

# Seminarski Rad

**Tema:** Razvoj Aplikacije u Pharo-u

**Student:** Stefan Stamenković 297/21

**Fakultet:** PMF Novi Sad

**Predmet:** Seminarski rad B - Pharo

Uvod.....	3
Struktura Programa.....	4
Klase i Njihove Funkcionalnosti.....	5
1. MainMenu Klasa.....	5
2. ExerciseMenu Klasa.....	6
3. SessionMenu Klasa.....	9
Zaključak.....	13
Izvori.....	14

# Uvod

Ovaj seminarski rad se bavi razvojem aplikacije za praćenje vežbi i sesija u programskom jeziku Pharo. Pharo je objektno-orijentisani programski jezik i okruženje koje podržava dinamičko programiranje i bogatu grafičku podršku za korisničke interfejsse. Cilj ovog rada je da se razvije aplikacija koja omogućava korisnicima da evidentiraju, pregledaju i upravljaju svojim fizičkim vežbama i sesijama.

# Struktura Programa

Program je razvijen u Pharo okruženju i sastoji se od tri glavne klase:

- **MainMenu:** Glavni meni aplikacije koji sadrži opcije za izlaz iz aplikacije i ulazak u meni za vežbe.
- **ExerciseMenu:** Meni za upravljanje vežbama gde korisnik može da vidi listu prethodno unesenih vežbi, doda nove vežbe ili obriše postojeće.
- **SessionMenu:** Meni za upravljanje sesijama svake vežbe gde korisnik može da vidi, doda ili obriše sesije za izabranu vežbu.

Svaka klasa ima svoju ulogu u aplikaciji i omogućava korisnicima različite funkcionalnosti kroz grafički korisnički interfejs.

# Klase i Njihove Funkcionalnosti

## 1. MainMenu Klasa

**MainMenu** klasa je odgovorna za prikaz glavnog menija aplikacije. Ovaj meni sadrži dva glavna dugmeta:

- **Exit Button**: Dugme za izlazak iz aplikacije.
- **Exercises Button**: Dugme koje otvara **ExerciseMenu** i omogućava korisnicima da upravljaju svojim vežbama.

Najbitnija metoda ove klase jeste initialize metoda koja kreira objekte dugmića i otvara ih.

- Prikaz initialize klase

```
initialize
    super initialize.

    window := SystemWindow new.
    window setLabel: 'Main Menu'.

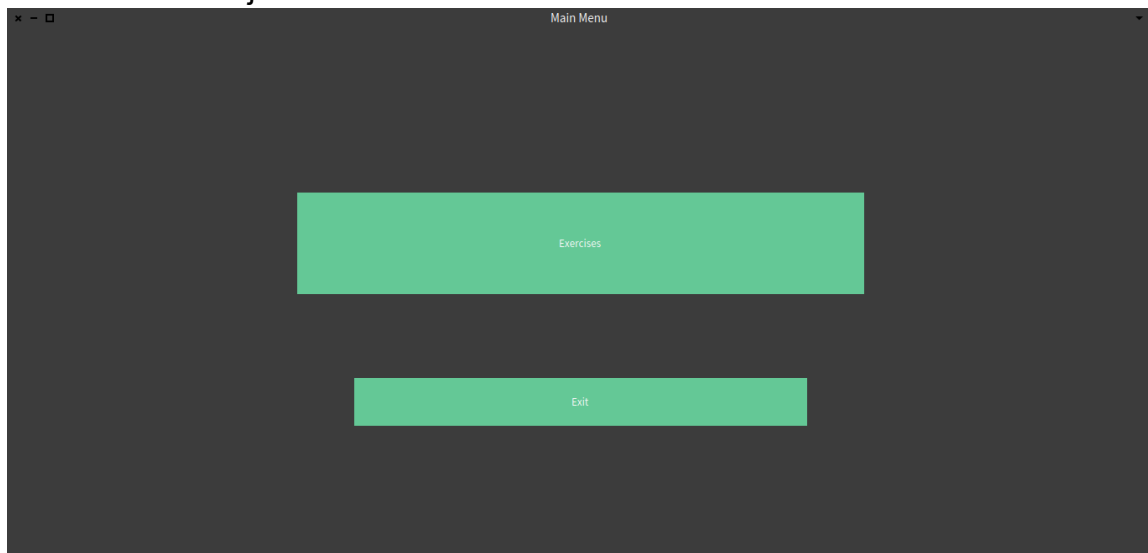
    exercisesButton := SimpleButtonMorph new
        label: 'Exercises';
        target: self;
        actionSelector: #openExerciseMenu;
        openInWorld.

    exitButton := SimpleButtonMorph new
        label: 'Exit';
        target: self;
        actionSelector: #exitApplication;
        openInWorld.

    window addMorph: exercisesButton frame: (0.3@0.45 corner: 0.7@0.55).
    window addMorph: exitButton frame: (0.3@0.35 corner: 0.7@0.45).

    window openInWorld.
```

- Prikaz main menija



## 2. ExerciseMenu Klasa

**ExerciseMenu** klasa pruža funkcionalnost za upravljanje vežbama. Sadrži listu sa svim prethodno unesenim vežbama, dugme za dodavanje nove vežbe i dugme za povratak u glavni meni.

Glavne metode u ovoj klasi uključuju:

- **initialize**: Metoda za inicijalizaciju interfejsa menija za vežbe. Razlikuje se od initialize metode mainMenu klase po PluggableMorph listi kojom prikazujemo vežbe

```
exerciseList := PluggableListMorph
on: self
list: #loadExercisesFromFile
selected: #selectedItem
changeSelected: #updateSelection:.
```

- **addNewExercise:** Metoda koja omogućava korisniku da doda novu vežbu.

```
addNewExercise
| newExerciseName |

newExerciseName := UIManager default request: 'Enter new exercise name:'.

"return if cancel pressed"
newExerciseName ifNil: [ ^ [ ]. ].

"if the field is empty dont add"
newExerciseName ifNotEmpty: [
    "add new exercise"
    normalList add: newExerciseName.
    "write list to file"
    self writeListToFile: normalList.
    "update the morph with new element"
    exerciseList update: #loadExercisesFromFile.
].
```

- **openSelected:** Metoda koja se poziva kada korisnik izabere određenu vežbu iz liste i otvara **SessionMenu** za tu vežbu.

```
openSelected
| sessionMenu exeName |

"returns if nothing is selected"
selectedItem ifNil: [
    UIManager default inform: 'Please select an item first.'.
    ^ [ ].
].

"get name of selectedItem"
exeName := (normalList at: selectedItem).

"open session menu with the selected exercise"
sessionMenu := SessionMenu newSessionWithName: exeName.
sessionMenu openInWorld.

window delete.
self delete.
```

- **loadExercisesFromFile**: Metoda koja učitava listu vežbi iz file-a.

```
loadExercisesFromFile
| fileReference contents |

"file path"
fileReference := 'exercises.txt' asFileReference.

"creates an empty collection"
normalList := OrderedCollection new.

"if a file doesn't exist create a new empty one"
fileReference exists ifFalse: [ fileReference writeStreamDo: [ :stream | ] ].

"read the contents"
fileReference readStreamDo: [ :stream | contents := stream contents. ].

"split into lines"
contents linesDo: [ :line | normalList add: line ].

"return list of lines"
^ normalList
```

- **writeListToFile**: Metoda koja upisuje prosleđenu listu u file.

```
writeListToFile: aList
| fileReference |

"file path"
fileReference := 'exercises.txt' asFileReference.

"clear file"
fileReference binaryWriteStreamDo: [ :stream |
    stream truncate.
].

"write new list into file"
fileReference writeStreamDo: [ :stream |
    aList do: [ :each |
        stream nextPutAll: each; cr.
    ].
].
```



- **removeExercise**: Metoda koja briše izabrani element iz liste i file-a.

```
removeExercise

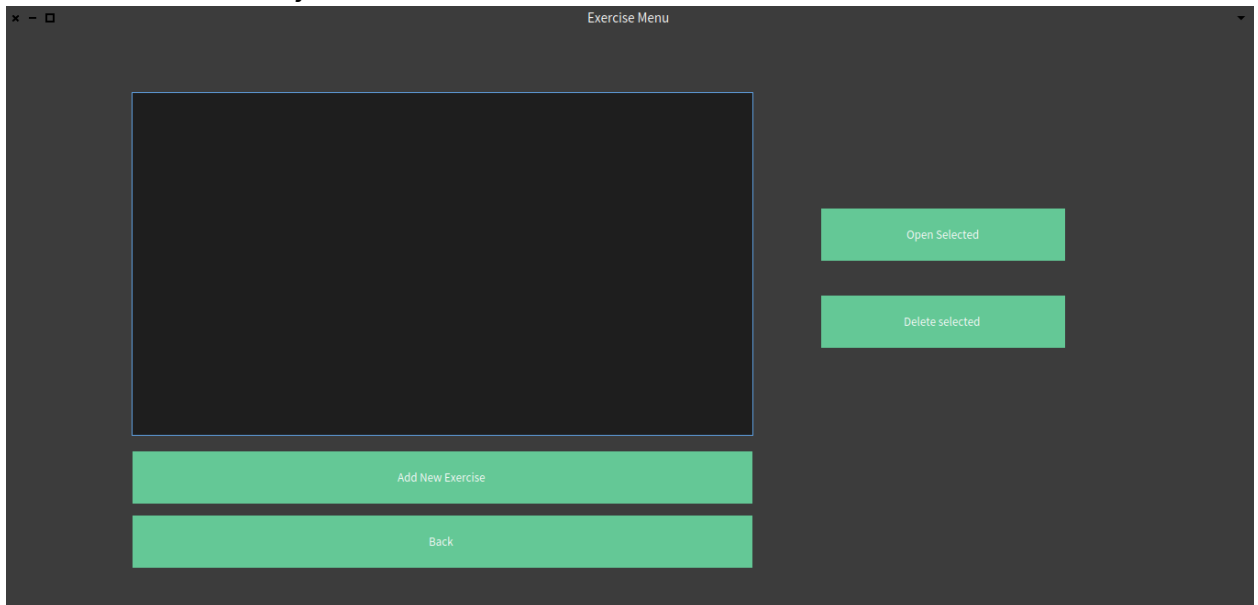
"check if sI is in bounds or nil"
(selectedItem isNil
 or: [ (selectedItem <= 0 or: [ selectedItem > normalList size ]) ])
ifTrue: [ ^ [ ] ].

"remove item from list"
normalList removeAt: selectedItem.

"write new list to file"
self writeListToFile: normalList.

"update visual"
exerciseList update: #loadExercisesFromFile
```

- Prikaz exercise menija



### 3. SessionMenu Klasa

**SessionMenu** klasa omogućava korisnicima da upravljaju sesijama za svaku vežbu. Kao i **ExerciseMenu**, ona omogućava korisnicima da dodaju nove sesije, brišu postojeće ili se vrate na prethodni meni.

Metode u ovoj klasi uključuju:

- **newSessionWithName:** Metoda je class metoda **SessionMenu** klase koja služi kao konstruktor koji pozivamo pri otvaranju izabrane vežbe u prethodnom meniju.

```
newSessionWithName: anExerciseName
| instance |
    "creates an instance where exerciseName is initialized before loadSessionData is called which needs exerciseName"

    instance := self new.
    instance setExerciseName: anExerciseName.

    instance setupWindow.
    instance setupUIComponents.

    ^ instance.
```

- **addNewSession:** Metoda koja omogućava dodavanje nove sesije.

```
addNewSession
| weight repetitions sets session sessionNumber |

"keep track of session number"
sessionNumber := normalList size + 1.

"user input"
weight := UIManager default request: 'Enter the weight used in the session'.
weight ifNil: [ ^ [ ] ].
weight ifEmpty: [ UIManager default inform: 'Field left empty.'. ^ [ ] ].

repetitions := UIManager default request: 'Enter the number of repetitions performed'.
repetitions ifNil: [ ^ [ ] ].
repetitions ifEmpty: [ UIManager default inform: 'Field left empty.'. ^ [ ] ].

sets := UIManager default request: 'Enter the number of sets done'.
sets ifNil: [ ^ [ ] ].
sets ifEmpty: [ UIManager default inform: 'Field left empty.'. ^ [ ] ].

"create a string to be added to collection "
session := 'Session #', sessionNumber printString, ' Weight used: ', weight, ' Sets: ', sets, ' Repetitions: ', repetitions.

"add the new session "
normalList add: session.

"write new list to file "
self writeListToFile: normalList.

"update visual"
sessionList update: #loadSessionData.
```

- **removeSession:** Metoda koja omogućava uklanjanje izabrane sesije.

```
removeSession

"check if SI is within bounds or nil"
(selectedItem isNil
 or: [ (selectedItem <= 0 or: [ selectedItem > normalList size ]) ])
ifTrue: [ ^ [ ] ].

"remove the element from list"
normalList removeAt: selectedItem.

"write the new list without item"
self writeListToFile: normalList.

"update visual"
sessionList update: #loadSessionData.
```

- **writeListToFile:** Metoda kojom upisujemo prosleđenu listu u file.

```
writeListToFile: aList
| fileReference filePath |

"get file path"
filePath := self getFilePath.
fileReference := filePath asFileReference.

"clear file before writing"
fileReference binaryWriteStreamDo: [ :stream |
    stream truncate.
].

"write list to file"
fileReference writeStreamDo: [ :stream |
    aList do: [ :each |
        stream nextPutAll: each; cr.
    ].
].
```

- **ensureDirectoryExists:** Metoda koja proverava da li postoji folder u kojem se kreiraju text file-ovi.

```
ensureDirectoryExists: directoryPath
| directoryReference |

"check if directory exists if not create a new one"
directoryReference := directoryPath asFileReference.
directoryReference isDirectory
    ifTrue: [ ^ self ].

directoryReference createDirectory
```

- **loadSessionData**: Metod sličan metodi u **exerciseMenu** klasi, drugačiji filepath u zavisnosti od izabrane vežbe.

```
loadSessionData
| directoryPath fileReference filePath contents |

"initialize list "
normalList := OrderedCollection new.

"get path"
directoryPath := self getDirectoryPath.

"create dir if needed"
self ensureDirectoryExists: directoryPath .

"file path"
filePath := self getFilePath.
fileReference := filePath asFileReference.

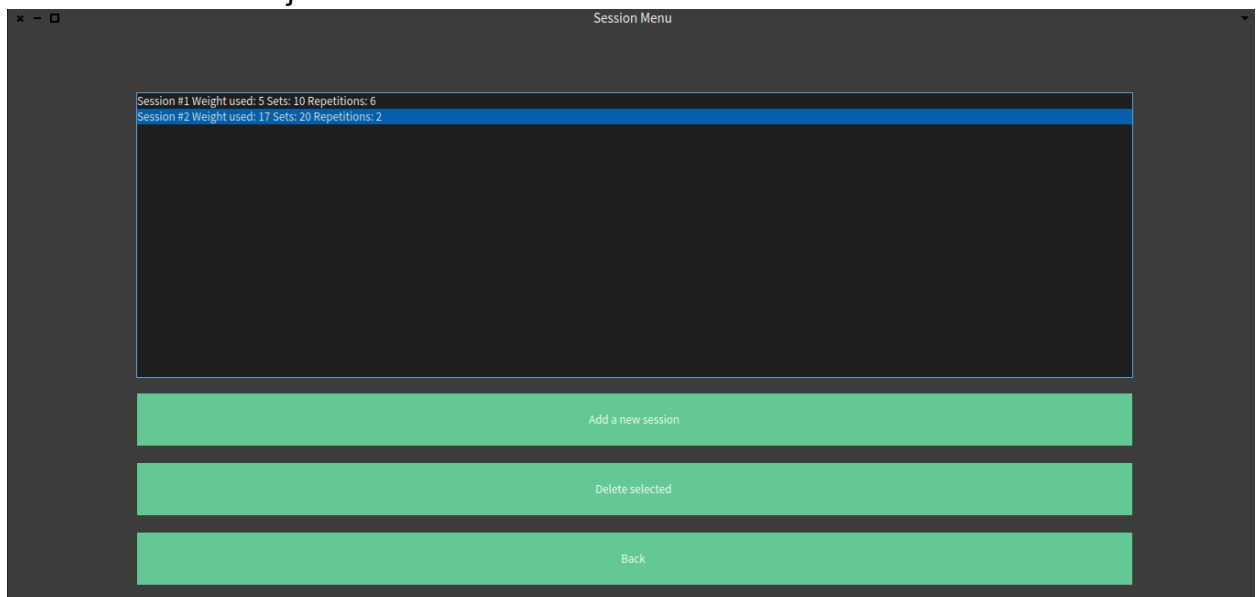
"if a file doesn't exist create a new empty one"
fileReference exists ifFalse: [ fileReference writeStreamDo: [ :stream | ] ].

"read the contents"
fileReference readStreamDo: [ :stream | contents := stream contents. ].

"split into lines"
contents linesDo: [ :line | normalList add: line ].

"return list of lines"
^ normalList
```

- Prikaz session menija



# Zaključak

Razvijanje ove aplikacije u Pharo okruženju pruža uvid u osnovne principe objektno-orijentisanog programiranja i grafičkog korisničkog interfejsa. Korisnici mogu efikasno pratiti svoje vežbe i sesije, dok programer ima priliku da primeni različite konceptualne pristupe u okviru Pharo Smalltalk-a, kao što su rad sa klasama, metodama, listama i interaktivnim elementima korisničkog interfejsa. Dalje unapređenje aplikacije može uključiti dodavanje statistike vežbi, praćenje napretka i povezivanje sa drugim sistemima za praćenje fizičke aktivnosti.

# Izvori

- Stéphane Ducasse, Gordana Rakic, Sebastian Kaplar, Quentin Ducasse, Pharo By Example: A Quick Introduction to Programming in Pharo.
- The Pharo Mooc - <https://mooc.pharo.org/>
- Mooc Immersive Object Programming with Pharo - <https://www.youtube.com/@inrialearninglab>
- Pharo forums - <http://forum.world.st/Pharo-f1294836.html>