

1. УВОД	2
2. ОБЈЕКТНО ОРЈЕНТИСАНО ПРОГРАМИРАЊЕ	3
2.1. ТЕМЕЉИ ОБЈЕКТНО ОРЈЕНТИСАНОГ ПРОГРАМИРАЊА	4
3. VISUAL FOXPRO	5
3.1. РАДНО ОКРУЖЕЊЕ VISUAL FOXPRO-A	5
3.1.1. Формирање табеле	6
3.1.2. Креирање форми	8
4. ФОРМИРАЊЕ КЛАСА У VISUAL FOXPRO-У	10
4.1. ВРСТЕ VISUAL FOXPRO КЛАСА	10
4.1.1. Основне класе у Visual FoxPro – у	11
4.1.2. Хијерархијско референцирање објекта	11
4.1.3. Релативно референцирање	12
5. РЕАЛИЗАЦИЈА СОФТВЕРА	13
5.1. ДИЗАЈНИРАЊЕ АПЛИКАЦИЈЕ	13
5.2. ПРОЈЕКТОВАЊЕ АПЛИКАЦИЈЕ	16
5.2.1. Главни програм	17
5.2.2. Програмски кодови	19
5.2.3. Демонстрација рада програма	25
6. ЗАКЉУЧАК	30
7. ЛИТЕРАТУРА	31

Софтверско инжењерство постало је једна од најважнијих области, јер утиче на готово све аспекте наших живота. Захваљујући бројним софтверима, данас нам је омогућено да брже и ефикасније радимо, комуницирамо, купујемо, информишемо се, и још много тога. Без софтверских решења, данас је немогуће замислити развој већине индустријских грана, медицине, образовања и готово свих важнијих области за напредак целокупног друштва.

Софтверско инжењерство фокусирано је на употребу рачунара као основног средства које се користи за решавање разних проблема. Функције рачунара у поменутој области употребљавају се као делови генералног решења, односно као алати у пројектовању софтвера.

За израду задатог софтвера коришћен је Visual FoxPro, у наредном тексту ће бити описан ток креирања апликације. На самом почетку, тј. у другом поглављу је писано о објектном оријентисаном програмирању, док се треће поглавље састоји од описа Visual FoxPro-а и његовог радног окружења.

Затим је у четвртом поглављу писано о формирању класа у Visual FoxPro-у, као и о врсти класа. Након тога следи реализација и опис пројектовања софтвера, као и демонстрација апликације.

2. ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ

Објектно оријентисано програмирање је парадигма програмирања, која користи објекте као основу за пројектовање рачунарских програма и различитих апликација софтвера. Заснива се на различитим техникама, као што су наслеђивање, модуларност, полиморфизам и енкапсулација.

Решавање проблема парадигмом објектно-оријентисаног програмирања, је врло слично људском начину размишљања и решавању проблема. Састоји се од идентификовања објекта и постављање објекта који ће се користити у одговарајућу секвенцу за решење одређеног проблема. Ради се о дизајну објекта чија ће понашања као јединица и у њиховој међусобној интеракцији, решити одређени проблем. Интеракција између објекта се састоји у размени порука, где одређена порука усмерена према одређеном објекту, покреће енкапсулиране операције у том објекту, чиме се решава део обично ширег и сложенијег проблема. Уопштено гледано, објектнооријентисано решавање проблема се састоји из четири корака:

1. идентификовање проблема
2. идентификовање објекта који су потребни за његово решење
3. идентификовање порука које ће објекти међусобно слати и примати
4. креирање секвенце порука објектима, које ће решавати проблем или проблеме.

У парадигми објектно оријентисаног програмирања, објекти су структуре података које представљају одређено и јасно дефинисано знање о спољашњем свету или стварности. Типична организација је у хијерархијске класе, где свака класа објекта поседује информације о особинама објекта које се чувају у инстанцама променљивих и које су повезане (концептом асоцијације) са сваком инстанцом у одређеној класи. Сваки објект препознаје други објект преко његовог интерфејса. Подаци и логика сваког објекта су скривени од других објекта. Тиме се омогућава раздвајање имплементације од понашања објекта у интеракцији са другим објектима.

Основне особине објекта су идентитет, стање и понашање. Идентитет представља назив објекта којим се одређени објект разликује од осталих. Стање објекта је део прошлости и садашњости које одређују понашање објекта у будућности. Понашање објекта је одређено операцијама које се над објектом могу извршити, а активирање операције се врши поруком.

Свака класа се састоји од:

- података – чланова
- објект – члан
- функција чланица – метода

Податак члан и објект члан су атрибути класе, и помоћу њих је описано стање објекта. Објект је модел ентитета а атрибути су есенцијалне особине ентитета које га описују. Методе дефинишу понашање објекта.

2.1. ТЕМЕЉИ ОБЈЕКТНО ОРЈЕНТИСАНОГ ПРОГРАМИРАЊА

1. Апстракција и скривање информација
2. Енкапсулација
3. Модуларност
4. Полиморфизам
5. Везе између класа и наслеђивање

Апстракција је поступак раздвајања битног од небитног. У току овог поступка неопходно је уочити који подаци и везе су битне за дати домен проблема, и оне битне податке имплементирати у класи. Постоје три групе апстракције: апстракција предмета, апстракција процеса и синтетска апстракција (апстракција виртуелне машине).

Скривање података се дефинише на следећи начин: све информације о модулу морају бити скривене осим оних које су експлицитно декларисане као јавне. Непосредна реализација чланова класе мора бити недоступна клијенту.

Енкапсулација је поступак обједињавања стања и понашања у једну цјелину. Крајњи резултат енкапсулације је класа. Други задатак енкапсулације је обезбјеђивање контроле приступа у циљу поштовања принципа скривања информација.

Модуларност се односи на поступак разбијања програма на мање делове који могу аутономно функционисати. Модуларност се примењује како би се омогућила вишеструка употреба софтверских компоненти.

Полиморфизам је контекстно зависно понашање. Програмске категорије које се могу полиморфно понашати су: промјенљиве односно инстанце класе, оператори и методе. Полиморфизам се дели на универзални и ад хок.

Насљеђивање представља могућност хијерархијске организације класа. Када једна класа наследи другу она задржава комплетан садржај класе коју наслеђује, и тај садржај се може редефинисати или проширити.

3. VISUAL FOXPRO

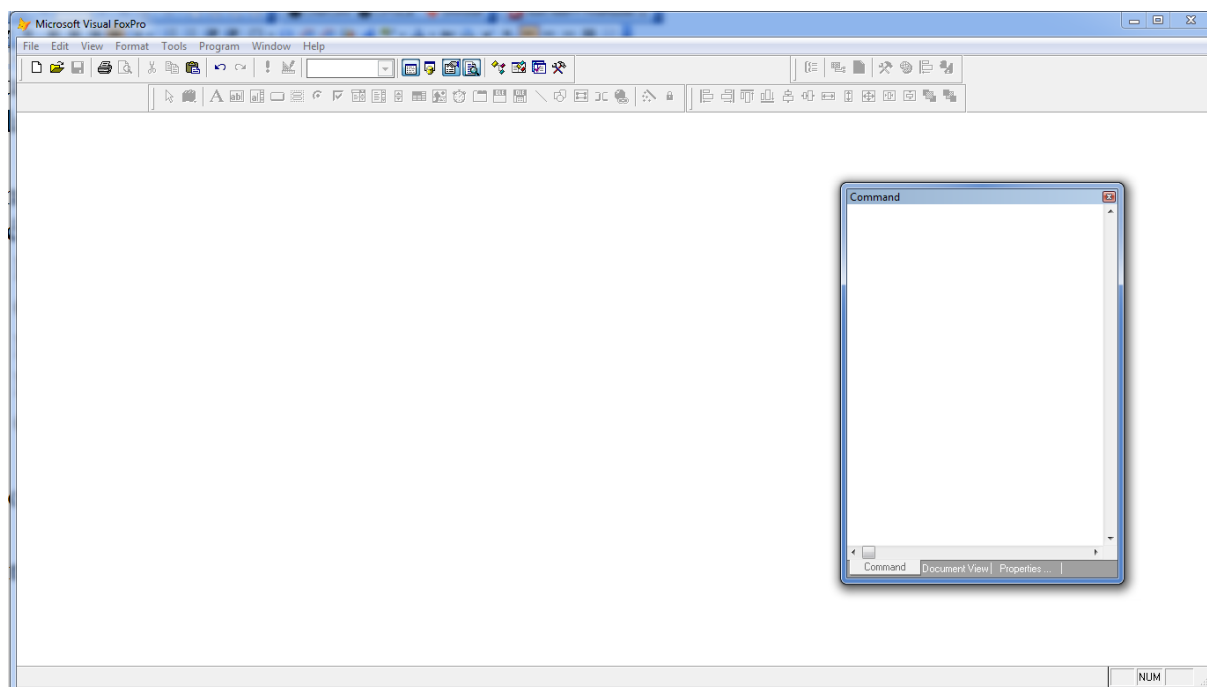
Visual FoxPro је процедурално оријентисан програмски језик и систем за управљање базом података (DBMS). Такође, припада групи објектно оријентисаних програмских језика, који је развила компанија Fox Software, односно компаније Microsoft, од краја 20 века.

Минимална конфигурација која је неопходна да би се инсталирао MS Visual FoxPro 9, је следећа:

- PC са класом Pentium процесора
- Оперативни систем Microsoft Windows XP или каснији, односно Win 7, Win 8, Win 10
- Меморија: 64 MB RAM минимум; препоручује се 128 MB или више;
- Минимум 20 MB слободног простора на хард диску,
- Графичка картица: минимум супер VGA 800 X 600.

3.1. РАДНО ОКРУЖЕЊЕ VISUAL FOXPRO-A

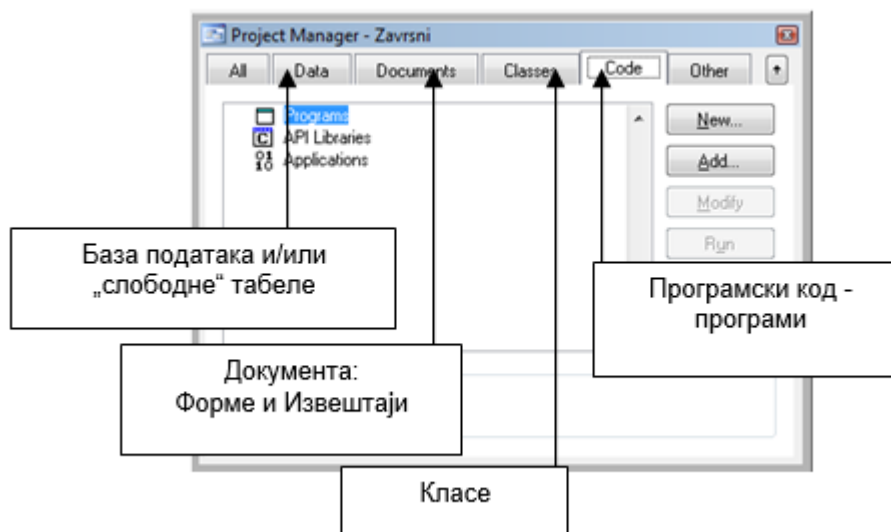
Након стартовања програма, појављује се прозор као што је приказано на слици 11.



Слика 1. Радно окружење Visual FoxPro 9

Систем рада у оквиру Win је заснован на бази пројекта (Project). Сваки пројекат може да садржи следеће објекте:

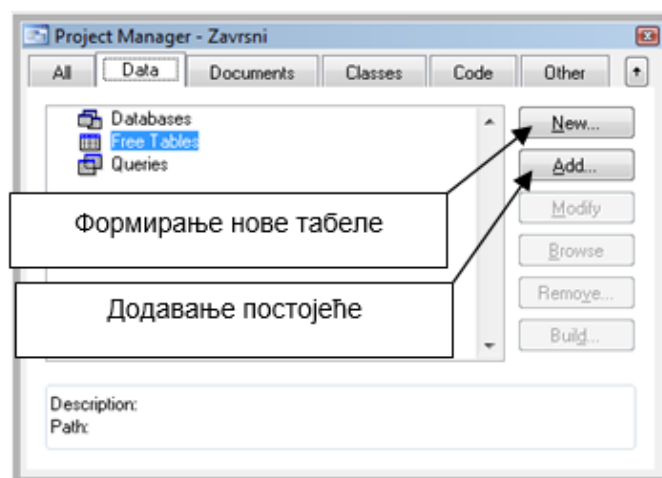
- Податке
 - Базу података и/или „слободне“ табеле.
- Документа
 - Форме (Forms)
 - Извештаје (Reports)
- Програмски код (Code)
- Класе (Classes)



Слика 2. Објекти VisualFox Pro-а 9

3.1.1. ФОРМИРАЊЕ ТАБЕЛЕ

Да би се формирале табеле потребно је активирати картицу Data из пројектног прозора приказаног на слици 13. У оквиру овог модула, могуће је формирати базу података (Databases), „слободне“ табеле (Free tables) или упите (Queries), слика.



Слика 3. Modul za kreiranje tabela ili baza podataka

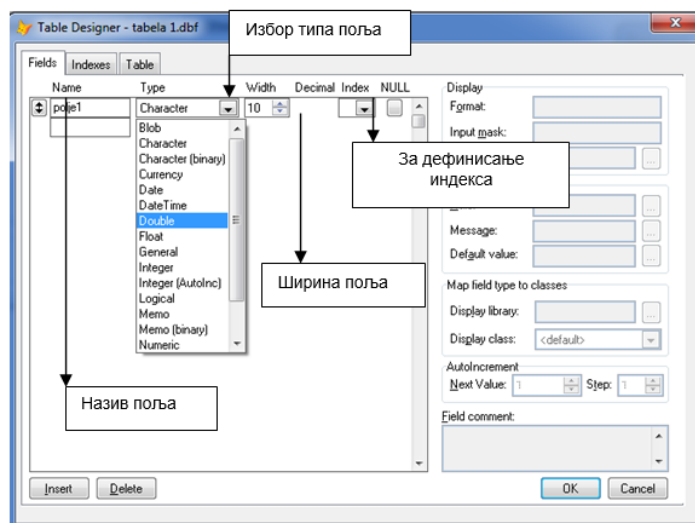
Табеле се могу креирати и упитима, као што ће се урадити и приказати касније у раду, или кликом на дугме New, односно могу се користити већ формиране табеле и укључити у пројекат кликом на дугме Add.

Кликом на дугме New, добиће се дијалог прозор којим корисник може да изабере да ли ће користити помоћ или ће сам креирати табелу.



Слика 4. Дијалог прозор за избор начина креирања табела

Кликом на New Table, корисник прво треба да да име табели, а затим да дефинише називе поља у табели и за свако поље тип и дужину.

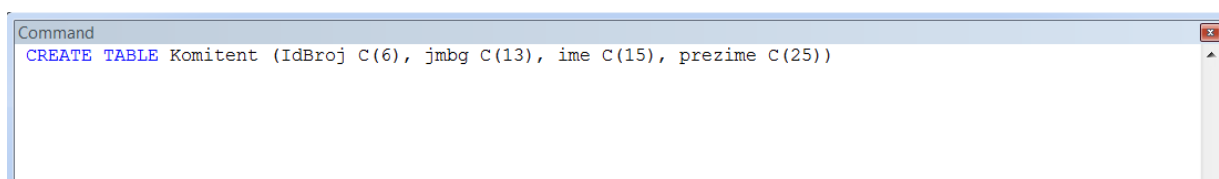


Слика 5. Дијалог прозор за дефинисање структуре табеле

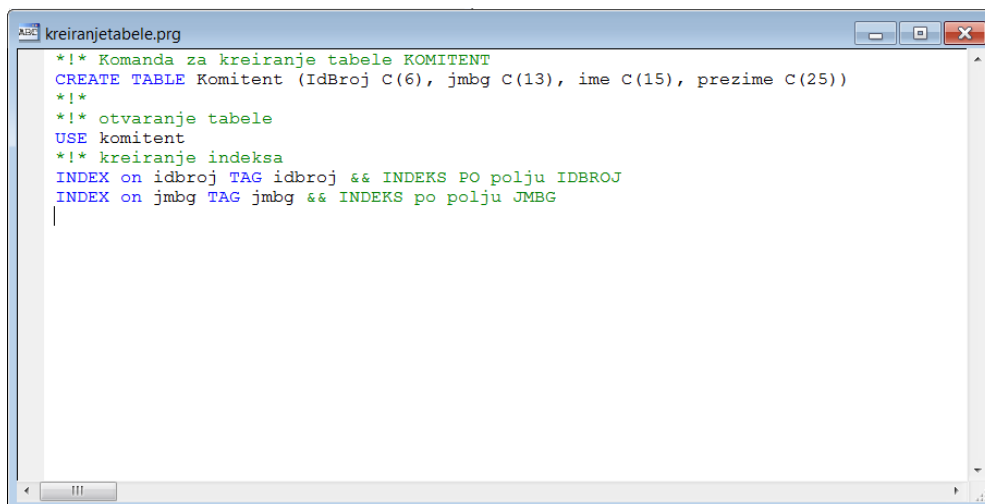
MS Visual FoxPro формира табеле у DBF стандарду, а типови поља који могу да се чувају у овим табелама су:

- Текстуални тип податка (Character), максимална дужина 254 карактера
- Нумерички тип податка:
 - Број – општи број може бити и целобројни и децимални (Нумериц)
 - Децимални број (Float)
 - Валута (Currency) – изведени тип податка из децималног типа са додатком ознаке за валуту
 - Целобројна вредност (Integer)
- Меморијски тип податка (Memo) за чување текстова
- Датумски тип податка (Date).

Табеле се могу формирати и помоћу SQL упита. SQL упит може да се унесе директно у командну линију, слика 18, или да се унесе у едитор текста у оквиру картице code, слика 19.



Слика 6. Унос SQL команде, у командној линији, за креирање табеле комитент.дбф



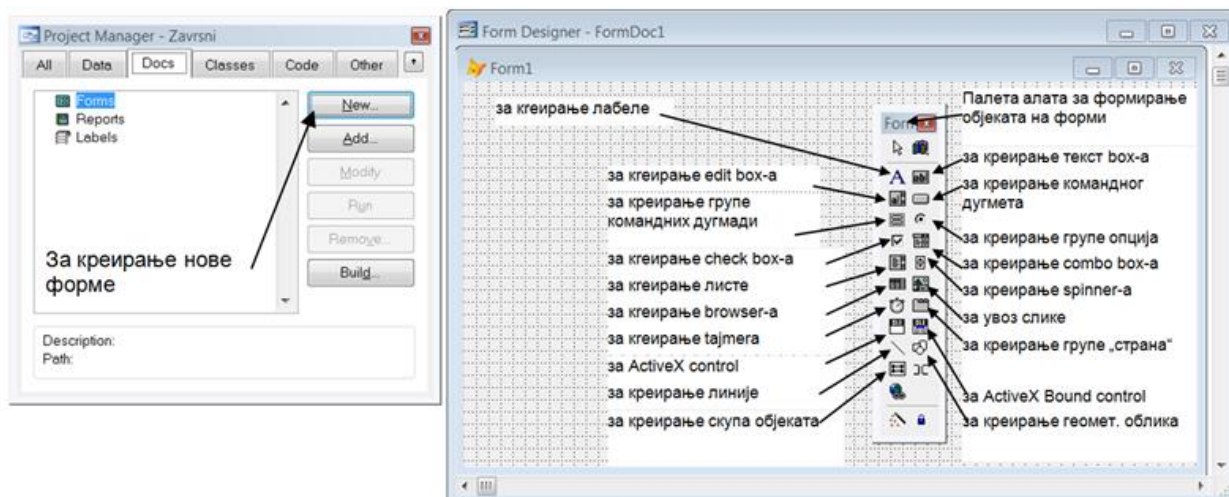
Слика 7. Унос SQL команди, у едитору текста, за креирање табеле komitent.dbf, као и креирање индекса

3.1.2. КРЕИРАЊЕ ФОРМИ

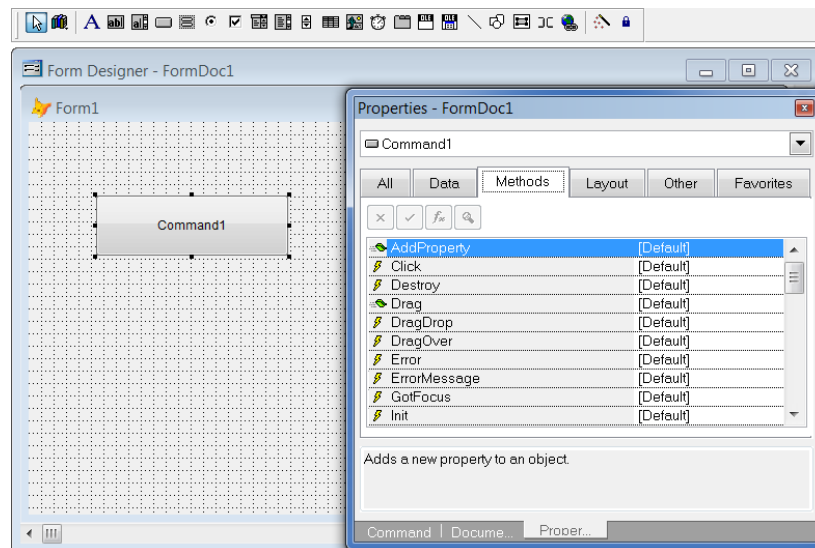
Поред писања програмског кода за израду апликације, која ће радити са креираном базом података, у Visual FoxPro-у може се радити и у графичком окружењу. Ово окружење, омогућава програмеру да практично „црта“ елементе екрана, односно да креира форму, командну дугмад, листе, комбо листе и слично, слика 20. Програмеру остаје само да програмира елементе форме.

Сваки објекат, у форми, који се креира има одређена својства, које је могуће програмирати, слика 21., односно подешавати. Својства су подељена у четири групе, и то:

- подаци (Data),
- методе (Methods),
- изглед (Layout),
- остало (Other).



Слика 8. Окружење за рад са формама



Слика 9. Својства објекта: Data, Methods, Layout, Other

4. ФОРМИРАЊЕ КЛАСА У VISUAL FOXPRO-У

Класе дефинишу особине, методе и догађаје, које представљају карактеристике и функционалност објеката и које се могу користити при манипулацији са објектима. Класе такође могу садржавати друге објекте или чланове података као што су променљиве. Када објекат или инстанца буду креирани из одговарајуће класе, тај објекат садржи својства, методе, догађаје и све друге објекте или чланове, као и класа из које је настао.

Особине које класа дефинише одређује карактеристике за објекат те класе. На пример, класа `CommandButton` има својство `Name` - Име, који дефинише име командног дугмета за разлику од својства `Caption` - Натпис, који дефинише текст који командно дугме приказује кориснику.

Методи класе дефинишу и чувају програмски код поступка који се покреће када се позива метода за објекат те класе. На пример, класа `CommandButton` има метод `Move`, који помера командно дугме на координате које се наводе.

Догађаји класе дефинишу програмски код који се извршава када систем или корисник изведе акције које покрећу догађаје за објекат те класе. На пример, класа `CommandButton` садржи `Click` догађај, што је процедура која се користи за дефинисање програмског кода који се извршава када корисник кликне на командно дугме.

Класе могу такође садржавати и друге објекте, на пример, класа `Form` може да садржи објекат контроле, као што је командно дугме. Такође, могуће је дефинисати чланове података, као што су променљиве, као део класе.

4.1. ВРСТЕ VISUAL FOXPRO КЛАСА

Класе и објекти Visual FoxPro – а сврставају се у два основна типа:

- класа контејнера, и
- класе контроле.

Контејнери могу се састојати од других објеката, као што су контроле, и омогућавају приступ објектима које садрже. Контролне класе су више инкапсулиране него класе контејнера. Међутим, из тог разлога, оне могу бити мање флексибилне. Контролне класе немају `AddObject` методу.

Следећа табела наводи примере објеката који могу постојати у класама контејнера – скупа објеката.

Табела 1. Садржај скупа објеката

Column – колона	Заглавља и сви објекти осим скупова форми, форме, алатних трака, тајмера и других
Command group	Командни дугман (<code>Command button Group</code>)
Container	Свака контрола
Control	Свака контрола
Custom	Било које контроле, окружење података, оквир странице, контејнер, прилагођени (<code>custom</code>)
DataEnvironment	Курсори, односи и курсери
FormSet	Скуп форми, Обрасци, алатне траке

Form	Оквири страница, окружење података, све контроле, контејнери, прилагођени (custom)
Grid	Мрежа
OptionGroup	Опциона дугмад
PageFrame	Стране
Page	Свака контрола, контејнери, прилагођени (custom)
Project	Датотеке, сервери
Toolbar	Било које контроле, оквир странице, контејнер

4.1.1. ОСНОВНЕ КЛАСЕ У VISUAL FOXPRO – У

Visual Fox Pronуди подразумевани скуп класа или базних класа, Табела 2., које је могуће одмах користити да би се обезбедиле основне функције у апликацијама.

Табела 2. Основне класе у Visual FoxPro – у

CheckBox	Collection	Column	ComboBox
CommandButton	CommandGroup	Container	Control
Cursor	CursorAdapter	Custom	DataEnvironment
EditBox	Empty*	Exception	Form
FormSet	Grid	Header	Hyperlink
Image	Label	Line	ListBox
OLE Bound	OLE Container	OptionButton	OptionGroup
Page	PageFrame	ProjectHook	Relation
ReportListener	Separator	Session Object	Shape
Spinner	TextBox	Timer	ToolBar
XMLAdapter	XMLField	XMLTable	

4.1.2. ХИЈЕРАРХИЈСКО РЕФЕРЕНЦИРАЊЕ ОБЈЕКТА

Контејнер и хијерархија класе су два одвојена ентитета. Објекти се референцирају у хијерархији скупа објеката, док Visual FoxPro претражује програмски код догађаја кроз хијерархију класе.

Да би се манипулисало објектом, он се прво мора идентификовати у односу на хијерархију контејнера. На пример, да би се манипулисало контролом у форми (Form) у скупу форми (FormSet), потребно је навести назив скупа форми, форме, а затим и објекта – којим се управља.

На слици 10. приказана је ситуација за референцирање објеката у контејнерској класи. Пример са ове слике у ствари садржи „родитељску“ класу FormSet, и у оквиру ње „дете“ класу Form. Са друге стране, класа Form је „родитељска“ класа свим осталим објектима који су приказани на слици 1.

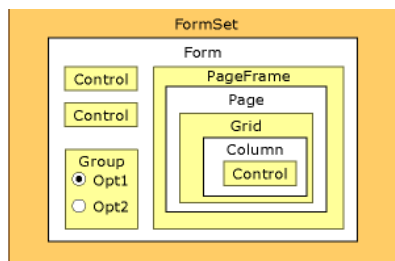
У наставку се даје пример програмских инструкција, (ПК .1) како се позивају и како се манипулише објектима са слике 10.

ПК1. Онемогућавање приступа колони у објекту Grid:

```
Formset.Form.PageFrame.Page.Grid.Column.Control.Enabled = .F.
```

ПК2. Постављање вредности објекта OptionGroup на 1:

```
Formset.Form.Group.Value = 1
```



Слика 10. Хијерахија објеката

4.1.3. РЕЛАТИВНО РЕФЕРЕНЦИРАЊЕ

Када се референцирају објекти из хијерархије контејнера (на пример, у случају Click на командно дугме на форми у сету) могу се користити пречице за идентификовање објекта са којим се манипулише. У табели 3., се наводе својства или кључне речи које олакшавају референцу објекта из хијерархије објекта.

Табела 3. Релативно референцирање

Својсто или кључна реч	Референца
<u>Parent</u>	Непосредни контејнер објекта.
<u>THIS</u>	Објекат
<u>THISFORM</u>	Форма која садржи објекат
<u>THISFORMSET</u>	Скуп форми који садржи објекат

THIS, THISFORM, и THISFORMSET могу се користити само у методи (Method) или код догађаја (Event).

Табела 4, садржи примере коришћења THIS, THISFORM, THISFORMSET и PARENT за постављање особина објекта.

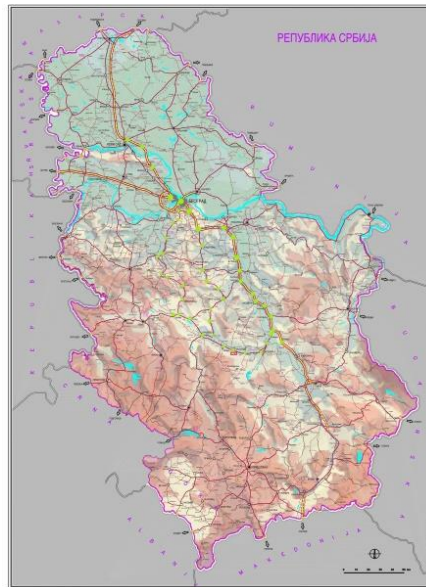
Табела 4. Примери скраћеног референцирања

Команда	Где се примењује команда
THISFORMSET.frm1.cmd1.Caption = "OK"	Код догађаја или метода код било које контроле на било којој форми у скупу форми – formsets.
THISFORM.cmd1.Caption = "OK"	У случају догађаја или метода код било које контроле на истој форми на којој је cmd1 укључен.
THIS.Caption = "OK"	У случају догађаја или метода код контроле чији се назив мења.
THIS.Parent.BackColor = RGB(192,0,0)	У случају догађаја или метода код контроле на форми. Команда мења боју позадине форме у тамно црвену боју.

5. РЕАЛИЗАЦИЈА СОФТВЕРА

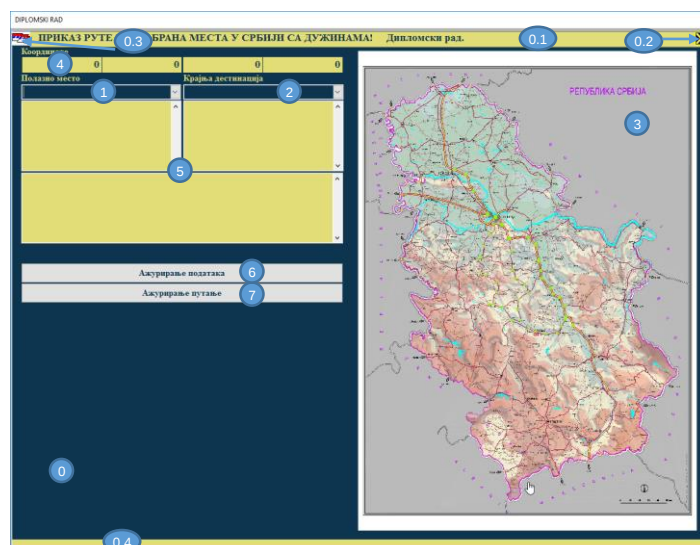
5.1. ДИЗАЈНИРАЊЕ АПЛИКАЦИЈЕ

Основу дизајна апликације за одређивање путање и растојања, између два изабрана града у Србији, чини мапа Србије, са путним правцима, слика 11. На мапи могу да буду дефинисане тачке као идентификатори градова и или места, али за саму апликацију ти идентификатори нису битни, јер као што ће се видети даље у раду, у класи мапа дефинисаће се тагови за свако изабрано место.



Слика 11. Мапа Р. Србије са уцртаним путним правцима

Да би се одредила путања – путни правац, потребно је изабрати два места, место од кога (1) и место до кога (2) се одређује путања и растојање. У ту сврху додата су два објекта класе ComboBox (1 и 2), слика 12.



Слика 12. Изглед апликације

Са слике 3, уочава се да, у главној форми (0), постоје поред наведених објеката (1,2 и 3), постоје и следећи објекти:

- Лабела заглавље (0.1) – за исписивање наслова форме.
- Слика (0.2) – са програмираним догађајем Click за прекид рада.
- Слика (0.3) – застава Р. Србије.
- Лабела статусна линија (0.4) – за помоћ при раду, исписивање ToolTip-а.
- Група текст боксева (4) за исписивање координата (KoordX1, KoordY1, KoordX2, KoordY2), приликом зумирања делова на мапи.
- Група едит боксева (5) за исписивање података о изабраним местима.
- Командно дугме (6) за ажурирање података о местима.
- Командно дугме (7) за ажурирање путање.
- Контејнер за унос података о местима (8), слика 13, појављује се када се кликне на командно дугме (6).
- Контејнер за дефинисање путање (9), слика 13, појављује се када се кликне на командно дугме (7).
- Контејнер за приказивање путање и растојања између изабраних места (10), слика 4, појављује се када се изаберу места у комбо боксевима (1) и (2)

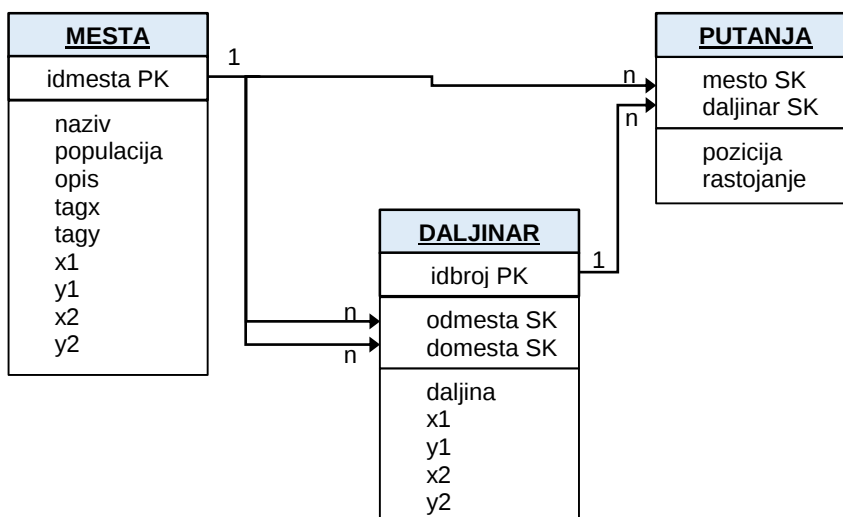
Апликација кориснику мора да омогући да унесе места (назив, карактеристике, и локацију на увеличаном делу мапе), као и да дефинише путању од места до места, слика 14. У ту сврху у Visual FoxPro – у дефинисана је база података са три табеле: места, даљинар и путања, (Табела 5 и Слика 15), односно за сваки од објеката приказаних на слици 12, дефинисане су класе које су служиле за формирање апликације. Класе су формиране писањем програмског кода као и коришћењем визуелних алатки које Visual FoxPro у себи садржи.

Контејнер за ажурирање података о местима

Контејнер за ажурирање путање

Контејнер за приказивање путање

Слика 13. Класе контејнера



Слика 15. Релационо повезане табеле

5.2. ПРОЈЕКТОВАЊЕ АПЛИКАЦИЈЕ

Као што је раније у тексту напомењено, за израду ове апликације коришћен је Visual FoxPro 9. Иако у себи има уграђене визуелне алатке за креирање елемената апликације, већина програмског кода је исписана, у едитору, на класичан начин поштујући синтаксу и правила писања програмских инструкција.

Овде је битно напоменути да је у овај пројекат укључена библиотека класа GDI+. Ова класа је укључена у пројекат јер омогућава манипулацију са графичким објектима, као што је зумирање слика у изабраном региону и исртање графичких објеката преко других графичких објеката.

Комплет функција у оквиру GDI+, заправо, се јавља у неколико облика. У .NET окружењу постоје System.Drawing и његови повезани именовани простори. У оквиру C++ постоји скоро идентичан API дефинисан у gdiplus.h. Међутим, интерфејс који обезбеђује GDIPLUS.DLL (доступан у Visual FoxPro преко API позива) је заправо не-објектно оријентисан интерфејс познат као "GDI+ Flat API".

Да би се олакшало рад са GDI+, нова VFP 9.0 FFC библиотека, _GDIPLUS.VCX, пружа објектно оријентисан интерфејс који је врло сличан са NET System.Drawing именским простором. То значи да се књиге, чланци и референтни материјал као што је MSDN, оригинално написан за програме .NET и C++, лако могу применити на програмирање помоћу FFC библиотеке уз одговарајућа прилагођавања.

Та прилагођавања се састоје у следећем:

- Сва имена класе имају префикс са словима "GP".
- Неки називи својстава су различити (на пример, PenWidth уместо Width).
- Константе су декларисане као #define константе са префиксом "GDIPLUS_" уместо набрајања.

У табели 6., приказане су класе дефинисане у _GDIPLUS.VCX, у поређењу са еквивалентним .NET класама. У табели су, у осенченим редовима, означене класе које су коришћене и модификоване у овој апликацији.

Табела 6. Упоредни приказ FFC и .NET класе

FFC класе	Опис	.NET класе
GpGraphics	Цртање површина (canvas) и операције цртања.	Graphics
GpColor	Дефинисање боје са RGB (red, green, blue) и alpha компонентама.	Color
GpPoint	Лоцирање у 2-D равни (x, y).	Point, PointF
GpSize	Димензије (ширина, висина).	Size, SizeF
GpRectangle	Локација и димензије.	Rectangle, RectangleF
GpPen	За цртање линија и кривих.	Pen
GpBrush	Апстрактна база класа алатке „четкица“.	Brush
GpSolidBrush	Испуњавање бојом ограниченог региона - површине.	SolidBrush
GpHatchBrush	Испуњавање шрафуром ограниченог региона - површине.	HatchBrush
GpFontFamily	Описује породицу сродних фонтова (на пример, "Arial").	FontFamily
GpFont	Описује изабрани фонт укачујући величину дебљину, декорацију. Користи се за израду текста.	Font
GpStringFormat	Распоред текста и информације о формату као што је поравнање.	StringFormat
GpImage	Основна класа за слике (укључује векторске и битмап формате).	Image
GpBitmap	Слика се састоји од 2 – D низа пиксела.	Bitmap

5.2.1. ГЛАВНИ ПРОГРАМ

Као што је напоменуто, већина програмског кода је исписана, а као и у већини осталих софтвера прво се пише главни програм из кога се позивају све остале процедуре и или функције. Главни програм је дат у ПК 3.

ПК3. Главни програм

```

1. PUBLIC odiplomski_rad, putklasa, putffc, oObjekat, oldCaption,;
   OldIcon, oldWidth, oldHeight
2. LOCAL startni_folder, wputanja,i,wp
3. SET DATE GERMAN
4. SET CENTURY ON
5. SET STATUS bar OFF
6. SET SCOREBOARD OFF
7. SET SYSMENU OFF
8. SET TALK OFF
9. SET STATUS OFF
10. SET CONFIRM ON
11. SET DELETED ON
12. SET HELP OFF
13. SET EXCLUSIVE OFF
14. SET ESCAPE OFF
15. CLOSE DATABASES
16. startni_folder=SYS(5)+SYS(2003)
17. putklasa=startni_folder+"\klase\mojaklasa"
18. putffc=startni_folder+"\ffc\_gdipus"

```

```

19. SET CLASSLIB TO "&putklasa"
20. wp=startni_folder++"\tabele,"
21. putanja=startni_folder+", "+wp+startni_folder+"\prg,"+;
    startni_folder+"\slike,"+startni_folder+"\forme,"+startni_folde
    r+;"\izvestaji,"+startni_folder+"\klase,"+startni_folder+"\txt"
22. SET PATH TO &putanja
23. oObjekat=CREATEOBJECT('custom')
24. oObjekat.addproperty('VratiSe(100)', '')
25. oObjekat.addproperty('Zatvori(100)', '')
26. oObjekat.addproperty('OdMesta', '')
27. oObjekat.addproperty('DoMesta', '')
28. oObjekat.addproperty('Zumirano', .f.)
29. oObjekat.addproperty('iListaOd', 1)
30. oObjekat.addproperty('iListaDo', 1)
31. oObjekat.addproperty('IzborOd', "")
32. oObjekat.addproperty('IzborDo', "")
33. oObjekat.addproperty('Odbroj', '')
34. oObjekat.addproperty('Pozicija', 0)
35. FOR i=1 TO 100
36.     oObjekat.VratiSe[i]='N'
37.     oObjekat.Zatvori[i]='N'
38. ENDFOR
39. WITH _Screen
40. oldCaption=.caption
41.     OldIcon=.icon
42.     oldWidth=.width
43.     oldHeight=.Height
44.     .controlbox=.f.
45.     .width = 1260
46.     .height = 950
47.     .autocenter = .t.
48.     .Caption='DIPLOMSKI RAD'
49.     .icon='&startni_folder\ikonice\zastava.ico'
50. ENDWITH
51. odiplomski_rad=NEWOBJECT("diplomski_rad")
52. odiplomski_rad.Show
53. READ EVENTS
54. RETURN

```

Линије 1 и 2 из ПК 3 декларишу глобалне (доступне у свим процедурама које се позивају из главног програма) као и локалне променљиве. У линијама 3 до 14 су нека од стандардних подешавања VFP – а. Основни скуп класа који ће се користити у овој апликацији дефинисан је путањом датом у линији 17, глобална променљива *putklasa*, односно дата је путања до класе *ffc*, такође глобалном променљивом под називом *putffc*.

У линијама од 23 до 34, креиран је глобални објекат – *oobjekat*, на бази класе *custom*. У ствари, овај објекат је дефинисан да у себи садржи скуп променљивих, па чак и низова који ће бити доступни у свим процедурама и функцијама апликације.

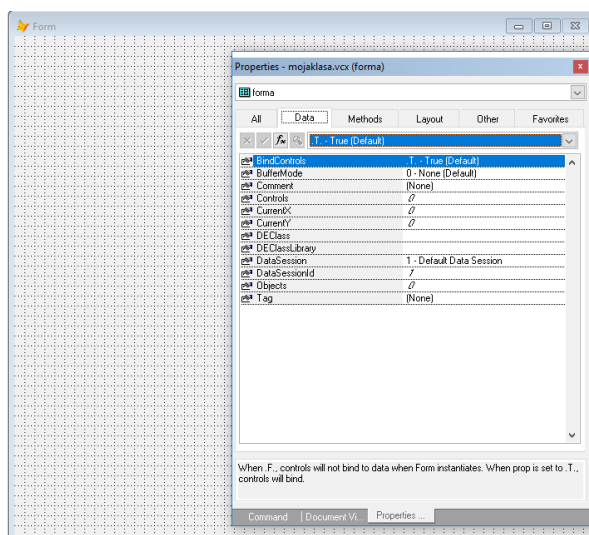
У линији 51 формиран је нов објекат под називом *odiplomski_rad* на бази класе *diplomski_rad*, а у чијој је основи класа *form*.

Главни програм покреће све дефинисане методе у објекту *odiplomski_rad* након извршења програмских редова 52 и 53.

5.2.2. ПРОГРАМСКИ КОДОВИ

У Visual FoxPro – у основу апликације чине форме (енг. Form). У формама се смештају сви остали објекти који чине саставне делове апликације. Основна класа форми, слика 16., садржи објекте, методе и својства подељене у четири категорије:

- Подаци (Data)
- Методе (Methods)
- Изглед (Layouts)
- Остало (Others)



Слика 16. Класа Form

Приликом дефинисања нове класе, из класе Form, могу се додавати нове процедуре, променљиве и нова својства, чије се вредности, касније току израде апликације или приликом рада са апликацијом, могу мењати.

У ПК 4. дат је део програмског кода за дефинисање форме апликације.

ПК 4. Програмски код за формирање форме апликације

```
1.  DEFINE CLASS diplomski_rad as Form
2.      DataSession=1
3.      Height=941
4.      Width=1255
5.      DoCreate=.t.
6.      ShowTips = .T.
7.      AutoCenter = .T.
8.      ShowWindow=1
9.      allowoutput=.f.
10.     Caption = "ДИПЛОМСКИ РАД - Марија Мојсиловић"
11.     ControlBox = .F.
12.     Closable = .F.
13.     FontName = "Times New Roman"
14.     FontCharSet = 204
15.     MaxButton = .F.
16.     MinButton = .F.
17.     Icon = "..\ikonice\zastava.ico"
18.     TitleBar = 0
19.     AlwaysOnTop = .T.
20.     BackColor = RGB(13,53,79)
```

```

21.      Name = "diplomski_rad"
      *-- Dodavanje objekata na bayi vizuelno definisanih klasa
22.      ADD OBJECT agradova as azuriranjeogradova
23.      ADD OBJECT aputanja as azuriranjeputanje
24.      ADD OBJECT listamesta as putanje
      *-- Dodavanje objekata na bayi vizuelno definisanih klasa
25.      ADD OBJECT karta as DefSlika
      *-- TEXT BOX za prikazivanje koordinate X i Y za zoom
26.      ADD OBJECT KoordX1 as textbox;
27.      WITH ;
28.          fontname = "Times New Roman",;
29.          fontsize = 14,;
30.          BackStyle = 1, ;
31.          BorderStyle = 0, ;
32.          fontBold = .t.,;
33.          BackColor = RGB(226,221,118),;
34.          ForeColor = RGB(13,53,79),;
35.          DisabledBackColor = RGB(226,221,118),;
36.          DisabledForeColor = RGB(13,53,79),;
37.          FontCharSet = 204,;
38.          Value=0,;
39.          Left = 18,;
40.          Height = 28,;
41.          Top = 47,;
42.          Width = 143,;
43.          Enabled = .f.,;
44.          Name = "KoordX1"
45.      ADD OBJECT KoordY1 as textbox;
46.      WITH ;
47.          ...
48.      ADD OBJECT KoordX2 as textbox;
49.      WITH ;
50.          ...
51.      ADD OBJECT KoordY1 as textbox;
52.      WITH ;
53.          ...
54.      ADD OBJECT zaglavlje as label ;
55.      WITH ;
56.          fontname = "Times New Roman",;
57.          fontsize = 14,;
58.          BackStyle = 1, ;
59.          BorderStyle = 0, ;
60.          fontBold = .t.,;
61.          BackColor = RGB(226,221,118),;
62.          ForeColor = RGB(13,53,79),;
63.          FontCharSet = 204,;
64.          Caption = ' ',;
65.          Left = 308,;
66.          Height = 28,;
67.          Top = 1,;
68.          Width = 209,;
69.          Name = "Zaglavlje"

```

У линији 25 дефинисан је објекат *karta* на бази класе *DefSlika*. У програмском коду ПК 5 у целиности је дат код за формирање класе *DefSlika* у коме је укључена и класа *ffc*.

ПК 5. Програмски код за формирање класе *DefSlika*

```

1.  DEFINE CLASS DefSlika As Image
2.      stretch=1
3.      nState=0
4.      PROCEDURE mousemove(nBut,nShift,x,y)
5.          IF this.nState=1
6.              IF x > thisform.Okvir.Left
7.                  thisform.Okvir.width=x - thisform.Okvir.Left
8.              ENDIF
9.              IF y > thisform.Okvir.Top
10.                  thisform.Okvir.height = y-thisform.Okvir.Top
11.              ENDIF
12.          ENDIF
13.      ENDPROC
14.      PROCEDURE mousedown(nBut,nShift,x,y)
15.          IF TYPE('x')='L' OR TYPE('y')='L'
16.              IF TYPE('x')='L'
17.                  x=615
18.              ENDIF
19.              IF TYPE('y')='L'
20.                  y=40
21.              ENDIF
22.          ENDIF
23.          IF TYPE('x')# 'L' AND TYPE('y')# 'L'
24.              IF NOT thisform.aputanja.visible AND NOT;
                thisform.agradova.visible
25.                  IF x>=610 AND x <=1246 AND y>=36 AND y<=1246
26.                      this.Parent.KoordX1.value=x
27.                      this.Parent.KoordY1.value=y
28.                      IF this.nState>0
29.                          thisform.Okvir.visible=0
30.                          this.picture=this.picture
31.                          this.nState=0
32.                          RETURN
33.                      ENDIF
34.                      IF VARTYPE(thisform.Okvir)='U'
35.                          thisform.AddObject("Okvir","Pravougaonik")
36.                      ENDIF
37.                      thisform.Okvir.top=y
38.                      thisform.Okvir.left = x
39.                      thisform.Okvir.height=1
40.                      thisform.Okvir.width=1
41.                      thisform.Okvir.visible=1
42.                      this.nState=1
43.                  ENDIF
44.              ELSE
45.                  IF x>=610 AND x <=1246 AND y>=36 AND y<=1246
46.                      thisform.agradova.tagX.value=x
47.                      thisform.agradova.tagy.value=y
48.                  ENDIF
49.              ENDIF
50.          ENDIF
51.      ENDPROC

52.      PROCEDURE mouseup(nBut,nShift,x,y)
53.          IF NOT thisform.aputanja.visible AND NOT
thisform.agradova.visible
54.              this.Parent.KoordX2.value=x
55.              this.Parent.KoordY2.value=y
56.              IF this.nState#1
57.                  RETURN
58.              ENDIF
59.              this.nState =2

```

```

60.         thisform.Okvir.visible=.f.
61.         thisform.karta.picture=thisform.karta.picture
62.         SET CLASSLIB TO "&putffc"
63.         LOCAL oGraphics as gpGraphics OF ("&putffc")
64.         LOCAL oRectDest as gpRectangle OF ("&putffc")
65.         LOCAL oImage as gpImage OF ("&putffc")
66.
        oImage=CREATEOBJECT("gpImage",thisform.Karta.picture)
67.         oGraphics=CREATEOBJECT("gpgraphics")
68.         wRatio=oImage.ImageHeight/oImage.
69.         pRatio=thisform.Karta.Height/thisform.Karta.Width
70.         IF wRatio>pRatio
71.             wwidth=thisform.Karta.height/wRatio
72.
            wleft=thisform.Karta.left+(thisform.Karta.width;
                - wwidth)/2
73.             wtop=thisform.Karta.top
74.             wheight=thisform.Karta.height
75.             oRectSrc=CREATEOBJECT("gpRectangle", (;
                thisform.Okvir.left-wleft)*;
                oImage.ImageWidth/wWidth, ;
                (thisform.Okvir.top-
                    thisform.Karta.top);
                    *oImage.ImageHeight;
                    /thisform.Karta.height, ;
                    thisform.Okvir.width*;
                    oImage.ImageWidth/wWidth, ;
                    thisform.Okvir.Height * ;

                oImage.ImageHeight/thisform.Karta.height)
76.             ELSE
77.                 wwidth=thisform.Karta.width
78.                 wleft=thisform.Karta.left
79.                 wheight=thisform.Karta.width*wRatio
80.                 wTop=thisform.Karta.top+(thisform.Karta.height-
                    ;
                    wheight)/2
81.                 oRectSrc=CREATEOBJECT("gpRectangle",;
                    thisform.Okvir.left * ;

                    oImage.ImageWidth/thisform.Karta.width, ;
                    (thisform.Okvir.top-wtop)*;
                    oImage.ImageHeight/wheight, ;
                    thisform.Okvir.width *;

                    oImage.ImageWidth/thisform.Karta.width, ;
                    thisform.Okvir.Height * ;
                    oImage.ImageHeight/wheight)
82.             ENDIF
83.             oRectDest=CREATEOBJECT("gpRectangle",wleft,;
                wtop,wwidth,wHeight)
84.             hWnd=SYS(2327, SYS(2325, SYS(2326,THISFORM.hwnd)))
85.             oGraphics.CreateFromHWND(hWnd)
86.             oGraphics.DrawImagePortionScaled(oImage, ;
                oRectDest,oRectSrc,2)
87.             ValidateRect(hWnd,0)
88.             oObjekat.Zumirano=.t.
89.             SET CLASSLIB TO "&putklasa"
90.         ENDIF
91.     ENDPROC
92. ENDDEFINE

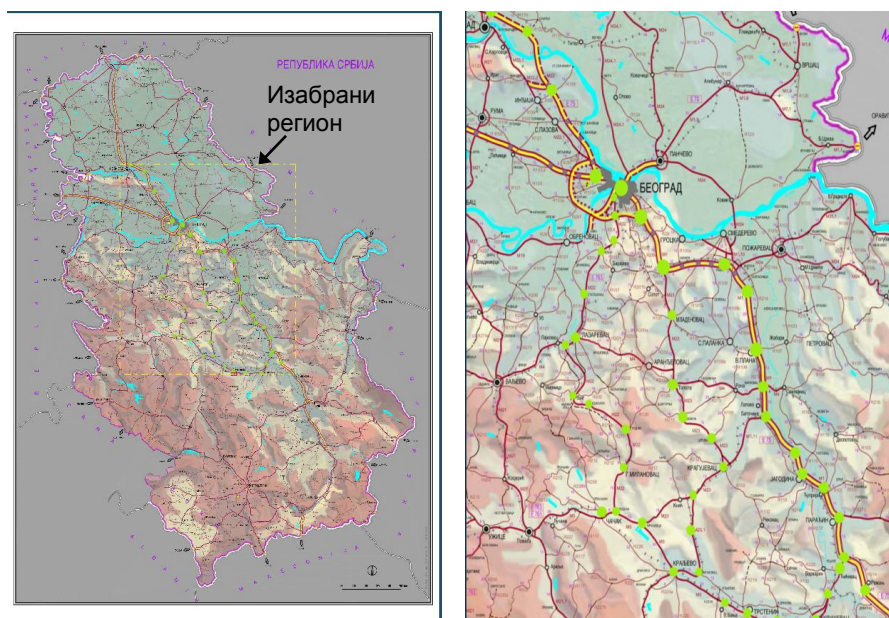
```

Класа *DefSlika* у себи има дефинисане три процедуре – методе, и то *mousemove*, *mousedown* и *mouseup*.

У оквиру процедуре *mousemove*, линије 4 – 13, у зависности од тога да ли је зумирана слика (изабран регион увећавања), на шта указује вредност променљиве *nstate* (=1), а у зависности од вредности позиције курсора *x* и *y*, заузимају се вредности висине (*thisform.Okvir.height*) и ширине правоугаоника (*thisform.Okvir.width*) којим се зумира одређена област на слици.

Процедуром *mousedown* (линије 14 – 51) уколико се не ажурирају подаци о местима или путањи, односно уколико већ није увећан део слике, додаје се објекат *okvir* линије 34 – 36 са иницијалним димеизијама (*.height* = 1 и *.width* = 1), односно позицијом дефинисаном тренутном позицијом курсора у тренутка клика на слику, слика 17.

Дефинисање региона који се увећава се окончава процедуром *mouseup* (линије 52 – 91). И у овој процедури се испитује да ли је у току неки од процеса ажурирања линија 53., односно да ли је део слике већ увећан, линија 56. Уколико није ништа од овога наведеног, креће се у процедуру позивања функција из класе *ffc*.



Слика 17. Формирање региона и ефекат зумирања процедурама *mousedown* и *mouseup*

Прво се поставља путања на библиотеку класа *ffc* линија 62., а затим се креирају графички објекти *oGraphics*, *oRectDest* и *oImage*, линије 63 – 65. У ствари, линије 66 и 67, омогућавају коришћење функција за манипулацију графичким објектима и објектима класа *image*.

На сличан начин се исцртава путања. Приликом дефинисања форме, у изведеној класи *diplomski_rad*, дефинисана је процедура *CrtajPutanju*, чији је програмски код дат у ПК 6.

ПК 6. Процедура за исцртавање путање

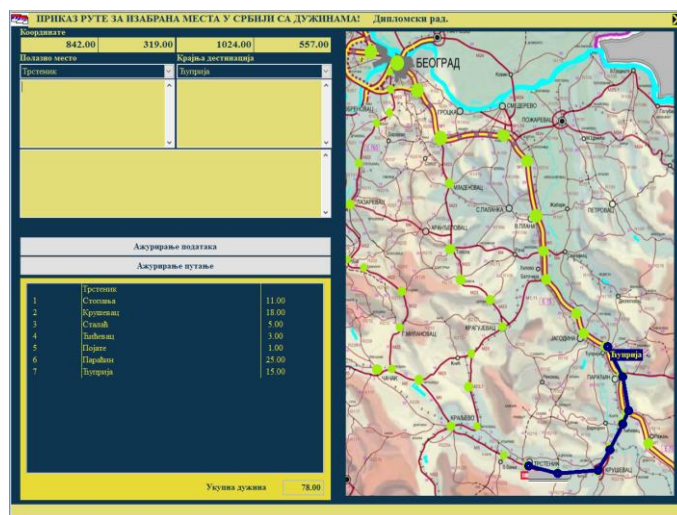
```
1.  PROCEDURE CrtajPutanju
2.    local oGr as GpGraphics of ("%putffc")
3.    oGr = newobject('GpGraphics','&putffc')
4.    oGr.CreateFromHwnd( Thisform.HWnd )
5.    oLineColor = newobject('GpColor','&putffc','', 0,0,128 )
6.    oPen = newobject('GpPen', '&putffc' )
7.    oPen.Create( m.oLineColor, 7 )
8.    LOCAL x_1,y_1,x_2,y_2,i,j, ntm, widm
```

```

9.      SELECT mesta
10.     ntm=TAG()
11.     SET ORDER TO idmesta
12.     FOR i=1 TO thisform.listamesta.putanja.listcount-1
13.         widm=thisform.listamesta.putanja.list(i,3)
14.         SEEK widm
15.         IF FOUND()
16.             x_1=tagx
17.             y_1=tagy
18.         ELSE
19.             x_1=99999
20.             y_1=99999
21.         ENDIF
22.         widm=thisform.listamesta.putanja.list(i+1,3)
23.         SEEK widm
24.         IF FOUND()
25.             x_2=tagx
26.             y_2=tagy
27.         ELSE
28.             x_2=99999
29.             y_2=99999
30.         ENDIF
31.         IF x_1<> 99999 AND x_2<> 99999
32.             oGr.DrawEllipse(m.Open,x_1-5,y_1-5,10,10)
33.             IF i=thisform.listamesta.putanja.listcount-1
34.                 oGr.DrawEllipse(m.Open,x_2-5,y_2-5,10,10)
35.                 thisform.tagmesta2.left=x_2+5
36.                 thisform.tagmesta2.top=y_2+5
37.                 thisform.tagmesta2.visible=.t.
38.                 thisform.tagmesta2.caption=;
39.                 ALLTRIM(thisform.listamesta.putanja.list(i+1,4))
40.             ENDIF
41.             oGr.DrawLine(m.Open, x_1,y_1,x_2,y_2)
42.         ENDIF
43.     ENDFOR
44.     SELECT mesta
45.     SET ORDER to &ntm
46. ENDPROC

```

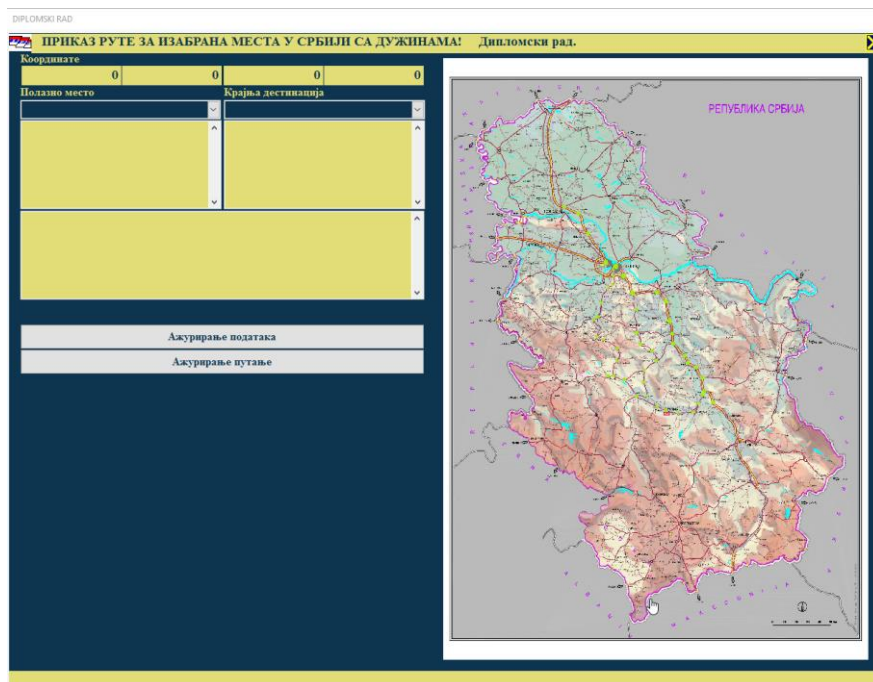
У програмским линијама 2, 3, 4, 5, 6, 7, 8, 32, 34 и 40 позивају се функције из објекта *GpGraphics* класе *ffc*. Да би се исцртала путања потребно је претражити табелу *mesta* да би се преузеле вредности координата таг-ова (*tagx* и *tagy*). Која су места, у путањи, даје нам листа *putanja* у објекту *listamesta*, слика 18., која се попуњава на основу избора полазног места – комбо бокс 1 и избора крајње дестинације – комбо бокс 2, слика 12.

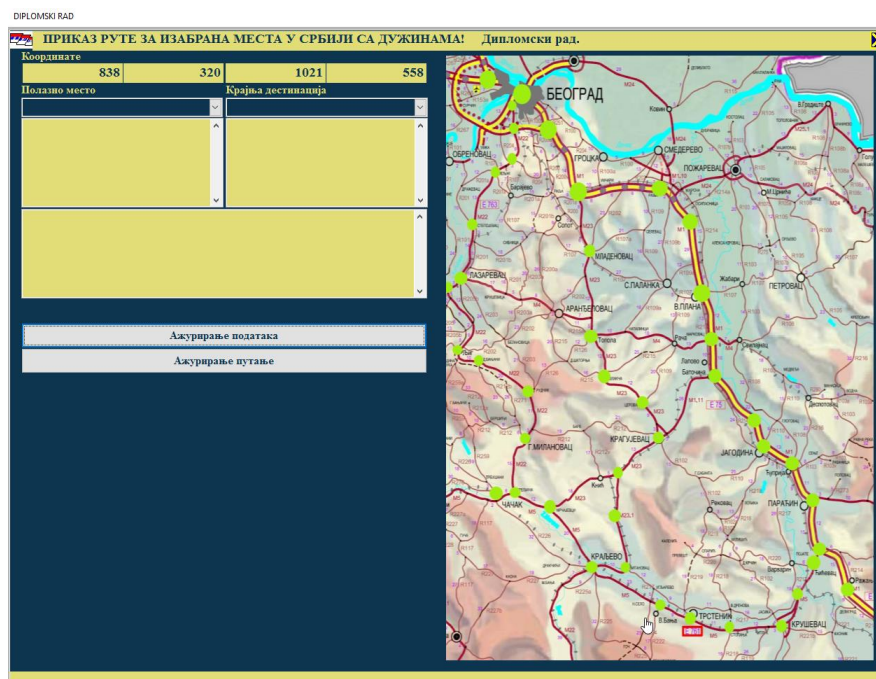


5.2.3. ДЕМОНСТРАЦИЈА РАДА ПРОГРАМА

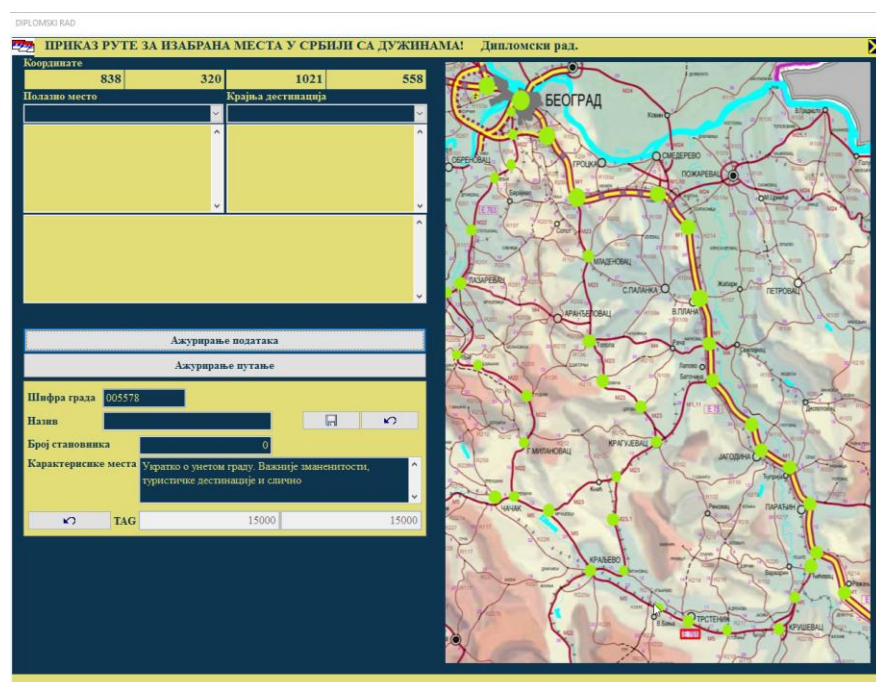
Апликација се покреће стартовањем *diplomski_rad.exe*. Да би се покренула Visual FoxPro апликација неопходно је у у њеном фолдеру буду и Fox – ове *dll* библиотеке *gdiplus.dll*, *msvcr71.dll*, *vfp9r.dll* и *vfp9renu.dll*. Ово у ствари значи да ова апликација са поменутиим библиотекама може да се покрене и на другим рачунарима, а на којима није инсталиран Visual FoxPro.

Даље у тексту сликама од 19 до 26 је демонстриран рад програма.

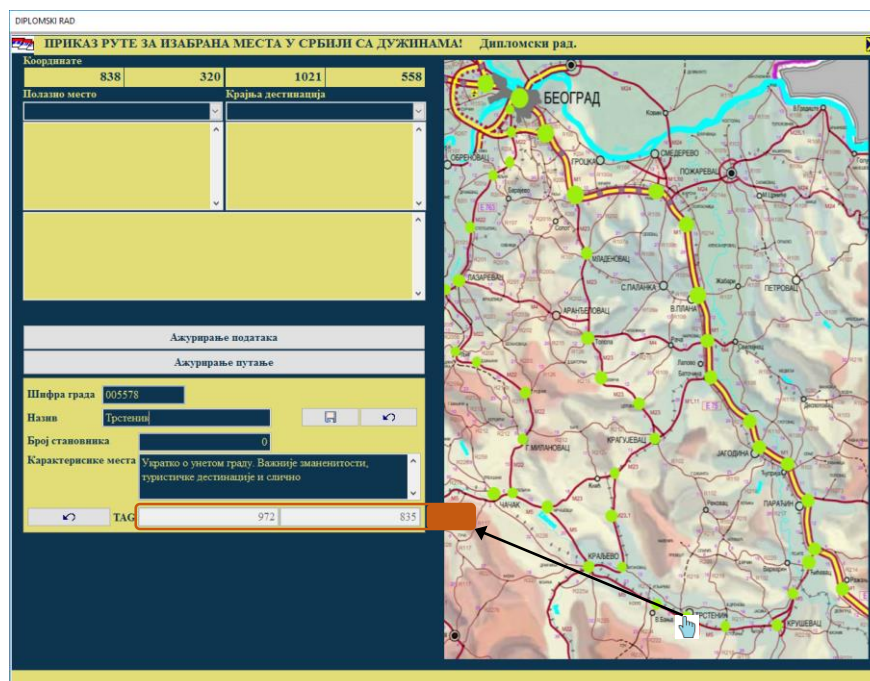




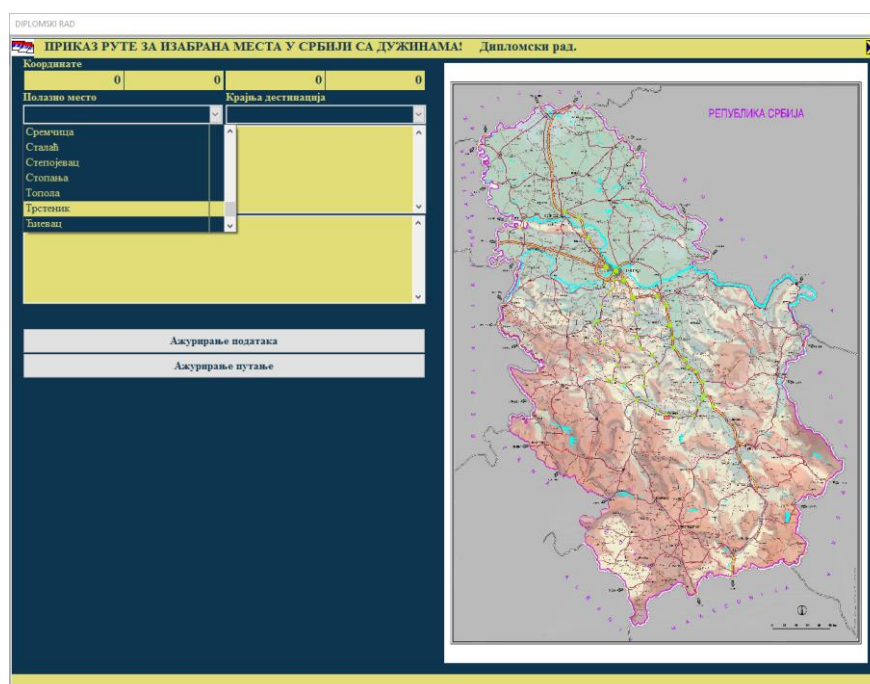
Слика 20. Зумирање региона мапе



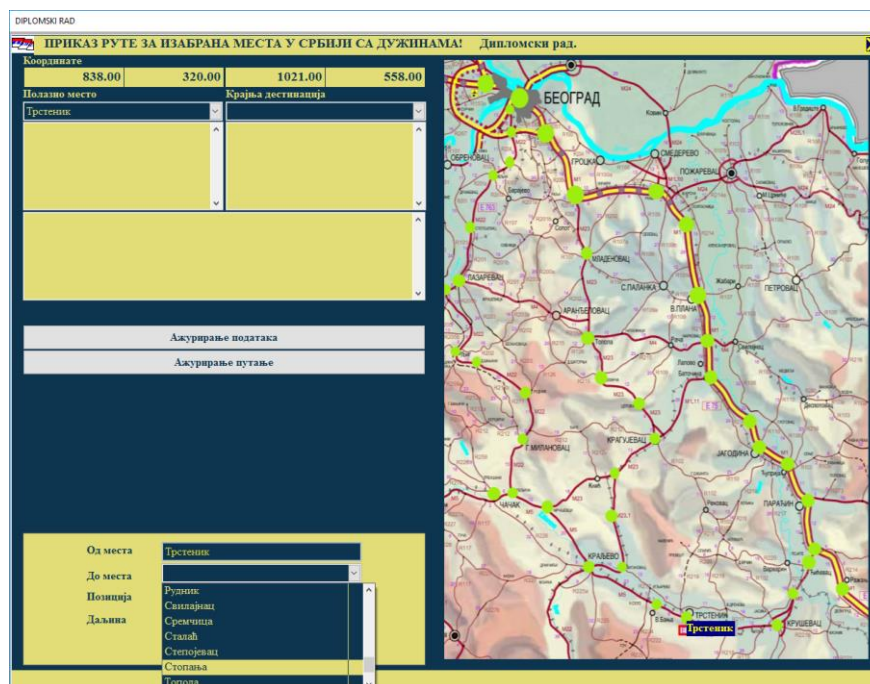
Слика 21. Унос места – додавање



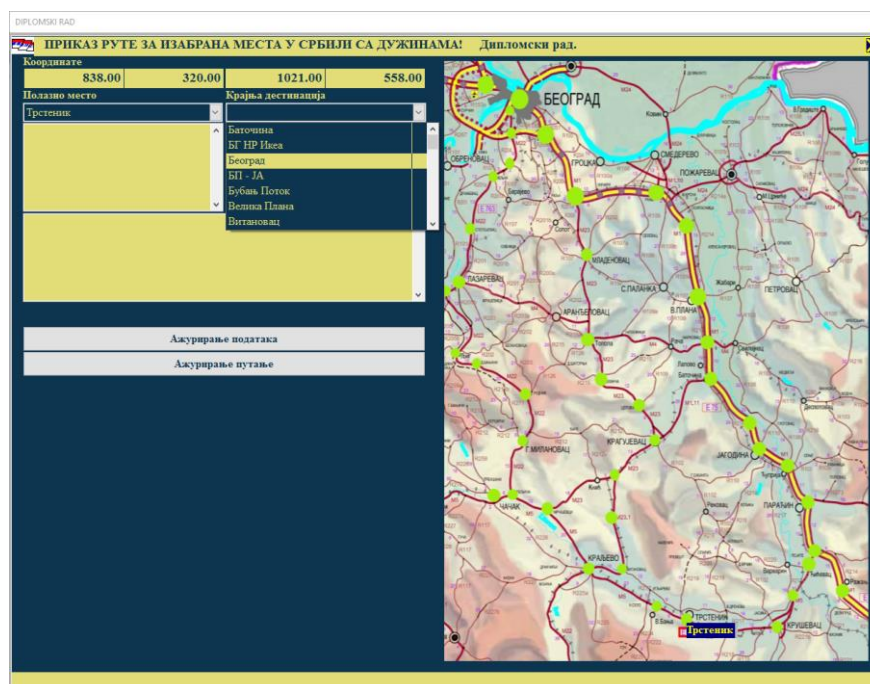
Слика 22. Додељивање позиције на увеличаном делу мапе - таговање



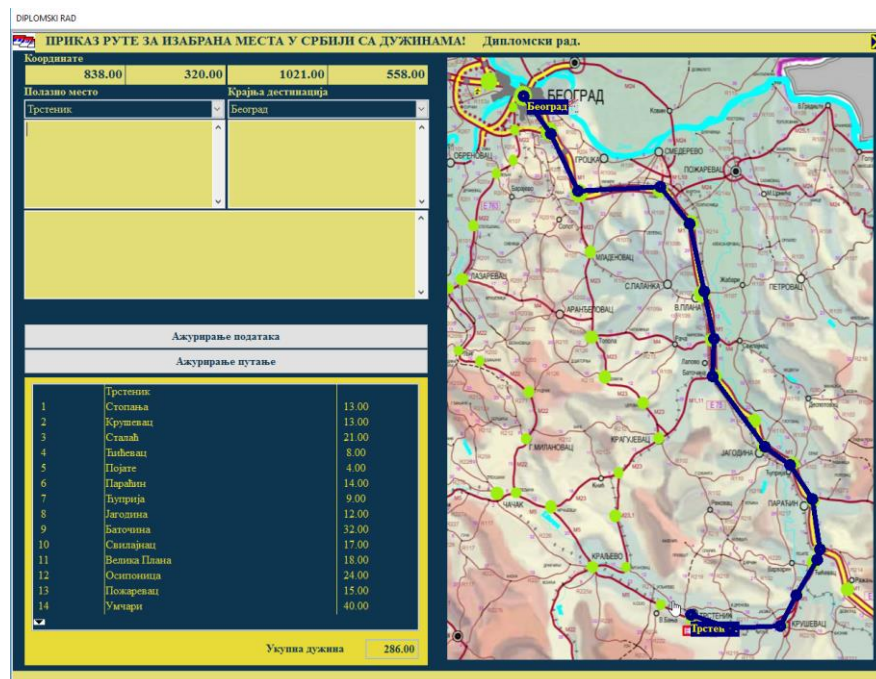
Слика 23. Избор места за дефинисање путање



Слика 24. Избор места на путањи – дефинисање путање



Слика 25. Избор почетног и крајњег места



Слика 26. Приказ путање

У раду је приказан комплетан поступак израде апликације у Visual FoxPro-у. Због специфичности апликације која је захтевала рад са графичким објектима коришћене су библиотеке класа GDI+ FFC, које су карактеристичне за с++, односно .NET окружење.

У другом поглављу су дате основне поставке објектно орјентисаног програмирања, док је у поглављу три представљено радно окружење Visual FoxPro 9. У том поглављу су дати и принципи рада креирања табела база података, рад са упитима, и начин коришћења визуелних алата при креирању форми.

Основу сваког објектног орјентисаног софтвера чине објекти и класе, зато се у поглављу четири посебна пажња поклонила класама у Visual FoxPro-у. У том поглављу су приказане основне класе и објекти који се користе у Visual FoxPro-у, као и начин референцирања-адресирања објеката.

Реализација софтвера је дата у поглављу пет. Кренуло се од структуре апликације слика14. и формирања базе података табела 5 и слика 15. Пошто је захтев апликације био могућност зумирања одређеног региона на слици која представља карту Р. Србије, дефинисана је класа *DefSlika*, на бази основне класе *image*, од које је направљен објекат *karta*. У поменути класу укључене су функције класе GDI+ FFC.

Такође у овом поглављу је илустративно представљен начин рада кроз слике од слике 19. до слике 26.

За све објекте који се појављују у апликацији је такође написан програмски код, а део од тих кодова је представљен у блоковима ПК3 до ПК6.

Visual FoxPro има све карактеристике једног доброг орјентисаног софтвера, добру визуалну подршку и разумљиву синтаксу за писање програмског кода. Поред ове предности друга предност је могућност коришћења графичких (и других) библиотека класа С++ .NET.

- [1] M. Akins, A. Kramek, R. Schummer, 1001 Things You Wanted To Know About Visual FoxPro, Hentzenwerke Publishing, 2000
- [2] Public Domain, Visual FoxPro Programming Standards and Guidelines, Version 4.0, May 16, 2004
- [3] <https://msdn.microsoft.com/en-us/library/ms947605.aspx>, 20.06.2017.
- [4] [https://msdn.microsoft.com/en-us/library/windows/desktop/ms533798\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms533798(v=vs.85).aspx), 21.06.2017.
- [5] [https://msdn.microsoft.com/en-us/library/aa984108\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/aa984108(v=vs.71).aspx), 21.06.2017.
- [6] http://www.ita-software.com/papers/borup_ffc.pdf, 23.06.2017.