

Granskningsmall för annan grupps projektarbete

- **Det skapade datorspråket**

- **Verkar språket uppfylla de krav som språket är tänkt att användas till?**

Nej. Egentligen så fick vi inte testa språkets funktionalitet eftersom nästan alla definierade språkkonstruktioner funkar inte.

- **Egna synpunkter på språket. Dess syntax och semantik.**

Språkets syntax ser bra ut och liknar de flesta andra programmeringsspråk. Målet med språket var att den ska ha blandning mellan C# och Ruby. Men den ser mest ut som Ruby just nu eftersom den har inte blivit som gruppen tänkte från början än.

- **Användarhandledningen. Är det möjligt att förstå hur språket fungerar och kan man enkelt lära sig det genom att läsa manualen? Finns bra exempel?**

Vi fick veta hur ska man köra och testa språket. Vi kunde även förstå hur språket fungerar och vi kunde lära oss en del av språket genom att läsa manualen. Det som vi tänkte säga är att det blir bättre när man skriver fullständigare exempel på hur vissa konstruktion fungerar, dvs att täcka de flesta funktionalitet som finns i språket. Till exempel, på print sektionen så finns ett exempel på hur man kan skriva ut en variabel, men det finns inte om hur man kan skriva ut ett heltal eller flyttal.

- **Språkbeskrivningen med grammatik**

- **Grammatiken. Verkar den vara korrekt, välskriven och ha korrekta benämningar på de olika språkkonstruktionerna?**

Vi tycker att grammatiken skulle se bättre ut om det inte finns några upprepningar det vill säga att det finns möjlighet att inte repetera vissa konstruktioner. Vi hade även lite svårt att skilja på några benämningar också vilket gjorde det lite förvirrande för oss i början.

Ex: matchregel som heter (:run) skulle exempelvis heta statement eller något liknande för att senare använda det på (:codeBlock) istället för att skriva alla de konstruktioner igen på (:codeBlock).

- **Överensstämmer beskrivningen av språket med grammatiken?**

Enligt vår tolkning, så stämmer överens beskrivningen som finns på dokumentationen av språket med grammatiken det vill säga BNF.

- **Finns det konstruktioner i språket som grammatiken ej verkar ta upp, eller beskriver grammatiken möjliga konstruktioner, som språket ej verkar kunna hantera.**

Eftersom gruppen håller på fortfarande att göra projektet klar så för den tidpunkten mesta saker i grammatiken dvs BNF inte hanteras i språket.

- **Ger systembeskrivningen en bra bild på hur implementeringen ser ut?**

Det vore uppskattas om det var skriven lite mer detaljerat på hur implementeringen ser ut men annars är det välskriven.

- **Implementering av verktyg eller användning av verktyg för språket**

- **Implementering av egna eller användning av existerande verktyg (lexer/parser etc). Verkar verktyget lämpat för uppgiften? Är verktyget använt på ett bra sätt? Finns det begränsningar i verktyget, som gör det svårt att implementera datorspråket?**

Gruppen använder sig av den befintliga parser (rdparse.rb) på ett bra sätt som med tanke på gruppens mål verkar vara passande.

- **Metoder och algoritmer.**

- **Synpunkter på valda metoder och algoritmer, verkar de vara bra val? Kan det bli mycket ineffektivt? Finns det alternativ, som hade blivit bättre?**

Vår förhoppning var att vi kunde testa om de metoder och algoritmer fungerar eller inte. Dock tycker vi att det finns några onödiga metoder och algoritmer som har gjort koden lite längre dvs lite upprepningar på vissa klasser (Int, Float och Bool som har exakt samma algoritm).

Vi märkte också att det finns vissa metoder/funktioner som är oanvända för nuläge, exempel: (get_list, get_var_name i variabellist klassen).

- **Koden för implementeringen i Ruby.**

- **Är koden bra modulariserad? Kan man särskilja de olika delarna som programmet består utav och de olika algoritmerna.**

Ja, koden har bra modularisering dvs vi kan ju särskilja på de olika delarna som programmet består utav, exempelvis de har separerat mellan klasserna dvs att de klasserna som handlar om funktioner är tillsammans och de klasserna som handlar om iterationer är det också osv.

- **Finns det en överensstämmelse mellan hur grammatiken är beskriven och motsvarande strukturer och kod i programmet.**

Utifrån vår observation så stämmer grammatiken och strukturer överens med varandra till exempel vi ser att det finns relevanta grammatik för en och varje struktur i programmet.

- **Är koden läsbar och förståelig för dig som läsare? Val av namn på olika storheter (identifierare för variabler, klasser, metoder/funktioner/procedurer mm).**

Koden är läsbar och det var inte svårt att förstå, Bra klass/funktioner/variabler namn.

- **Ta gärna ut en del som du tycker var bra och motivera varför och en del som du tycker var dåligt / som du inte alls enkelt kan förstå och motivera varför.**

Vi kunde inte komma på ett bra exempel men vi tycker att koden är skriven på ett bra sätt i sin helhet. Det är lite jobbigt för oss att hitta vad som är bra och vad som är dåligt på koden eftersom den är inte fungerande för denna tidpunkten när vi svarar på dessa frågor.

- **Code complete-boken eller andra kodstandarder**

- **Verkar programmerarna ha följt en egen kodstandard?**

Enligt gruppens dokumentation skulle deras kodstandard vara en kombination mellan C# och Ruby men det som vi kan se just nu i koden så ser den ut mer eller mindre som en Ruby kodstandard. Dock har vi tyvärr inte något erfarenhet på C# språket.

- **Testkörning av språket**

- **Var det lätt att komma igång med systemet?**

Ja det var faktiskt lätt att komma igång på hur systemet fungerar. Enligt gruppens manual så står det hur det går till så det är bara att följa deras manual.

- **Är det lätt att skriva program i språket? Ger användarhandledningen stöd för detta?**

Vi har inte lyckats tyvärr skriva något program i testfilen och detta på grund av att den var oklar. Annars användarhandledningen ger ett ganska bra stöd för att kunna skriva program.

- **Visade egna testkörningar på några problem? Felaktigheter uppstår vid felaktig kod i datorspråket.**

Tyvärr efter många försök så kunde vi inte köra tester på deras program , vi har försökt testa alla möjliga konstruktioner som if-sats eller matematik och flera andra saker men vi kom tyvärr inte fram på ett resultat.

- **Finns oklarheter i konstruktioner, vad som borde hända?**

Vi tolkade frågan så här att det måste visas något slags felmeddelande(felhantering) när det förekommer en oklarhet i konstruktion. Till exempel, vi hade felhantering i vår funktioner så om användaren kallar på en funktion som inte finns i programmet då visas felmeddelandet som vi skapade i vårt program i terminalen. I gruppens fall så är det samma svar att vi inte fick testa så vi fick inte någon felhantering heller. Dessutom såg vi inte felhantering i deras kod.