

Prirodno-matematički fakultet
Novi Sad

Seminarski rad

Pharo:
Upotreba 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
Student: Božidar Đurić

Apstrakt

Pharo je open source, čisto objektno-orijentisano programsko okruženje koje je osnovano u **SmallTalk**-u. Korisnicima pruža lakoću u upravljanju objektima i njihovim klasama i metodama, kao i u intuitivnom kreiranju istih.

On nudi dinamično okruženje u kojem se vaši objekti igraju na igralištu, gde se javlja greška ukoliko ne napišete opis klase koju kreirate. Takođe vam debugger izjavljuje greške i načine optimizacije postojećeg koda. Imate i potpun pristup svim komponentama okruženja koje 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 potpuno 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 je neophodan svakom modernom programeru.

Proučićemo 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) ugrađen sistem **Iceberg** za povezanost sa GitHub-om. Bavićemo se i osnovama 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 bespotrebno komplikovano na prvi pogled, uveravam vas da je okruženje koje je više nego zaslužilo moju pažnju kao jedan od čisto objektno orijentisanih programskih jezika.

Sadržaj

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

Pre nego što počnemo da se bavimo Pharo-om i 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 folder koji se nalazi na GitHub-u u kojem stoje svi naši fajlovi potrebni za projekat za kog je taj repozitorijum vezan. Repo automatski beleži sve izmene fajlova koje može da inicira i obavlja više osoba nad jednim projektom.

Da bismo otvorili naš prvi GitHub repo, potrebno je da napravimo GitHub nalog. Idite na ***github.com*** i napravite nalog.

Nakon što napravite 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” kako biste ga kreirali.

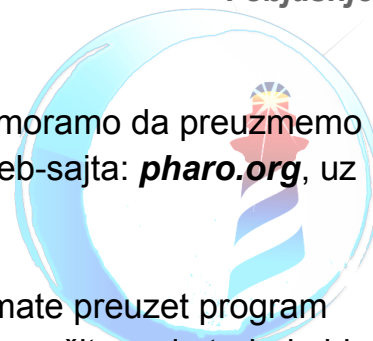
*Razlika između javnog i privatnog repozitorijuma je u tome ko ima pristup njemu: Ukoliko je repozitorijum javan (Public), svako može da pristupi, dok ako je privatan (Private), mogu samo oni koje vi kao vlasnik repozitorijuma odobrite.

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

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

I šta sad?

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



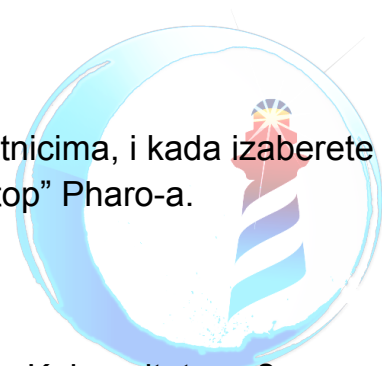
Da bismo počeli praviti prve projekte u **Pharo-u**, prvo moramo da preuzmemo Pharo programersko okruženje sa zvaničnog Pharo web-sajta: **pharo.org**, uz uputstva dostupne na sajtu.

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 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 imali biste da nismo na svežoj instalaciji Pharo-a. Sa ovog ekrana ćete 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 “New” (obeleženo je ikonicom izlaska sunca koja 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 konfuznih naziva!
- Ne brinite! Saberite se malo, i izaberite distribuciju. 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.
- 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 se obavlja 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 nas sada zanimaju, 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 kraj 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 oba vidite jasno.

Playground je Pharo-u ono što je konzola nekim drugim programskim jezicima, u njemu možemo da pokrećemo male delove koda, kreiramo i baratamo 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 objasniću neke delove koda koje ćemo koristiti.

Transcript se inače koristi kao output (izlaz) iz konzole, u njemu se javljaju objašnjenja grešaka i slično, ali 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 koristite se desnim klikom na red u kojem ste napisali tu poruku i izaberite polje **"Do it"** (ili **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, sasvim razumljivo! 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 donju, koji ima naziv "New class".

U gornjem delu prozora, na desnoj strani, nalaze se **paketi**. Trenutno imate samo sistemske 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 zatrebati.

Većina njih su trenutno potrebni sistemu, dakle za sada ih ne dirajte.

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 smo stigli i do njega! **Iceberg** 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 Git repozitorijumima direktno iz unutrašnjosti Pharo-a.

Otvorite **Iceberg**

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

Objasnićemo sve polako.

Na ekranu vidite različite repozitorijume 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 trebaju 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 biste dobili odgovarajući javni i privatni ključ odgovarajuće bezbednosti na vašoj mašini u folderu po izboru.
- Kada ste napravili ključeve, nađite ih u vašem direktorijumu i otvorite vaš GitHub profil, i u podešavanjima profila nađite 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) u prozor, imenujte ključ i sačuvajte izmene.
- Hajde da naučimo da Fork-ujemo tuđi repozitorijum. Na GitHub-u pronađite **repozitorijum** na adresi “***github.com/pharo-project/pharo***”, i kliknite dugme “**Fork**”. Imenujte ga, i on će biti dodat kao vaš repozitorijum. Svaki javni repozitorijum 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, u gornjem delu ekrana izaberite Pharo, Settings, i u njima pronađite opciju “**Use custom ssh keys**”. U ponuđenoj opciji možete da unesete adresu vaših ključeva u lokalnoj memoriji, i sačuvajte izmene.
- Izađite 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 Protocol**, a Local Directory ne morate da menjate.
- Pritisnite “**Ok**”

Uh! To je bilo dosta posla, ali konačno smo povezali **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, u donjem delu prozora imamo template za novu metodu, kao i uzvičnik koji nam govori da još nismo napisali komentar klase.

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

Pozdrav: ime

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

Ova jednostavna metoda, kad je pozvana 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. Pišemo: **poruka Pozdrav “svete”**.

Kad pokrenemo ovu naredbu sa “**Do it**”, ništa se ne dešava, jer nismo rekli u metodi da igde štampa rezultat. Zato, da bismo videli rezultat naše metode, možemo 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**". 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 novootvorenog 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**. Izmene i vaš nov paket treba da vidite u vašem **GitHub** repozitorijumu, uz poruku da ste vi izvršili **push**.

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>