

Prirodno-matematički fakultet  
Novi Sad

# Seminarski rad

## **Pharo:**

**Korišćenje Iceberg-a u kombinaciji sa Git i  
GitHub kao osnova za kreiranje projekata u  
Pharo programskom okruženju**

**i**

**Osnove kreiranje paketa, klasa i metoda u  
Pharo-u**

Profesor: Gordana Rakić  
Autor: Božidar Đurić

# Apstrakt

**Pharo** je open source, čisto objektno-orijentisano programsko okruženje koje je osnovano u **SmallTalk**-u korisnicima pruža lakoću u upravljanju sa objektima, i njihovim klasama i metodama, kao i intuitivno kreiranje istih.

On nudi dinamično okruženje u kojem se vaši objekti igraju na igralištu, u kojem se javlja greška ukoliko ne napišete opis klase koju kreirate, u kojem vam debugger izjavljuje greške i načine optimizacije postojećeg koda, u kojem imate potpun pristup svim komponentama okruženja koji se nalazi na virtuelnoj mašini za vreme pokretanja, kreiranja, debug-ovanja i testiranja vaših projekata, i mnogo više.

**GitHub** je alat koji nudi mogućnost kontrole verzija kao i timski rad na svim mogućim projektima, on je kompletno ostvaren i funkcionalan na web-u. GitHub može da sadrži izvorni kod bilo kog projekta, i može da prati kompletnu istoriju izmena izvršenu na tom projektu. Nudi mnoge alate koje pomažu u tome da svi projekti budu realizovani brže i efikasnije. GitHub je alat koji mora biti dostupan svakom modernom programeru.

Videćemo kako tačno možemo da personalizujemo **Pharo** da odgovara svakom korisniku, i kako se on tačno povezuje sa **GitHub**-om, kako se prave **.ssh** ključevi i kako se koristi (po default-u) ugradjen sistem **Iceberg** za povezanost sa **GitHub**-om. Videćemo i osnove koje su potrebne za kreiranje prvog projekta, i proći ćemo kroz lagan uvod u kreiranju tog projekta i njegovog rada vezanog za **Git** i **GitHub**.

Ja sam takođe početnik u **Pharo**-u, i dok, iskreno, može da izgleda zastrašujuće i bezpotrebno kompleksno na prvi pogled, uveravam vas da je okruženje koje je više nego zaslužilo moju pažnju kao jedan od čisto objektno orijentisanih programskih jezika.

## **Sadržaj**

<b>GitHub - Uvod, kreiranje prvog repozitorijuma, njegova svrha i korišćenje</b>	<b>4</b>
<b>Pharo - Uvod u instalaciju Pharo-a, Pharo launcher, kreiranje slike (image)</b>	<b>5</b>
<b>Pharo - Upoznavanje sa Pharo-om</b>	<b>6</b>
<b>GitHub - Kreiranje GitHub naloga, konfiguracija .ssh ključeva i upotreba</b>	<b>8</b>
<b>Pharo - Kreiranje prvog paketa, klase i metode, prvi commit i prvi push</b>	<b>11</b>
<b>Reference</b>	<b>13</b>

Pre nego što počnemo da se bavimo Pharo-om, pre nego što napravimo prvi paket sa našim prvim klasama, hajde prvo da se podsetimo kako se pravi **GitHub** repozitorijum, šta je on i kako se koristi.

GitHub **repozitorijum** (skraćeno: *repo*) je zapravo samo folder koji se nalazi na GitHub-u u kojem stoje svi naši fajlovi potrebni oko projekta za kojeg je taj repozitorijum vezan. Repo automatski beleži sve izmene fajlova koje mogu da iniciraju i obavljaju više od jedne osobe nad jednom projektu.

Da bismo otvorili svoj prvi GitHub repo, potrebno je da napravimo svoj GitHub nalog. Prosto idite na ***github.com***, i napravite svoj nalog.

Nakon što ste napravili svoj GitHub nalog, videćete zeleno dugme: **“Create a repository”**, kliknite na njega, unesite željeno ime repozitorijuma, unesite kratak opis, izaberite da li želite privatan ili javan repozitorijum\*, da li želite README fajl\*\*, i kliknite opet dugme “Create a repository” da biste ga kreirali.

\*Razlika između javnog i privatnog repozitorijuma je u tome ko ima pristup njemu: Ukoliko je repozitorijum javan (Public), svako može, dok ako je privatan (Private), mogu samo oni koje vi kao vlasnik repozitorijuma odobrite.

\*\*README fajl je samo tekstualni fajl u kojem možete da napišete duži opis vašeg projekta.

Čestitam! Sada posedujete vaš sopstveni Repozitorijum! Na njemu možete da uploadujete fajlove, možete da ih izmenjujete i snimate na vašu lokalnu mašinu, kao i delite sa ljudima širom sveta, koji mogu vam pomoći u realizovanju baš vašeg projekta kroz vaš sopstveni repozitorijum.

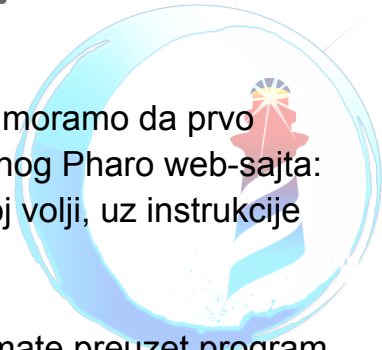
Verovatno se pitate: ***I šta sad?***

I ne brinite, vратиćemo se na GitHub malo kasnije, hajde sada da se pozabavimo upoznavanju sa Pharo-om.

# Pharo

## *Uvod u instalaciju Pharo-a, Pharo launcher, kreiranje image-a, objašnjenje šta su te stvari u Pharo-u*

Da bismo počeli praviti naše prve projekte u **Pharo-u**, moramo da prvo preuzmemo Pharo programersko okruženje sa zvaničnog Pharo web-sajta: **pharo.org** i možete preuzeti i instalirati Pharo po vašoj volji, uz instrukcije dostupne na sajtu.



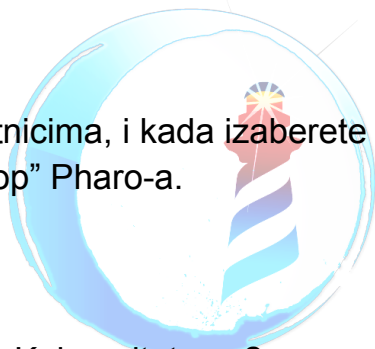
Kada završite preuzimanje i instalaciju, trebalo bi da imate preuzet program **Pharo launcher**. Pokrenite ga, i nakon kratkog first-time učitavanja trebalo bi da se otvori.

**Pharo launcher** je program koji služi za upravljanje sa Pharo image-ovima. Image (*Slika*) fajl je u Pharo-u zapravo celina koja predstavlja jedan projekat. U Pharo launcher-u imate sortirana sva imena vaših image fajlova, to jest stajala bi, da nismo na svežoj instalaciji Pharo-a, i sa ovog ekrana ćete, kasnije i videti, moći pročitati kratke opise o verziji Pharo-a za svaku *sliku*, kao i kratak opis tog projekta, lokaciju slike na lokalnoj memoriji i slično. Iz Pharo launcher-a možete da pokrenete, obrišete, importujete i kreirate nove Pharo slike.

*Hajde da kreiramo novu sliku!*

- Prvo u gornjem levom uglu aplikacije locirajte i kliknite na dugme na kojem piše “New”, i nacrtana je ikonica izlaska sunca koji tematski ilustruje šta ćemo dobiti nakon klika na to dugme - novu Pharo sliku (image).
- O ne! Umesto prelepog novog početka, otvorio se nekakav užasan prozorčić sa mnogo nerazumnih naziva!
- Ne brinite! Saberite se malo, i izaberite distribuciju i iz nje izaberite verziju Pharo-a na kojoj želite da radite. (Ja sam izabrao **Pharo 11.0 64 bit**)
- Nakon toga želite da imenujete vašu sliku u polje “**Image name**”, i odmah ispod toga joj date i kratak opis.
- Za sad ćemo ostaviti “Initialization script” polje prazno i nećemo objašnjavati više o tome, ukoliko ste zainteresovani Pharo ima bogatu dokumentaciju. Koju možete pronaći na njihovom sajtu.
- I kliknite dugme “Create image”.

Kada završite sa čitanjem zida teksta namenjenog početnicima, i kada izaberete tamnu temu (*Buuu!*) Doći ćete na početnu stranu, "Desktop" Pharo-a.



*Dobrodošli u Pharo!*

Unutar ovog prozora ćete raditi celo razvijanje projekata. Kako, pitate se? Polako!

Hajde prvo da se upoznamo sa osnovama koje će nam trebati za početak.

Kliknite "Browse" dugme na vrhu prozora. U njemu ćete videti četiri stavke koje će nas sada zanimati, a to su: **System browser**, **Iceberg**, **Playground**, i **Transcript**.

Prvo ćemo pričati o Playground i Transcript-u, pa ćemo pričati o System browser-u, a za zadnje ćemo ostaviti Iceberg, kojim ćemo se služiti uz Git i GitHub za kontrolu verzija projekata i dr.

### ***Playground i Transcript***

Otvorite Playground i Transcript i pomerite prozore tako da ih oboje vidite jasno.

**Playground** je Pharo-u ono što je konsola nekim drugim programskim jezicima, u njemu možemo da pokrećemo male delove koda, kreiramo i baratamo sa objektima i koristimo se njime da testiramo naše klase, kad budemo radili na njima. Ja ovde neću prelaziti preko sintakse koda Smalltalk-a (Pharo-a), ali hoću objašnjavati neke delove koda koje ćemo koristiti.

**Transcript** se inače koristi kao output (izlaz) iz konzole, u njemu se javljaju objašnjenja grešaka i slično, ali u njega možemo i mi da koristimo.

# Pharo

*Dobrodošli u Pharo, upoznavanje sa korišćenjem Pharo-a*

Unutar prozora **Playground**-a, unesite kod:  
Transcript show: 'Pozdrav iz konzole'.

Ovaj kod šalje show: **poruku** objektu **Transcript** sa parametrom string koji sadrži naš pozdrav.

Da biste pokrenuli kod, desnim klikom na red u kojem ste napisali tu poruku i izaberite polje **"Do it"**, radi takodje i **CTRL + D**.



Vidimo da se u prozoru **Transcript**-a pojavio naš pozdrav!

*To je za sada to što se tiče ove dvojice, idemo na System browser!*

*Ukoliko vas više zanima o korišćenju jezika, imate kratak interaktivni tutorijal "ProfStef", kao i besplatan kurs na Mooc platformi.*

*Možete zatvoriti ova dva prozora.*

## System browser

Otvorite **System browser**.

Ukoliko ste kao ja, na prvi pogled ovo vam izgleda strašno, i sa dobrim razlogom! Verovatno ste već čuli da je u ljudskoj prirodi da se plašimo nepoznatog, zato hajde da požurimo i objasnimo šta to tačno mi ovde vidimo!

Vidimo da je prozor nazvan **Current image** podeljen na dve celine, gornju, koja je podeljena na još 4 dela, od kojih su 3 prazna, i donji koji ima naziv "New class".

Gornji deo prozora, na desnoj strani, to su vam **paketi**, trenutno imate samo systemske pakete koji dolaze uz Pharo, i unutar svakog paketa nalaze se klase koje su povezane nekom logikom. Razmišljajte o paketima kao o kutijama alata, mnogo ih ima, i svaki od njih sadrži nešto što će nam možda nekad trebati, ali velika većina njih su trenutno potrebni sistemu, zato ih ne dirajte, za sada. Donji deo prozora koristimo za unošenje koda, kreiranje klasa i metoda i svo ono *konkretno* programiranje. Hajde da pričamo malo o Iceberg-u, a vratićemo se na pakete i klase kasnije. *Možete izaći iz prozora.*

## ***Iceberg***

Evo konačno stigli smo i do njega! Ovo je heroj ove priče, a sad ćete saznati i zašto.

**Iceberg** je glavni alat koji se koristi za kontrolu verzija u Pharo-u. Daje nam mogućnost da baratamo sa Git repoizitorijima direktno iz unutrašnosti Pharo-a.

Otvorite **Iceberg**

Polako! Nemojte da vas plaši gomila “*Local repository missing*” poruka što viču na vas!

Objasnićemo sve polako.

Šta vidite na ekranu trenutno su različiti repozitorijumi koji se trenutno **ne** nalaze na vašoj mašini, i to vas samo obaveštavaju. Kad podesimo sve, izgledaće sve mnogo bolje...

...dobro, iskreno, za sada ćemo da “popravimo” samo jedan od njih, koji ćemo i koristiti - Pharo.

Sad kada smo završili sa upoznavanjem sa Pharo-om, hajde da se vratimo na glavnu temu: ***Kako da koristimo Iceberg u vezi sa našim GitHub repozitorijumom?***

***Hajde da vidimo i to!***



## Hajde da povežemo GitHub sa našom mašinom!

- Prvo što želite da uradite je da preuzmete i instalirate **Git**

**Git** je osnova koja čini GitHub, on je open source program koji se koristi za distribucije projekata efikasno i brzo.

Možete ga preuzeti sa njihovog web sajta ***git-scm.com***.

Proces instalacije je jednostavan, klasičan “next, next, next, finish”, tako da neću ulaziti u detalje, tokom instalacije ne treba vam nikakva dodatna podešavanja.

- Nakon što ste instalirali **Git**, vreme je da generišete vaše .ssh ključeve. Sad, različito je na svakom sistemu, na Windows-u preko Power Shell-a pokrenite komandu **ssh-keygen -t rsa -b 4096** da bismo dobili odgovarajući javni i privatni ključ odgovarajuće bezbednosti na našoj mašini u folderu po našem izboru.
- Kada ste napravili ključeve, nađite ih u vašem direktorijumu i otvorite vaš GitHub profil, i u podešavanjima profila nadjite sekciju “**SSH and GPG keys**”, unutar nje kreirajte novi SSH key, i kopirajte sadržaj vašeg ključa iz direktorijuma .ssh koji ima ekstenziju .PUB (Možete koristiti i notepad) unutar prozora, imenujte ključ i sačuvajte izmene.
- Hajde da naučimo da Fork-ujemo tuđi repositorijum. Na GitHub-u pronadjite **repositorijum** na adresi “***github.com/pharo-project/pharo***”, i kliknite dugme “**Fork**”. Imenujte ga, i on će biti dodat kao vaš repositorijum. Svaki javni repositorijum možete Fork-ovati i kopirati kao vaš, na kojem možete da vršite izmene, kad ga kasnije učitate Iceberg-om u Pharo-u.

- Otvorite opet prozor **Pharo**-a, gornji deo ekrana izaberite Pharo, Settings, i u njima pronadjite opciju **“Use custom ssh keys”**. U ponudjenoj opciji možete da unesete adresu vaših ključeva u lokalnoj memoriji, i sačuvajte izmene.
- Izadite iz Settings-a, uđite opet u **Iceberg**, desnim klikom kliknite na **“pharo”** repozitorijum, i kliknite **“Repair repository”**
- Izaberite **“Clone again this repository...”**
- Izaberite **“Clone from github.com** (automatski odabrano)
- Na desnoj strani unesite ime **vašeg GitHub** naloga pod **“Owner name”**  
Pod **“Project name”** unesite naziv koji ste dali **vašem** **Pharo repozitorijumu** koji ste **Fork**-ovali
- Izaberite **SSH Protokol**, a Local Directory nemorate da menjate.
- Pritisnite **“Ok”**

Uh! To je bilo dosta posla, ali sada konačno imamo povezan **GitHub** nalog sa našim **Pharo** projektom!

Hajde da napravimo naš prvi **Commit**!

## Pharo

*Kreiranje klase, kreiranje metode, prvi commit, prvi push.*

Hajde sad da kreiramo prvi paket sa našom prvom klasom!

Uđite opet u System Browser, i desnim klikom na bilo koji paket na desnoj strani kreirajte novi paket i dajte mu neko ime, recimo “**PaketPozdrav**” (Uz ovaj rad je dostupan moj prvi paket “**GamesAndToys**” na mom GitHub-u koji možete forkovati. **Shoof-q/pharo**).

U našem novom **paketu**, u donjem prozoru možemo kreirati novu **klasu**, prosto promenite ime “**#MyClass**” u ime klase koje mi želimo, za sad neka to bude samo probna klasa, recimo “**ClassPozdrav**” i sačuvajte je.

Vidimo da se u drugom prozoru sa leve strane kreirala klasa, i imamo u donjem delu prozora template za novu metodu, kao i uzvičnik koji nam govori da još nismo ostavili komentar klase.

U komentar možemo prosto napisati “Ovo je klasa koja pozdravlja korisnika”, a i sample **metodu** možemo uneti kod:

**Pozdrav: ime**

```
^ 'Pozdrav' , ime
```

Ova jednostavna metoda, kad je pozvana prosto uzima kao parametar **ime**, i vraća pozdrav tom imenu u obliku stringa.

Sačuvajte sve i uđite u Playground, kreirajte novi objekat ClassPozdrav sa naredbom: **poruka = ClassPozdrav new.**

Sada nam se u promenljivoj “**poruka**” nalazi objekat klase **ClassPozdrav**, i tom objektu možemo da pošaljemo poruku naše metode, i pišemo: **poruka Pozdrav “svete”**.

Kad pokrenemo ovu naredbu sa “**Do it**”, ništa se ne dešava, jer nismo rekli u metodi da igde štampa rezultat, i zato da bi smo videli rezultat naše metode, možemo prosto da umesto “Do it”, pokrenemo naredbu sa “**Print it**”, i ta naredba će nam odštampati rezultat koji će biti: “**Pozdrav svete**”

## Pharo

*Kreiranje klase, kreiranje metode, prvi commit, prvi push.*

Sada kad imamo našu klasu, i naš metod, red je da naučimo da commit-ujemo taj paket.

U **IceBerg**-u dvaput kliknite na repozitorijum "**pharo**"m kliknite na plus u gornjem desnom uglu, i u otvorenom prozoru pronađite vaš paket koji smo nazvali "**PaketPozdrav**", čekirajte ga, i dodajte ga u repozitorijum sa "**Ok**".

Sada se možemo vratiti nazad u System Browser, pronaći naš paket, i desnim klikom izabrati "**Commit**", unesite komentar u donjem delu novo otvorenog prozora "**Moj prvi commit**", proverite da li je odgovarajući paket izabran u gornjem prozoru, i kliknite "**Commit**" dugme u donjem desnom uglu prozora.

Nakon što se commit-uje paket, možete ući u **Iceberg** opet, izabrati **pharo** repozitorijum u kom smo commit-ovali paket, i desnim klikom izabrati "**push**".

Ukoliko je sve išlo po planu, upravo ste izvršili prvi **push** u Pharo-u, koristeći **Iceberg**, i izmene, i vaš novu paket treba da vidite u vašem **GitHub** repozitorijumu, uz poruku da ste vi je učinili.

***Srećno programiranje u Pharo-u!***

## Reference

1. Pharo official website: <https://pharo.org/>
2. GitHub official website: <https://github.com/>
3. Iceberg dokumentacija dostupna na GitHub-u: <https://github.com/pharo-vcs/iceberg/wiki>
4. Pharo logo: <https://files.pharo.org/media/logo/>
5. GitHub logo: <https://github.com/logos>