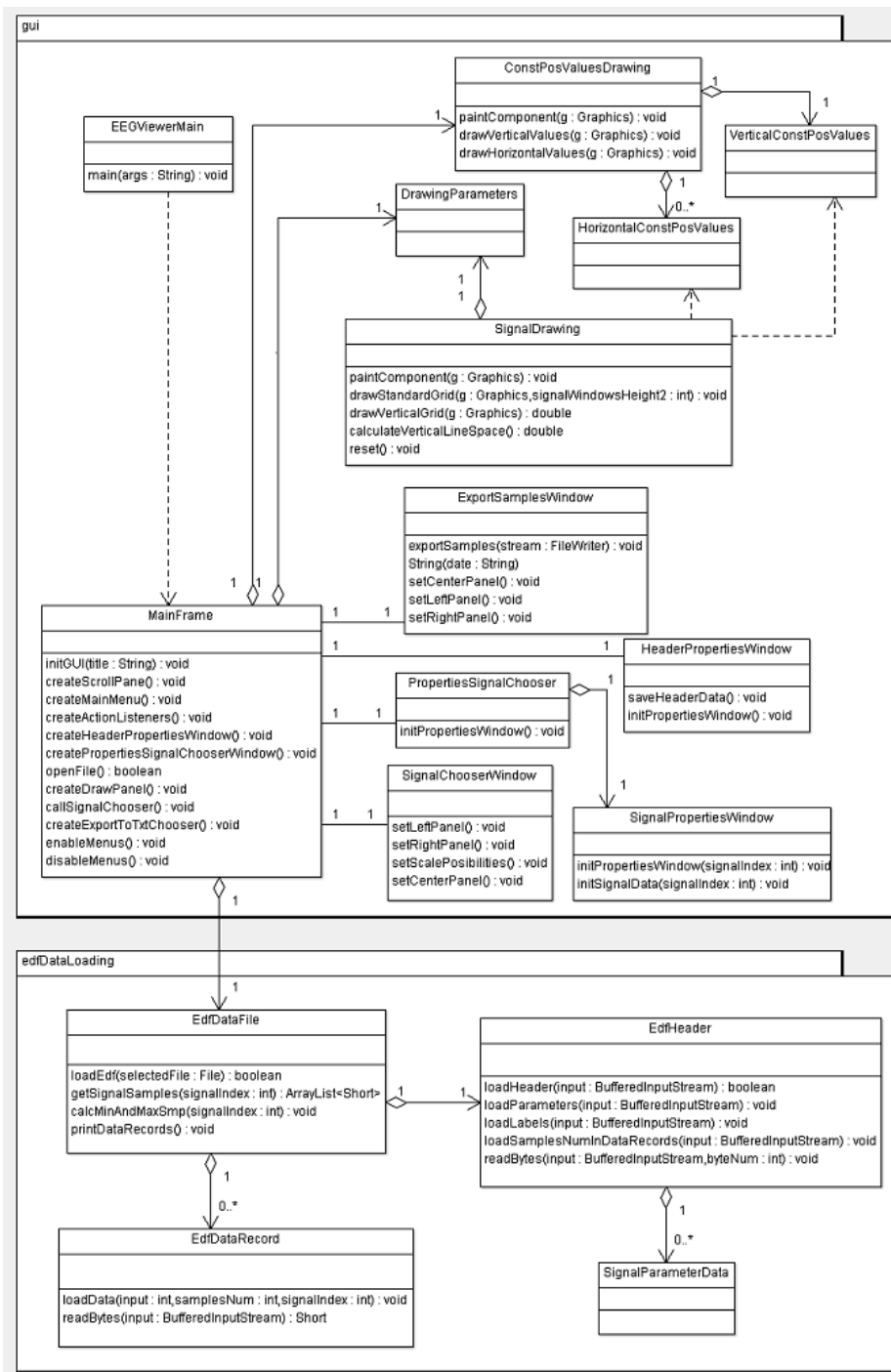
13. *OpenFileFilter*

Apstraktni razred koji nasljeđuje ugrađeni *javax.swing.filechooser.FileFilter*, a koristi se u klasi *MainFrame* za filtriranje ekstenzije prilikom učitavanja datoteke, kako bi se prikazivale i mogle odabrati samo „.edf“, odnosno „.EDF“ datoteke.

14. *SaveFileFilter*

Apstraktni razred koji nasljeđuje ugrađeni *javax.swing.filechooser.FileFilter*, a koristi se u klasi *ExportSamplesWindow* za filtriranje ekstenzije datoteke u koju se spremaju uzorci („.txt“).

Odnos razreda paketa *edfDataLoading* i *gui* prikazan je dijagramom razreda (Slika 4.2).



Slika 4.2. Dijagram razreda prikazuje cjelokupan sustav. Zbog preglednosti su navedene samo neke javne metode. Razredi *ConstPosValuesDrawing* i *SignalDrawing* nasljeđuju ugrađeni razred *JPanel*, razredi *ExportSamplesWindow*, *PropertiesSignalChooser*, *SignalChooserWindow* i *SignalPropertiesWindow* nasljeđuju ugrađeni razred *JDialog*, a razredi *MainFrame* i *HeaderPropertiesWindow* nasljeđuju ugrađeni razred *JFrame*, no to zbog preglednosti nije prikazano na dijagramu. Nisu prikazani ni razredi *OpenFileFilter* i *SaveFileFilter*

4.3 Vizualizacija signala

Prikaz signala čini temeljnu funkcionalnost programskog produkta. Razred koji se bavi skaliranjem i iscrtavanjem signala u ovisnosti o različitim parametrima jest *SignalDrawing*. Metoda *paintComponent*, koja obavlja konkretna iscrtavanja, poziva se prilikom svake promjene veličine prozora, korištenja pomične trake i odabira određene opcije u izborniku.

4.3.1 Određivanje veličine prostora za iscrtavanje

Prilikom iscrtavanja potrebno je panelu na koji se crta postaviti željenu veličinu. Naime, kao površina za prikaz signala koristi se prilagođena inačica ugrađenog razreda Swing paketa *JPanel*, koji svoju veličinu određuje prema komponentama koje sadrži. Kako on ima samo ulogu platna na koje se crta, bez metode koja će na neki način postaviti ili vratiti veličinu će biti moguće iscrtati samo dio signala koji se nalazi u dijelu koji je vidljiv. Kako bi bilo moguće iscrtati cijeli signal i omogućiti korištenje horizontalne i vertikalne pomične trake, koristi se metoda *getPreferredSize* koja vraća tip *Dimension*, odnosno širinu i visinu panela. Ukoliko je broj signala odabranih za prikaz veći od 0, potrebno je odrediti širinu i visinu u pikselima.

Kod određivanja širine potrebno je izračunati faktor kojim se množi ukupan broj uzoraka nekog od signala (*signalXScale*), a koji ovisi o širini vidljivog dijela panela, broju sekundi odabranih za prikaz, broju uzoraka signala u jednom zapisu podataka i trajanju jednog zapisa podataka, te predstavlja horizontalni razmak između dva uzorka na ekranu (kako *signalXScale* nije cijeli broj, a razmak je u pikselima, prilikom iscrtavanja postoje mala odstupanja od stvarnih vrijednosti).

Visinu je lakše odrediti, jer ovisi samo o broju signala odabranih za prikaz. Ukoliko je taj broj manji od zadanog broja (zbog preglednosti je određeno da po jednom prikazu ne smije biti više od deset signala) ili je odabrana standardna opcija $50\mu\text{V} \times 0.2\text{s}$, svi će signali biti vidljivi (unutar istog „prozora“), bez potrebe za vertikalnom pomičnom trakom. Ako je broj izabranih signala veći od deset, bit će

potrebno odrediti novu visinu prozora, množenjem visine jednog „prozora“ s omjerom broja izabranih signala i maksimalnog broja signala po prikazu (odnosno deset).

4.3.2 Iscrtavanje signala u ovisnosti o parametrima

Sâmo iscrtavanje signala je riješeno iznimno jednostavno – koristi se ugrađena metoda *drawLine*, koja povezuje po dva piksela s određenim x i y koordinatama. Međutim, određivanje tih koordinata s obzirom na različite opcije koje mogu biti odabrane (amplituda, vremenska skala, skaliranje po visini) nešto je složenije.

Prije samih signala crtaju se vertikalne i horizontalne linije, kojima se označavaju bitni parametri prikaza, kao što su određeni vremenski trenuci ili oznake amplituda. Pozicija vertikalnih linija određuje se na temelju odabrane vremenske skale i predefiniranog broja linija (koje označavaju vremenske trenutke) za odabrani broj sekunda po prikazu. Odabrani broj sekunda se, jednako kao amplituda, broj signala i faktor vertikalnog skaliranja, dohvaća iz instance razreda *DrawingParameters*, u kojoj je zapisan. Pozicija horizontalnih linija određuje se zasebno za svaki signal u *for* petlji. Prije ulaska u petlju dijeljenjem visine glavnog „prozora“ s brojem signala koji se prikazuju, računa se visina „prozora“ koju na raspolaganju ima svaki signal. Zatim se u petlji dohvaćaju indeksi signala (također zapisani u instanci klase *DrawingParameters*), za svaki se računa pozicija sredine i horizontalnih linija za amplitudu. Prilikom iscrtavanja se za horizontalnu poziciju ne uzima u obzir izračunat faktor horizontalnog skaliranja, a za vertikalnu *signalYScale*, faktor koji se računa za svaki signal i množi s amplitudom svakog uzorka, i u izborniku odabran faktor skaliranja po y osi (kojim se povećava ili smanjuje razmak između gornje i donje oznake amplitude).

Ukoliko je odabrana opcija $50 \mu\text{V} \times 0.2 \text{ s}$, računa se razmak između vertikalnih linija koji predstavlja 0.2 sekunde, te se iscrtavaju horizontalne linije, a

razmak između njih predstavlja 50 μ V. Signali se iscrtavaju unutar jedne visine prozora, bez mogućnosti vertikalnog pomicanja.

Ako nisu odabrani signali za prikaz, iscrtava se samo osnovna mreža horizontalnih i vertikalnih linija.

4.3.3 Iscrtavanje vrijednosti s pozicijom neovisnom o pomicanju

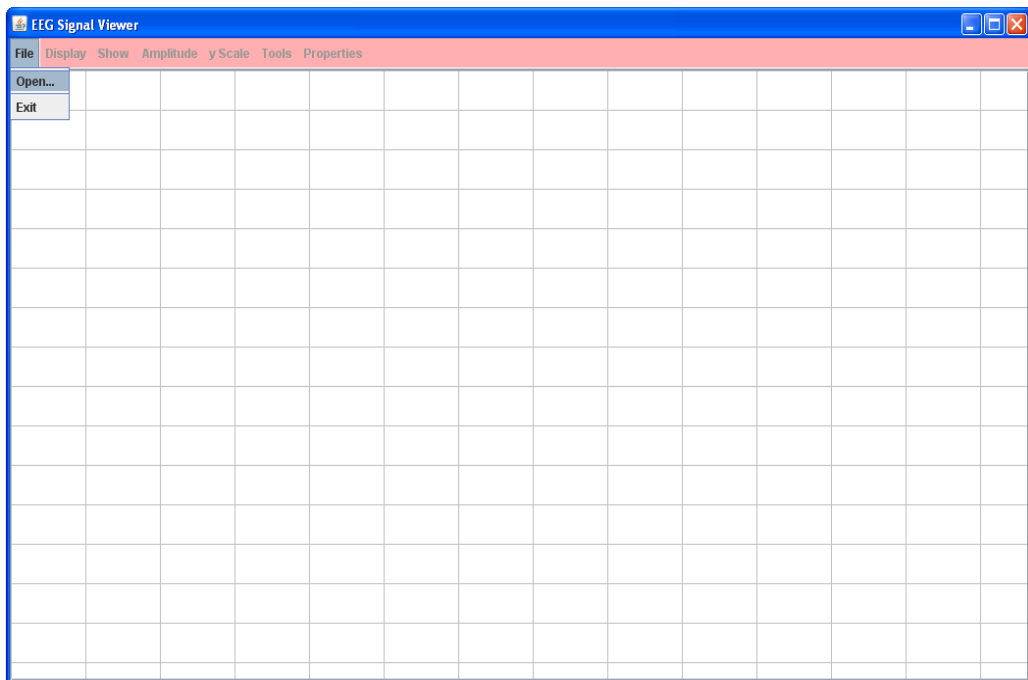
Razred *ConstPosValuesDrawing* preko iscrtanih signala ispisuje vrijednosti koje, uvjetno rečeno, uvijek moraju biti vidljive bez obzira na korištenje pomičnih traka. Međutim, to će za horizontalne vrijednosti značiti da će uvijek biti vidljive na lijevom rubu glavnog okvira ukoliko se pomiče samo horizontalna pomična traka (kako bi uvijek bila vidljiva amplituda, naziv i dimenzija signala), odnosno imati konstantnu poziciju na x osi, ali za vertikalno pomicanje horizontalne vrijednosti će se pomicati po y osi će se morati pribrajati trenutna pozicija vertikalne pomične trake. Slično tome se ponašaju i vertikalne vrijednosti koje imaju istu y poziciju kod vertikalnog pomicanja (vremenske oznake na donjem rubu glavnog okvira), a kod njih će se x poziciji pribrajati trenutna pozicija horizontalne pomične trake

Sâm razred ostvaren je kao tzv. *glass pane*, odnosno transparentan sloj koji prekriva čitav glavni prozor. Sadrži metodu *paintComponent* koja je zadužena za iscrtavanje vrijednosti, ukoliko je *glass pane* postavljen vidljivim [9]. Pritom se najprije iscrtavaju sve komponente ispod transparentnog sloja (dakle izvršava se *paintComponent* metoda instance klase *SignalDrawing* i iscrtavaju se svi signali), a tek potom se izvrši *paintComponent* klase *ConstPosValuesDrawing*.

5. Opis i primjeri korištenja programskog produkta

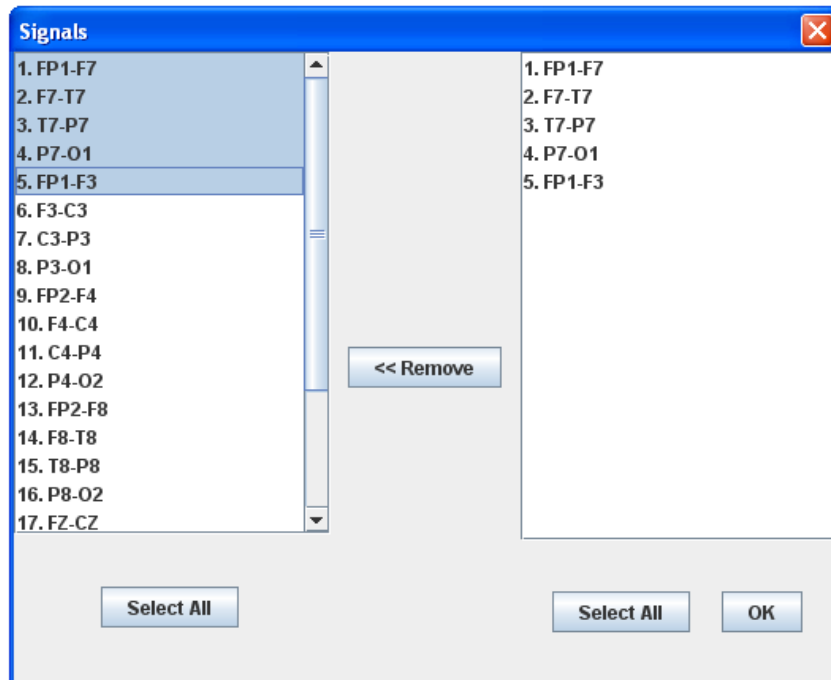
Svi primjeri korištenja pokazani su na datotekama iz referentne internetske banke fizioloških podataka PhysioBank, konkretno iz baze „CHB-MIT Scalp EEG Database“. Sve datoteke su u osnovnom EDF formatu, korištenjem sustava 10-20. U nastavku će se rad demonstrirati na datoteci `chb01_27.edf`, koja se sastoji od 23 signala mjerenih bipolarno.

Programski produkt je intuitivan za korištenje. Prilikom pokretanja otvara se glavni prozor koji se sastoji od izbornika i pozadine za prikaz signala, na kojoj na početku nema ništa osim osnovne pozadine (Slika 5.1). U početku je moguće odabrati samo opciju *File* u izborniku. Klikom na *Open...* otvara se dijalog za odabir ulazne datoteke, koja mora biti u osnovnom formatu opisanom u poglavlju „2.1.1 Specifikacija EDF formata“, ili u proširenom, EDF+ formatu. U primjeru je odabrana datoteka `chb01_27.edf`.



Slika 5.1. Pokretanje – početni prozor

Nakon odabira ulazne datoteke i učitavanja podataka, pojavljuje se prozor za odabir signala za prikaz (Slika 5.2). Isti prozor se pojavljuje i odabirom opcije *Display* → *Add/Remove Signals*.

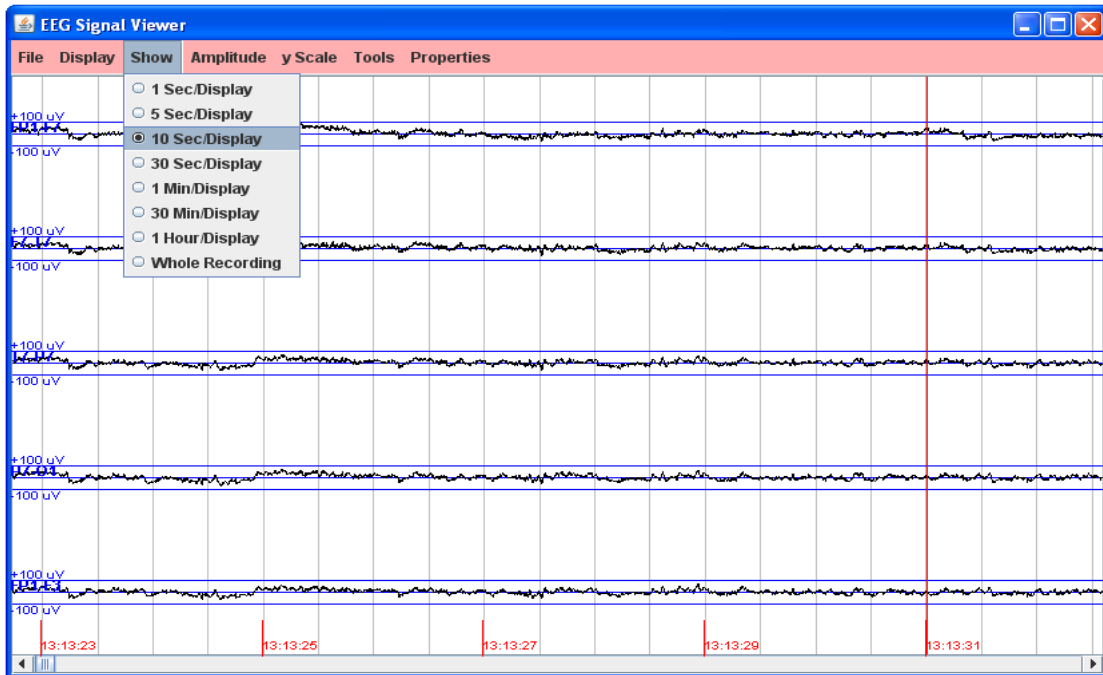


Slika 5.2. Dijalog za odabir signala za prikaz ili brisanje. Na slici je odabrano pet signala

Pritiskom na tipku *OK* odabrani signali se iscrtavaju.

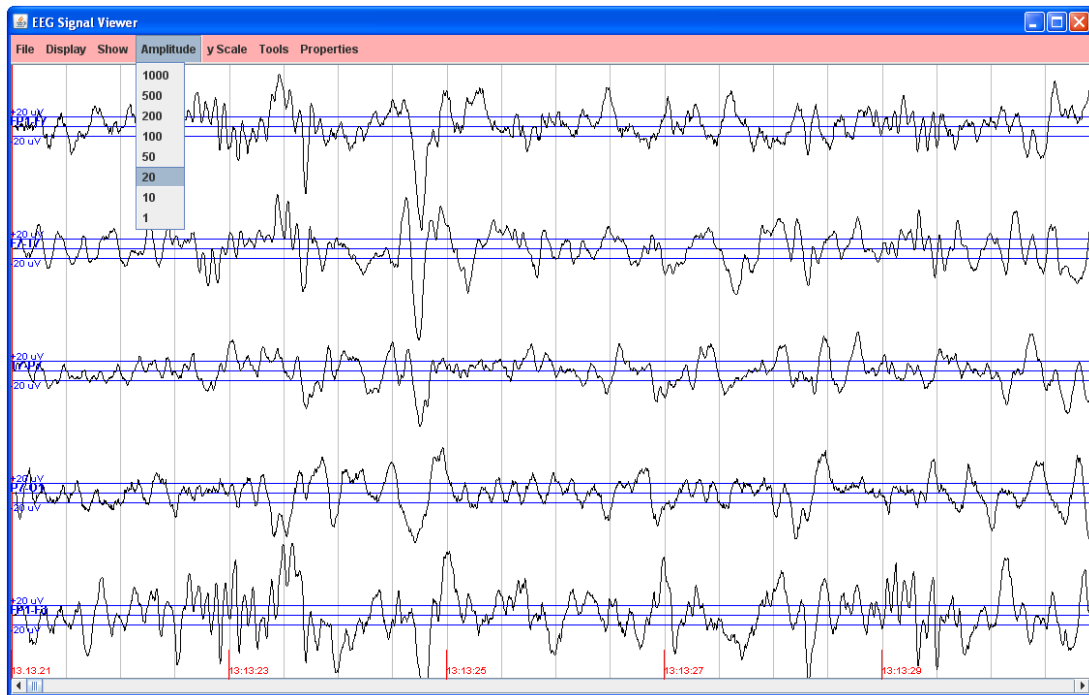
5.1 Opcije za prikaz signala

Moguće je odabrati različite opcije za prikaz signala. Signali mogu biti prikazani na različitim vremenskim skalama, odabirom opcije *Show* u izborniku. Primjerice, odabirom opcije *10 Sec/Display* (Slika 5.3), po jednom prikazu je vidljivo deset sekundi signala, a svakih deset sekundi iscrтана je vertikalna crvena crta zbog bolje preglednosti i lakšeg snalaženja. Deset sekundi po prikazu je ujedno i zadana opcija, koja je automatski odabrana kod prvog odabira signala za prikaz.



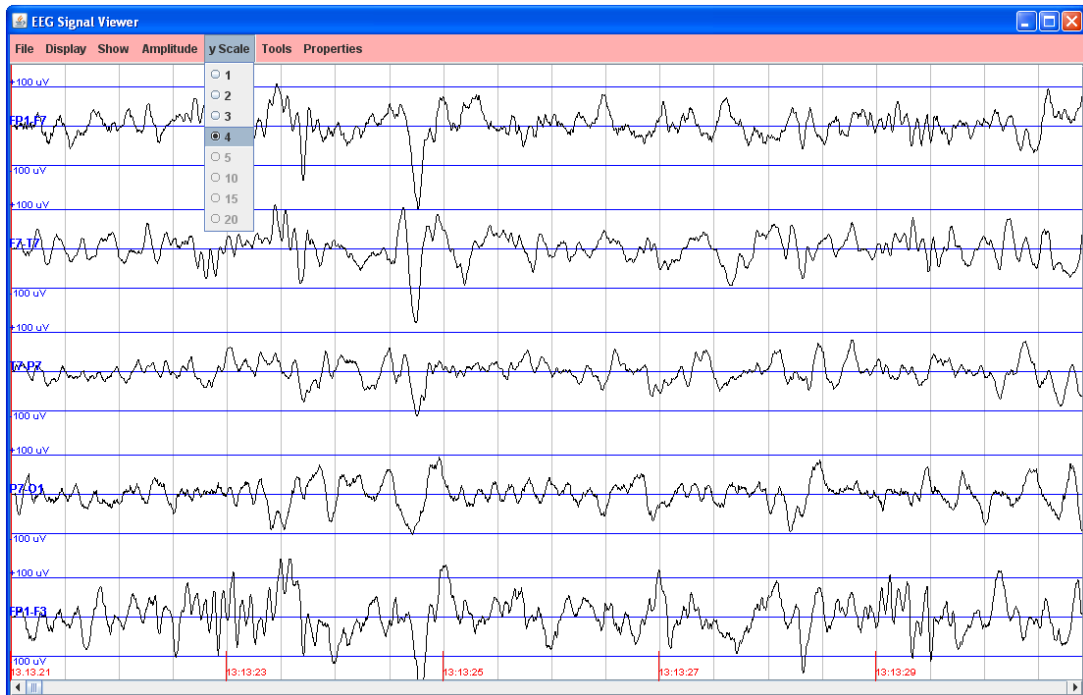
Slika 5.3. Prikaz pet odabranih signala u vremenskoj skali od 10 sekundi po prikazu s amplitudom 100 μV , vidljiva je vertikalna crvena crta nakon prvih 10 sekundi, na dnu su crveno prikazane vremenske oznake

Odabirom opcije *Amplitude* u meniju moguće je mijenjati amplitudu signala. Amplituda je EDF datoteci najčešće zadana u mikrovoltima (μV), a kako je amplituda spontanog EEG-a reda veličine 100 μV , zadana amplituda kod prvog odabira signala iznosi 100. Ukoliko je prikazano više signala odjednom, svi imaju jednaku amplitudu (Slika 5.4).



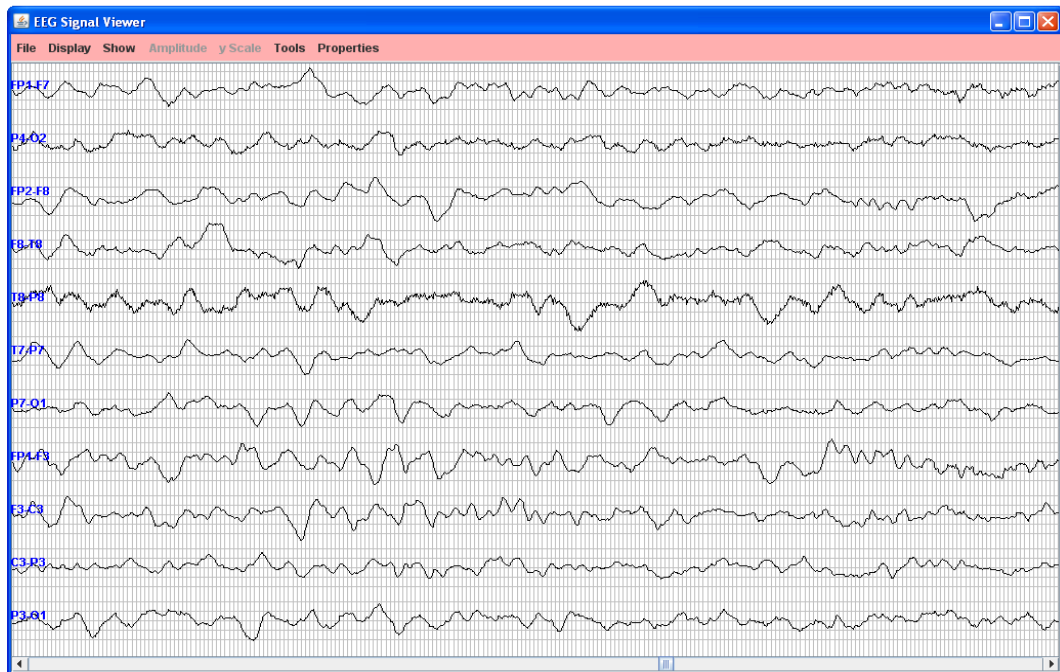
Slika 5.4. Prikaz pet istih odabranih signala u vremenskoj skali od 10 sekundi po prikazu, ali s amplitudom 20 μV

Kod prikaza signala postoji još i mogućnost vertikalnog skaliranja, odabirom opcije `yScale` u izborniku (Slika 5.5). Dostupne opcije skaliranja postoje ovisno o broju signala koji se prikazuju (zbog mogućih nepreglednih preklapanja veliki faktori skaliranja nisu dostupni kod prikaza više signala). Zadana vrijednost faktora vertikalnog skaliranja je 1, i odabrana je kod svakog dodavanja ili micanja signala.



Slika 5.5. Prikaz pet istih odabranih signala u vremenskoj skali od 10 sekundi po prikazu, s amplitudom 100 μV , ali odabranim faktorom skaliranja 4

Označavanjem opcije *Tools* \rightarrow *Standard Display(0.2sx50uV)* moguće je prikazati signale pomoću standardne opcije s intervalima između vertikalnih linija od 0.2 sekunde, te intervalima između horizontalnih linija od 50 μV (Slika 5.6). Pritom su svi signali prikazani u istom prozoru, odnosno da ne postoji mogućnost vertikalnog pomicanja. Nije moguće mijenjati amplitudu ni faktor skaliranja.



Slika 5.6. Opcija 0.2 s x 50 μ V uz odabranih 5 s po prikazu– svi signali su prikazani u istom prozoru, razmak između dvije vertikalne linije predstavlja 0.2 s, a razmak između dvije horizontalne linije predstavlja 50 μ V

Kako se ne bi ručno morali označavati svi signali, u slučaju potrebe za brisanjem svega iscrtanog i vraćanja svih opcija u početno stanje (zadane vrijednosti), postoji opcija *Display* \rightarrow *Clear All*, koja „resetira“ prikaz.

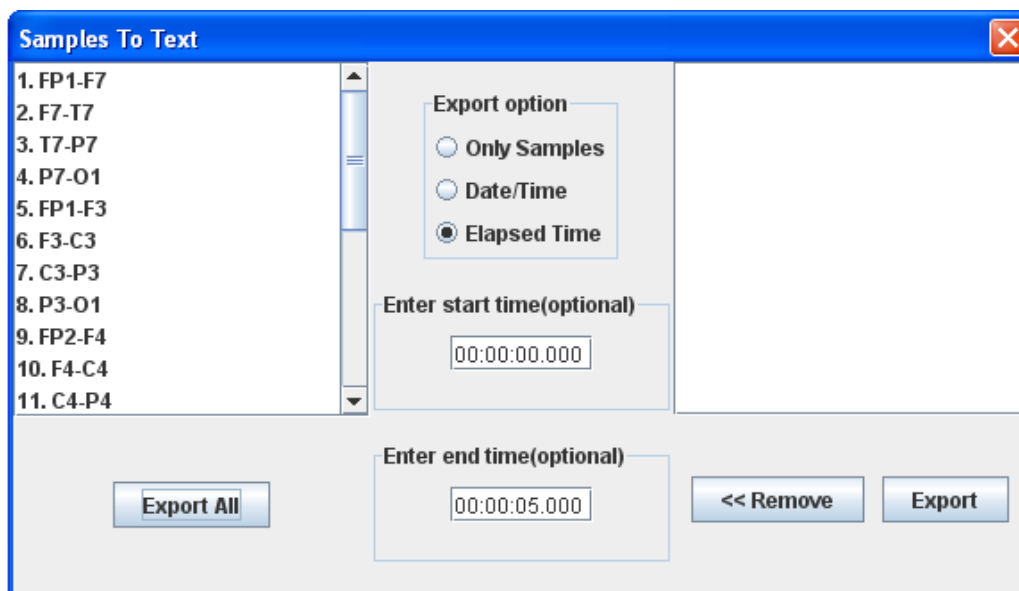
5.2 Dodatne opcije

Programski produkt osim prikaza signala pruža i mogućnosti ispisa uzoraka odabranih signala u tekstualnu datoteku, te prikaza bitnih informacija iz zaglavlja datoteke.

5.2.1 Ispis uzoraka u tekstualnu datoteku

Odabirom opcije *Tools* \rightarrow *Export Samples As Text* \rightarrow *All Signals* ili *Tools* \rightarrow *Export Samples As Text* \rightarrow *Displayed Signals*, otvara se dijalog u kojem je moguće izabrati vrstu ispisa (samo uzorke, uzorke i proteklo vrijeme, uzorke i stvarno vrijeme i datum protekli od početka snimanja), kao i ispis od željenog početnog do željenog završnog trenutka (Slika 5.7). Ukoliko korisnik ne upiše početno i završno vrijeme, ispisat će se svi uzorci, bez obzira na odabranu vrstu ispisa. Opcija *Only Samples* ispisuje uzorke i najmanji vremenski razmak između dva uzorka (u slučaju različitih

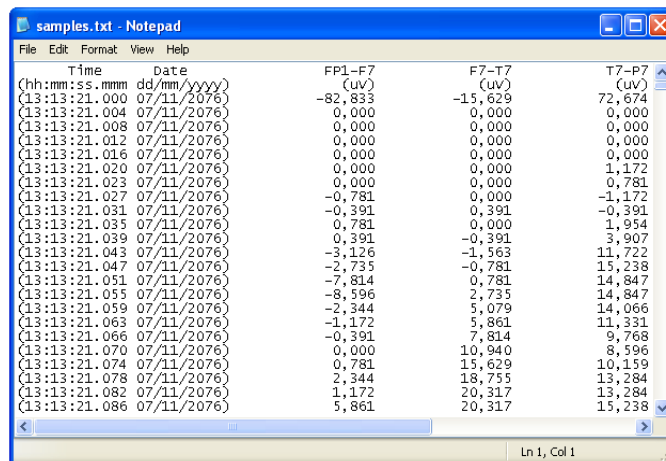
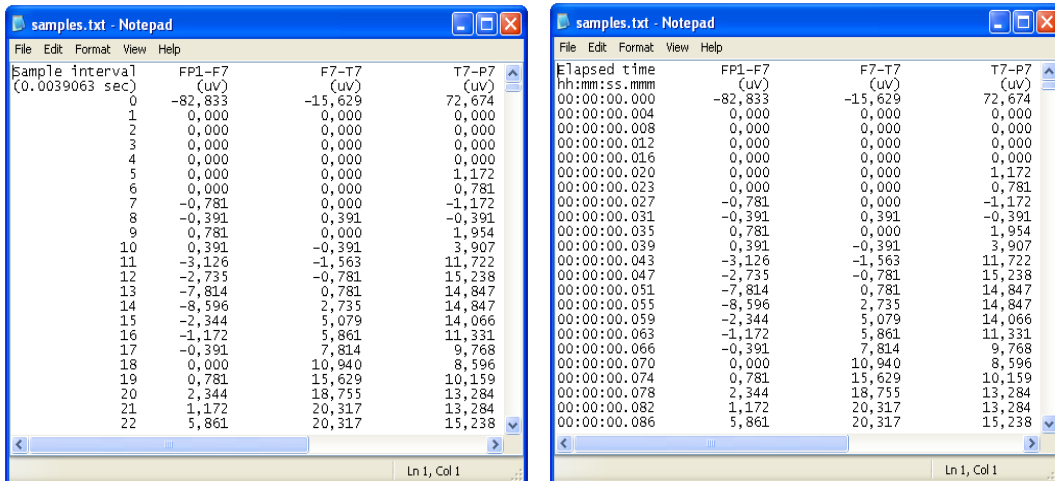
frekvencija uzorkovanja). Opcija *Date/Time* ispisuje vrijeme u formatu *hh:mm:ss.mmm*, te datum u formatu *dd/mm/yyyy*, a prilikom unosa početnog ili završnog vremena potrebno je paziti na to da se vrijeme računa od stvarnog vremena početka snimanja.



Slika 5.7. Dijalog za odabir signala i opcija za ispis u tekstualnu datoteku. Odabrana je opcija isписа proteklog vremena i uneseno je vrijeme do kojeg će se ispisivati uzorci, na slici je to pet sekundi

Opcija *Elapsed Time* omogućuje ispis proteklog vremena, a kao i *Only Samples* opcija, ne podržava korisnički unos koji ovisi o stvarnom vremenu početka snimanja. U slučaju da korisnik unese početno vrijeme veće od trajanja čitavog zapisa, u datoteku će se ispisati samo nazivi signala. Ukoliko je unesen početni trenutak manji od početnog trenutka prvog uzorka, ispis će početi od prvog uzorka. Ukoliko je unesen završni trenutak veći od završnog trenutka zadnjeg uzorka, ili je završni trenutak manji od početnog, ispisat će se svi uzorci, te će ispis stati na posljednjem.

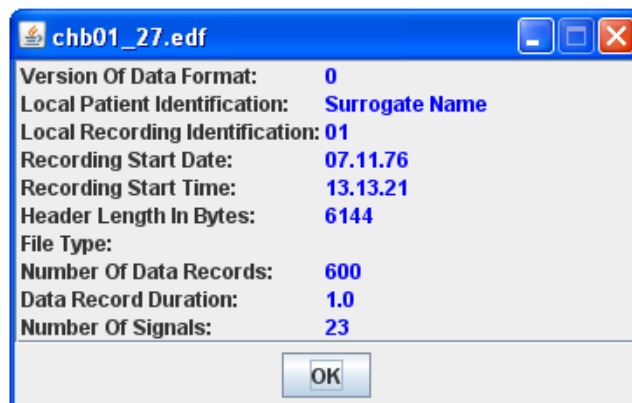
Na slici 5.8 prikazan je primjer nekoliko datoteka u kojima su ispisani uzorci signala s obzirom na različite opcije.



Slika 5.8. Na slici gore lijevo prikazana je datoteka nakon odabira opcije *Only Samples*, dok slika gore desno prikazuje izgled datoteke nakon odabira opcije *Elapsed Time*. Slika ispod prikazuje datoteku nakon odabira opcije *Date/Time*. U sva tri slučaja ispisani su svi uzorci, no zbog preglednosti nisu prikazani svi odabrani signali ni uzorci

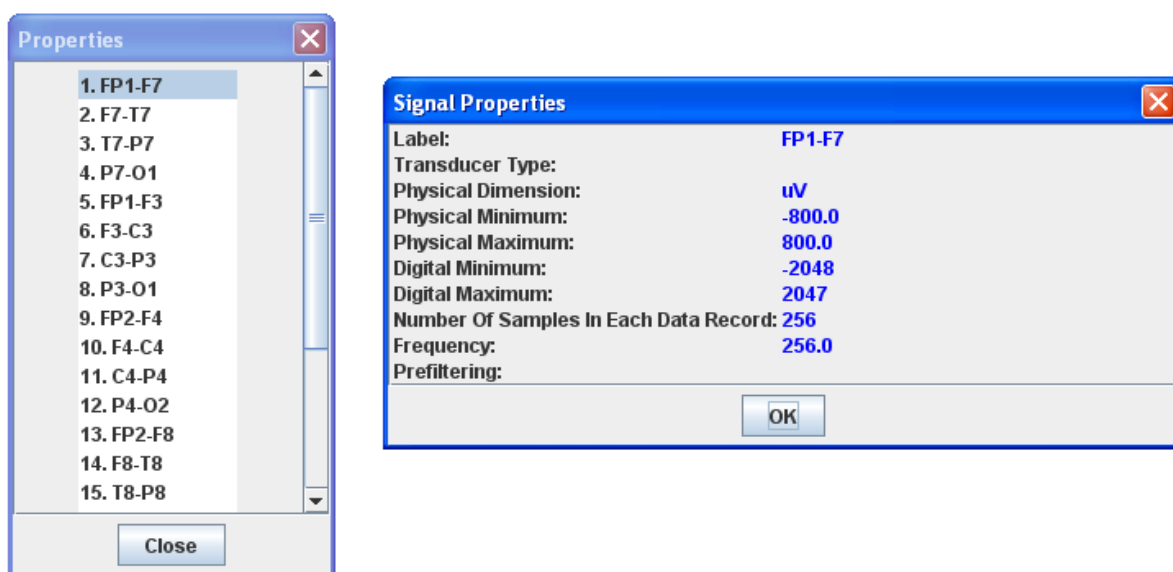
5.2.2 Prikaz bitnih informacija iz zaglavlja

Moguće je izabrati prikaz bitnih svojstava EDF datoteke i signala zapisanih u njezinom zaglavlju. Odabirom opcije *Properties* → *EDF file* otvara se prozor koji prikazuje informacije o datoteci i snimanju zapisane u zaglavlju, kao što su verzija, broj zapisa podataka, trajanje zapisa podataka, broj signala u svakom zapisu i drugi (Slika 5.9).



Slika 5.9. Prikaz informacija o datoteci i snimanju zapisanih u zaglavlju

Odabirom opcije *Properties* → *All Signals* ili *Properties* → *Displayed Signals* otvara se prozor sa signalima, a odabirom željenog signala u zasebnom prozoru prikazuju se podaci iz zaglavlja o njemu, poput fizikalne veličine, broja uzoraka u zapisu podataka i drugo (Slika 5.10).



Slika 5.10. Na slici gore lijevo prikazan je prozor sa signalima i odabran je signal „FP1-F7“, slika desno prikazuje prozor s podacima o signalu „FP1-F7“

6. Zaključak

Elektroencefalografija ima brojne prednosti, ali i ograničenja. Osnovni problem je ograničena prostorna razlučivost, a zbog manje osjetljivosti elektroda nije baš precizna, jer se ne mogu uhvatiti potencijali između pojedinih neurona, već samo između velikih grupa. Ova ograničenja bolje savladavaju druge, danas često korištene metode, kao što je primjerice fMRI (engl. *functional magnetic resonance imaging*), odnosno snimanje moždane aktivnosti korištenjem magnetske rezonancije.

Ipak, prednosti su mnogobrojne i značajne – elektroencefalografija je neinvazivna i bezopasna metoda, brza, jeftina i ima veliku vremensku razlučivost (od čak 1 ms) u usporedbi s fMRI, te se široko primjenjuje.

Izgrađeni programski produkt može se koristiti u edukacijske i istraživačke svrhe, jer omogućava pregledan prikaz i usporedbu EEG signala. Omogućen je odabir i pregled pojedinačnih signala, njihovo pohranjivanje u izlaznu tekstualnu datoteku, te prikaz svojstava signala i same datoteke, zapisanih u zaglavlju.

Korisno bi bilo implementirati mogućnost prikaza EDF+ datoteka koje nisu kontinuirane (trenutno je moguć samo prikaz nekontinuiranih u obliku kontinuiranih), kao i omogućiti učitavanje i prikaz drugih tipova datoteka koje sadrže EEG zapise, poput datoteka u BDF ili BDF+ formatu. Jedno od korisnih proširenja koje bi se moglo implementirati jest označavanje interesantnog područja pomoću miša, te potom njegova zasebna obrada i mogućnost zumiranja i prikaza konkretnih uzoraka. Osim toga, moguće je proširiti funkcionalnost produkta vizualizacijom potencijala na modelu glave čovjeka, pri čemu bi se mogla različitim bojama označiti aktivna područja po intenzitetu kroz vrijeme. Sustav bi se također mogao unaprijediti uvođenjem izlučivanja statističkih i nelinearnih značajki s ciljem modeliranja i predviđanja poremećaja.

Lea Suć

7. Literatura

- [1] Sörnmo, L., Laguna, P. *Bioelectrical signal processing in cardiac and neurological applications*, USA, Elsevier Inc., 2005
- [2] 13. *Electroencephalography*, <http://www.bem.fi/book/13/13.htm#03> , 10.05.2011.
- [3] *Action potential*, http://en.wikipedia.org/wiki/Action_potential, 20.05.2011.
- [4] Berger, H. *Über das Elektroencephalogramm des Menschen*, Archiv für Psychiatrie und Nervenkrankheiten, Vol. 87 (1929), pp. 527-570
- [5] *Electroencephalography*, <http://en.wikipedia.org/wiki/Electroencephalography>, 02.03.2011.
- [6] Herwig, U., Satrapi, P., Schönfeldt-Lecuona, C. *Using the International 10-20 EEG System for Positioning of Transcranial Magnetic Stimulation*, Brain Topography, Volume 16, Number 2, Winter 2003
- [7] Kemp, B., Värri, A., C. Rosa, A., D. Nielsen, K., Gade, J. "A simple format for exchange of digitized polygraphic recordings" *Electroencephalography and Clinical Neurophysiology*, 82 (1992): 391-393
- [8] Kemp, B., Oliven, J. *European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data*, *Clinical Neurophysiology*, 114 (2003): 1755-1761
- [9] *The Java Tutorials, How to use root panes*, <http://download.oracle.com/javase/tutorial/uiswing/components/rootpane.html>, 15.05.2011.
- [10] Eckel, B. *Thinking in Java*, Massachusetts, Prentice Hall, 2006

Oblikovanje i implementacija programskog produkta za prikaz elektroencefalografskih signala

Sažetak

Ljudski mozak se sastoji od neurona kojima se prenose informacije, čime nastaje potencijal kojeg je moguće izmjeriti. Elektroencefalografija je neinvazivna metoda kojom se pomoću elektroda na glavi, čiji se raspored definira nekim od postojećih sustava, može mjeriti moždana električna aktivnost. Postoji više tipova formata koji pohranjuju rezultat snimanja, elektroencefalogram. Uobičajen je normirani format EDF.

U sklopu završnog rada izgrađen je programski produkt koji učitava datoteku u EDF formatu, te omogućava prikaz jednog ili više signala odjednom na različitim vremenskim skalama, s različitim amplitudama i faktorima skaliranja. Omogućen je ispis signala u tekstualnu datoteku, kao i pregled svojstava zapisa i signala, koja se nalaze u zaglavlju.

Ključne riječi

elektroencefalografija, prikaz signala, programski produkt u Javi, sustav 10-20, format EDF

Design and implementation of application for displaying electroencephalographic signals

Abstract

Human brain consists of neurons used for information transfer. Neurons create measurable electric potential. Electroencephalography is a noninvasive method that uses electrodes placed on the subject's head, with their layout defined by some of the existing placement systems, in order to measure the brain's electric activity. There are many types of data format used to digitally record electroencephalogram, which is the result of electrode-based measuring. Most commonly used standard format is EDF.

This thesis describes implementation of software for visual presentation of signals stored in files in EDF format. The software enables visualization of many signals simultaneously, at different time-scales, amplitudes and scaling factors. Software is also capable of storing signal data in textual file, as well as to give an overview of data record and signal properties that are stored in file headers.

Keywords

electroencephalography, , visual representation of signals, application in Java, system 10-20, format EDF