

Jämförelse av Java och Python i webbserverapplikationer

Peter Engström

30.3.2015

Åbo Akademi

Fakulteten för naturvetenskaper och teknik

Datateknik

Handledare: Dragos Truscan

Referat

Java och Python är två av de populäraste högnivåprogrammeringsspråken. De är särskilt relevanta inom webbserverapplikationer, det vill säga program som körs på datorer som en webbläsare hämtar webbsidor från. Webbserverapplikationer används ofta för att erbjuda dynamiskt innehåll på webbsidor. Dynamiska webbsidor kan även åstadkommas med hjälp av program som körs i webbläsaren, så kallad klientprogramvara. Klientprogramvara ligger dock utanför den här avhandlingens omfattning. I avhandlingen undersöks vilka skillnader Java och Python har i webbserverprogram i allmänhet, samt specifika typer av dessa. Skillnaderna behandlas bland annat på historisk, syntaktisk och prestandamässig nivå. Slutligen presenteras vilka slutsatser som följer av de tidigare kapitlens jämförelser.

Nyckelord: Java, Python, webb, server, jämförelse

Innehåll

Referat	II
1 Inledning	1
1.1 Webbapplikationer.....	1
1.2 Java och Python	1
2 Jämförelsekriterier	2
3 Historisk bakgrund.....	4
3.1 Java	4
3.2 Python	5
4 Användarbas.....	7
5 Syntax.....	9
5.1 Avgränsning av kodstycken.....	9
5.2 Klasser	10
5.3 Typer.....	10
6 Prestanda	12
6.1 Mätningar.....	12
6.2 Resultat	13
7 Tillgängliga verktyg.....	14
8 Slutsatser	16
Litteratur	18

1 Inledning

1.1 Webbapplikationer

Webbapplikationer är ett sätt att erbjuda dynamiskt innehåll på webbsidor, det vill säga innehåll som kan ändras automatiskt och som användaren kan växelverka med. Webben bygger på att en webbläsare (klient) skickar en förfrågan till en annan dator, och datorn (servern) skickar ett svar. Detta kallas för en klient-server-arkitektur.

Webbapplikationer kan köras antingen i webbläsaren eller på servern. Fördelar med webbserverapplikationer i förhållande till webbklientapplikationer, är bland annat att serverapplikationer kan erbjuda bättre säkerhet, prestanda och mångsidigt innehåll. Serverapplikationer används i synnerhet för informationssökning och -lagring i databaser [1, s. 11-13].

1.2 Java och Python

Java och Python är objektorienterade högnivåprogrammeringsspråk. De är idag bland de populäraste programmeringsspråken [2], särskilt inom servrar med mycket trafik [3]. I avhandlingen undersöks vilka för- och nackdelar vardera språket har i särskilda typer av serverapplikationer, och serverapplikationer i allmänhet.

Java utvecklades 1995 av James Gosling för Sun Microsystems [4]. Sun köptes senare upp av Oracle, som nu äger den implementation av språket som de facto är standarden. Javas syntax liknar C-språkets, men Java är i motsats till C klassbaserat och objektorienterat. Java är mycket portabelt, eftersom samma källkod i princip kan köras på vilken som helst hårdvara som har JVM-programmet (Java Virtual Machine) installerat [5, s. xiii].

Python utvecklades 1991 av Guido van Rossum [6]. Språkets utveckling leds av Python Software Foundation. Pythons syntax bygger på radbyten och indrag. Liksom Java är Python ett mycket portabelt språk [7, s. 2].

2 Jämförelsekriterier

Kapitel 3: Historisk bakgrund

Bakomliggande historiska faktorer bör aldrig bortses från, och programmeringsspråk är inga undantag. En teknologi blir inte alltid populär utifrån sina verkliga meriter, utan saker som rätt val av tidpunkt, bra marknadsföring och tur kan ofta vara avgörande. Dessutom kan klimatet i programmeringsindustrin på den tiden då språket utvecklats ha påverkat tänkandet bakom utvecklingen.

Kapitel 4: Användarbas

Popularitet är inte ett säkert tecken på kvalitet, men en stor användarbas kan i sig ha stor betydelse, i synnerhet i fråga om programmeringsspråk. Bland annat är det lättare att hitta kompetenta programmerare i ett mycket använt språk, och det är mer sannolikt att ett använt språk stöds av programmeringssamhället. Det senare har särskilt stor betydelse i språk som kan utvecklas av ”vem som helst”.

Kapitel 5: Syntax

Syntaxen är en väsentlig del av vilket som helst programmeringsspråk, men det är speciellt viktigt att fästa uppmärksamhet vid den, då det gäller två så olika syntaxer som Pythons och Javas. Trots att det ofta är en subjektiv fråga vilken syntax som är ”bättre”, kan vissa skillnader ha verklig betydelse.

Kapitel 6: Prestanda

Man kan resonera att prestanda inte har samma betydelse i webbapplikationer som i andra program, eftersom flaskhalsen brukar vara i databasen eller överföringshastigheten. I kapitlet om prestanda kommer dock att visas, att det finns stora skillnader även inom detta område.

Kapitel 7: Tillgängliga verktyg

Även om avhandlingen jämför själva språken, är ett programmeringsspråk oskiljbart från de verktyg som finns tillgängliga för det. Detta kan innebära bibliotek, ramverk och annat, både inbyggda och tredjepartsverktyg. I avhandlingen behandlas

särskilt sådana verktyg som är relevanta för serverapplikationer, till exempel webbutvecklingsramverk.

3 Historisk bakgrund

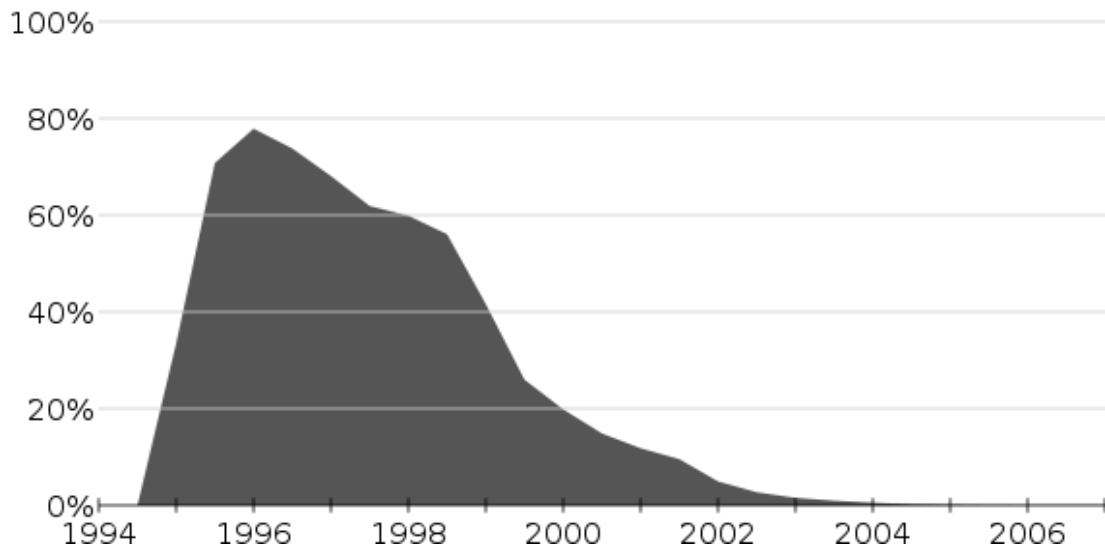
3.1 Java

Java började som ett projekt för den så kallade Green Team-arbetsgruppen i Sun Microsystems. År 1992 demonstrerade arbetsgruppen en hemmabiofjärrkontroll kallad *7. Fjärrkontrollen kunde styra apparater av olika tillverkare och med olika operativsystem, eftersom programvaran den utnyttjade var skriven i det plattformsoberoende språket Oak, utvecklat av James Gosling. Oaks namn ändrades senare till Java [4].

Eftersom portabilitet ofta är ett problem i nätverk (på grund av nätverksnoders diversitet), avsåg Green Team att Java skulle användas i nätverksapplikationer, specifikt kabeltelevision och dylikt. Då kabeltevevarmarknaden inte verkade mottaglig för teknologin, insåg arbetsgruppen att en tillämpning kunde vara Internet [4].

Green Teams andra demo var den Javabaserade webbläsaren WebRunner, som utgavs 1994. WebRunners innovation var att den kunde erbjuda dynamiskt innehåll på webbsidor, exempelvis animering [4].

Den första offentliga versionen av Java kom ut 1995. Samtidigt meddelade företaget Netscape (vars webbläsare Green Team hade utgått ifrån då de utvecklade WebRunner) att deras webbläsare Navigator skulle stöda språket [4]. Navigator var då den mest använda webbläsaren [8], vilket kan tänkas ha påverkat Javas popularitet som webbprogrammeringsspråk.



3-1: Netscape Navigators andel av webbläsarmarknaden 1994-2006 [8]

3.2 Python

Python har sina rötter i språket ABC, som utvecklades inom CWI (Centret för matematik och datavetenskap, på nederländska Centrum voor Wiskunde en Informatica) i Nederländerna på 1980-talet. CWI är känt även för utvecklandet av Algol 68 [9], ett mycket inflytelserikt programmeringsspråk.

ABC var avsett att vara lätt till och med för icke-programmerare att lära sig. Ett existerande språk, BASIC, hade liknande mål, men arbetsgruppen var missnöjd med hur det var förverkligat, bland annat behovet att använda sig av lågnivåprogrammering för att utföra vissa enkla uppgifter [6].

En av de personer som varit med i utvecklingen av ABC, Guido van Rossum, började senare arbeta med operativsystemet Amoeba i CWI. Han skapade Python som ett skriptspråk (ett språk för att automatiskt utföra instruktioner avsedda för en kommandotolk) för Amoeba, med ABC som grund. En av de största skillnaderna mellan ABC och Python var att van Rossum gjorde Python mer utökbart [6].

Van Rossum valde språkets namn innan han började utveckla själva språket. Namnet Python är en hänsyftning på den brittiska komedigruppen Monty Python. Utvecklingen av Python påbörjades i december 1989. Språket offentliggjordes i

februari år 1991 via nyhetsgruppen alt.sources, ett slags diskussionsforum på Internet [9].

4 Användarbas

TIOBE Software är ett holländskt bolag som specialiserar sig på undersökning av programvarukvalitet (<http://www.tiobe.com>). Enligt TIOBE var Java det andra mest använda och Python det åttonde mest använda programmeringsspråket i mars 2015 (Figur 4-1) utgående från mätningar av mycket använda sökmaskiner. Första och andra kolumnen anger språkets popularitet (där 1 betyder det mest använda) i mars 2015 respektive 2014 [2].

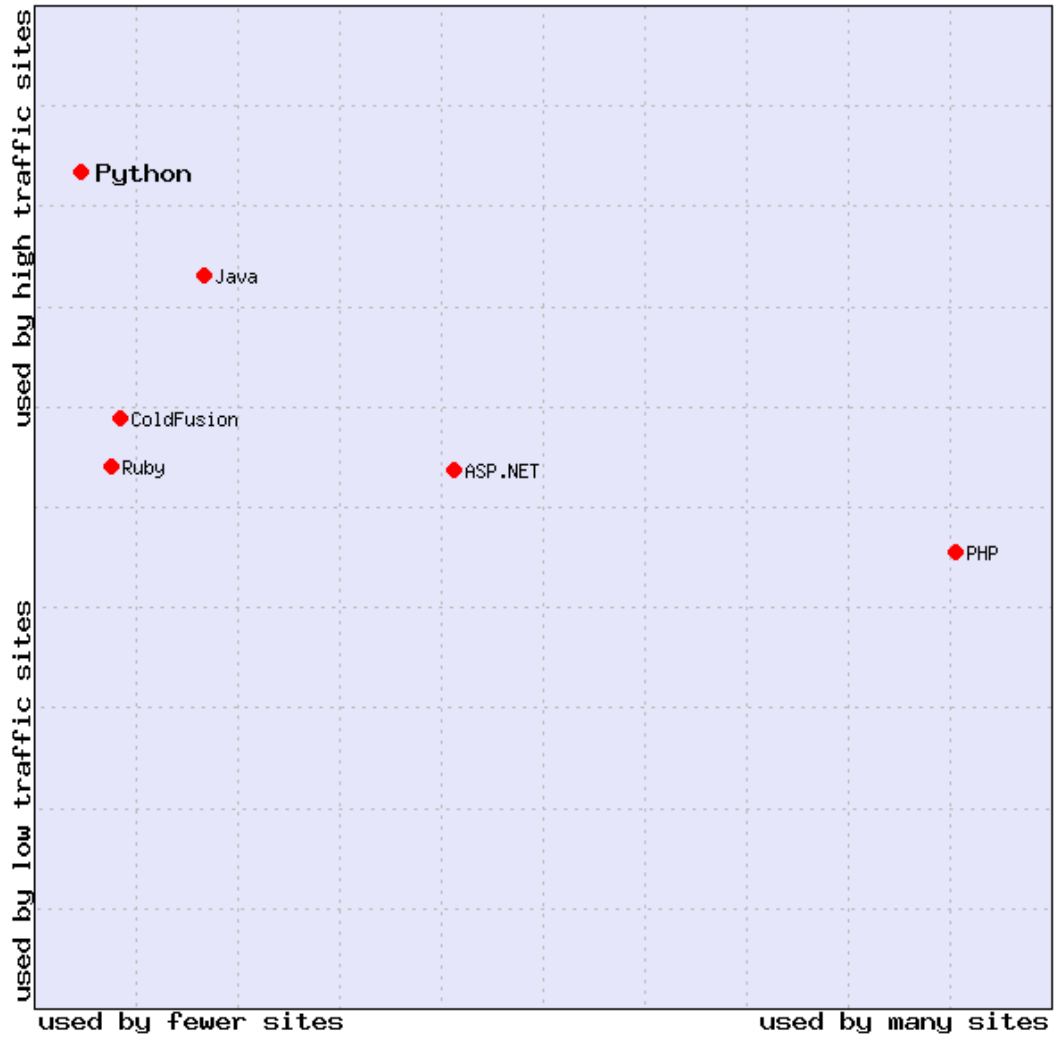
Av samma tabell framgår även att Javas andel av programmeringsspråken för tillfället avtar lite, medan Pythons ökar något. Ratings-kolumnen anger språkets andel av alla användare i procent, medan Change-kolumnen anger förändringen i denna andel i procentenheter mellan mars 2014 och mars 2015 [2].

Mar 2015	Mar 2014	Change	Programming Language	Ratings	Change
1	1		C	16.642%	-0.89%
2	2		Java	15.580%	-0.83%
3	3		Objective-C	6.688%	-5.45%
4	4		C++	6.636%	+0.32%
5	5		C#	4.923%	-0.65%
6	6		PHP	3.997%	+0.30%
7	9	▲	JavaScript	3.629%	+1.73%
8	8		Python	2.614%	+0.59%
9	10	▲	Visual Basic .NET	2.326%	+0.46%
10	-	▲	Visual Basic	1.949%	+1.95%

Figur 4-1: TIOBE:s register på programmeringsspråks popularitet [2]

W3Techs är en avdelning av det österrikiska företaget Q-Success, som rådgör i frågor om mjukvarukvalitet (<http://www.q-success.com>). Utgående från W3Techs mätningar används Python och Java speciellt mycket i hårt trafikerade servrar (Figur 4-2) [3]. Notera att TIOBE:s mätdata anger hur många som programmerar i språket, medan W3Techs anger hur många webbplatser som använder sig av språket.

Python Market Position, 9 Mar 2015, W3Techs.com



4-2: W3Techs graf för serverprogrammeringsspråks marknadsposition [3]

5 Syntax

5.1 Avgränsning av kodstycken

En av de mest märkbara skillnaderna mellan Javas och Pythons syntax, är att det förra använder sig av klammerparenteser för att markera gränserna för kodstycken [10, s. 23-24], medan det senare gör detta med hjälp av indrag och radbyten [11, s. 40]. Eftersom man enligt vedertaget bruk använder radbyten även i Javakod, gör detta ofta att antalet kodrader är mindre i Python, liksom i följande exempel:

Java:

```
if (a < b) {  
    doSomething();  
    doSomethingElse();  
}
```

Python:

```
if a < b:  
    doSomething()  
    doSomethingElse()
```

För Javaexemplet krävdes fyra kodrader, för motsvarigheten i Python endast tre. Detta beror naturligtvis på hur Javaprogrammeraren väljer att formatera sin kod. Ovanstående exempel kunde ha skrivits som

Java:

```
if (a < b) {doSomething(); doSomethingElse();}
```

och koden skulle fungera på samma sätt, men av konvention och för lätläslighetens skull är versionen med radbyten och indrag mycket vanligare. Av exemplet framgår också att varje sats måste avslutas med ett semikolon i Java, medan Python anger slutet av en sats helt enkelt med en ny rad (dock kan även stilen

```
if a < b: doSomething(); doSomethingElse()
```

användas i Python). Dessutom saknar Python parenteser kring `if`-satsens påstående, men har å andra sidan ett kolon efter det.

5.2 Klasser

En skillnad som endast märks i vissa fall, är språkens hantering av klasser. Klasser är en grundläggande del av Java, i och med att varje metod och variabel måste vara del av en klass [10, s. 23-24]. I Python är användning av klasser möjligt [11, s. 180], men inte nödvändigt. I Java måste dessutom offentlig (`public`) klass definieras i en enskild fil [11, s. 23-24]. Om man alltså har 15 offentliga klasser i sitt Javaprogram, behöver man 15 filer. Inget sådant krav finns i Python.

I samband med detta kan nämnas att de attribut som används i Java för att bestämma tillgängligheten av ett fält (`private`, `public` och så vidare) [10, s. 23-24] fattas helt och hållet i Python; alla fält i Python är offentliga (det vill säga motsvarar Javas `public`). Enligt vedertaget bruk betecknas dock fält, som man inte vill att ska användas direkt utanför klassen (motsvarande Javas `private`), med inledande understreck [11, s. 186-187], till exempel

Python:

```
class SomeClass:  
    self._object = "Variabeln _object är privat"
```

5.3 Typer

Javas typning är statisk. Då en variabel deklarerar, måste dess typ (heltal, flyttal, tecken och så vidare) definieras uttryckligen [10, s. 28]. Typningen i Python är dynamisk, det vill säga en variabel initialiseras utan att dess typ definieras (utom underförstått). Dessutom kan en variabels typ ändras i mitten av ett program [11, s. 19]. Följande exempel åskådliggör skillnaden:

Java:

```
String abcString = "abc";
```

Python:

```
abcString = "abc"
```

I exemplet initialiserades en variabel med namnet `abcString` som en textsträng med värdet `"abc"`. I Java definieras det som en sträng med hjälp av nyckelordet `String`. I Python är `abcString` en sträng eftersom det initialiserats med värdet `"abc"`, en sträng. I Pythonexemplet kunde man tilldela `abcString` värdet `1`, och variabelns typ skulle ändras till ett tal, medan ett försök att tilldela ett liknande värde i Java skulle leda till ett fel.

Dynamisk typning gör Python något enklare, men uttrycklig bestämning av en variabels typ kan ibland vara tydligare.

6 Prestanda

6.1 Mätningar

TechEmpower är ett amerikanskt företag som verkar inom IT-industrin (<http://www.techempower.com>). Företaget har gjort flera mätningar av prestationsförmågan för 93 av de mest använda webbserverramverken. Det hittills nyaste av dessa slutfördes 1 maj 2014. Ramverkens prestanda mättes med tre olika slags hårdvara [12], i följande kategorier:

- JSON-serialisering
 - Servern svarar med ett JSON-objekt. (JSON är ett filformat som kan användas för att serialisera objekt, det vill säga ändra abstrakta objekt till text [1, s. 107-109].)
- Enkel förfrågning (eng. Single query)
 - Servern hämtar vid varje förfrågning en rad från en databas och skickar den som ett JSON-objekt.
- Flera förfrågningar (eng. Multiple queries)
 - Samma som för enkel förfrågning men med flera rader.
- Fortunes
 - Servern hämtar alla rader från en tabell vars storlek den inte känner till. Tabellen innehåller Unix Fortune Cookies, en typ av slumpmässiga meddelanden, och en till Fortune Cookie läggs till under utförandet. Servern skickar sedan dessa Fortune Cookies som HTML-kod.
- Datauppdateringar (eng. Data updates)
 - Servern hämtar flera rader från en databas och ändrar dem till objekt i minnet. Sedan gör den en ändring till varje objekt och sparar ändringarna i databasen. Till sist skickar servern dessa data som JSON-objekt.
- Klartext (eng. Plaintext)
 - Servern skickar en vanlig textsträng ("Hello, World") [12].

6.2 Resultat

De Javabaserade ramverken presterar alldeles klart bättre än Pythons. De enda mätningarna där det är oklart vilket språks ramverk är snabbare, är datauppdateringarna, och även då endast med en slags hårdvara. I alla andra kategorier är alltid minst nio av de 14 Javaramverken snabbare än det snabbaste av de 18 Pythonramverken i mätningen [12].

Vad som också är intressant, är att Java inte endast klarat sig bra i jämförelse med Python, utan också jämfört med de andra språken. I varje kategori är något Javaramverk bland de tre snabbaste ramverken [12].

7 Tillgängliga verktyg

Både Java och Python har inbyggda verktyg för vanliga serverrelaterade uppgifter, såsom ORM (Object Relational Mapping), webbmallar (eng. templates) och förbindelser. I Java hör en stor del av dessa till Java EE (Java Platform, Enterprise Edition), plattformen avsedd främst för företag.

Trots att något av de inbyggda serververktygen i ena språket i de flesta fall har en motsvarighet i det andra, finns det vissa betydliga skillnader mellan språkens funktionalitet inom områden som är relevanta för serverapplikationer. De skillnader som tas upp här är i flertrådning och webbserverramverk.

Båda språken har klasser för flertrådning i sina standardbibliotek, men Pythons (CPython) är mer begränsad. Detta beror på att Python använder sig av ett GIL-lås (Global Interpreter Lock) som gör att endast en tråd kan utföras åt gången [13, s. 307-308]. Tack vare detta är Javaprogram mycket mer skalbara än Pythonprogram, vilket är särskilt viktigt i fråga om servrar med hög belastning.

Java EE innehåller ett standardramverk för serverapplikationer, JavaServer Faces (JSF) [14, s. 195-197]. Flera tredjepartsramverk finns tillgängliga för Python, men inget standardramverk. Tillgång till ett standardiserat ramverk kan vara nyttigt, eftersom sannolikheten att det stöds och kommer att stödjas av utvecklare är större.

I flera fall kan de ovannämnda bristerna i Python överbyggas med användning av någon alternativ implementation av Python. Jython är en implementation skriven i Java, i motsats till den vanligaste och ursprungliga implementationen skriven i C, CPython. Fördelar med Jython är att den saknar GIL [15, s. 417], och att man kan importera Javaklasser och använda deras funktionalitet i Pythonkoden [15, s. xxvi].

Det är värt att nämna ett serverramverk som finns tillgängligt för Python, nämligen Django. Förutom att vara det populäraste serverramverket för Python, är det dessutom mer använt än JSF eller något annat serverramverk för Java (utgående från deras popularitet på Github, och antalet frågor på Stack Overflow som har ramverkets namn som nyckelord) [16]. Detta betyder inte nödvändigtvis att Django är bättre, men

en stor användarbas innebär åtminstone att det är lättare att hitta programmerare som är kompetenta i användningen av ramverket.

8 Slutsatser

Javas övergång till serverapplikationer kan ses som ett naturligt steg från dess rötter som ett språk som kördes i webbläsaren. I Pythons tidiga historia finns inget som specifikt skulle knyta det till webbapplikationer. Eftersom Pythons sämre prestanda dock inte är ett lika stort problem i serverprogram, där flaskhalsen vanligen är annanstans, är servrar en av de mest logiska tillämpningarna för språket.

Java verkar ha den större användarbasen av de två språken (i allmänhet, inte specifikt i webbservrar). Å andra sidan växer Pythons användarbas till synes något snabbare än Javas. Dessutom består de servrar som använder sig av Python ännu mer av starkt trafikerade sådana, än de som använder Java.

Pythons syntax är klart enklare och koncisare än Javas, vilket kan tänkas leda till snabbare produktivitet med det tidigare. Vissa programmerare kan dock föredra Javas statiska typning och dess klassbaserade läggning. Det är svårt att hitta någon motivering för Javas användning av klammerparenteser och semikolon, utom att dessa är vanligt förekommande i andra programmeringsspråk.

Java är rätt entydigt det snabbare språket, åtminstone i serverramverk. Detta är av stor betydelse i servrar med mycket trafik, eller med program som är av mer resursmässigt krävande natur (vilket är något besynnerligt med tanke på att Python används speciellt mycket i hårt trafikerade servrar). Eventuellt kan det ändå finnas vissa specialiserade applikationer, där Python kunde prestera bättre.

I fråga om tillgängliga verktyg är båda språken bra försedda. Detta gäller såväl inbyggda egenskaper som de tredjepartsverktyg som finns på marknaden. Ingetdera språket har någon tydlig fördel i detta område, utan det är närmast fråga om användarens individuella behov och subjektiva preferenser.

Generellt kan konstateras att Python, tack vare den produktivitet som dess enkelhet och korthet tillåter, är ett bra val för till exempel nyetablerade företag som vill komma ut på marknaden snabbt, och för programmeringsundervisning [17, s.106]

[18, s. 49]. Java är ett bra alternativ särskilt för större företag på grund av dess höga prestanda och stora användarbas.

Litteratur

- [1] B. Brinzarea-Iamandi, C. Darie och A. Hendrix (2009). *AJAX and PHP: building modern web applications*. Birmingham: Packt Publishing, 308 sidor, ISBN: 978-1847197726.
- [2] TIOBE Software. ”TIOBE Index for March 2015”. Uppdaterad mars 2015. Tillgänglig: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. [Hämtad 9 mars 2015].
- [3] W3Techs. ”Usage Statistics and Market Share of Python for Websites”. Uppdaterad 9 mars 2015. Tillgänglig: <http://w3techs.com/technologies/details/pl-python/all/all>. [Hämtad 9 mars 2015].
- [4] J. Byous. ”Java Technology: An Early History”. Uppdaterad maj 1998. Tillgänglig: [http://gcc.uni-paderborn.de/www/WI/WI2/wi2_lit.nsf/64ae864837b22662c12573e70058bbb4/abf8d70f07c12eb3c1256de900638899/\\$FILE/Java%20Technology%20-%20An%20early%20history.pdf](http://gcc.uni-paderborn.de/www/WI/WI2/wi2_lit.nsf/64ae864837b22662c12573e70058bbb4/abf8d70f07c12eb3c1256de900638899/$FILE/Java%20Technology%20-%20An%20early%20history.pdf). [Hämtad 9 mars 2015].
- [5] J. S. Harbour (2008). *Beginning Java Game Programming*. Boston: Thomson Course Technology, 368 sidor, ISBN: 978-1598634761.
- [6] B. Venners. ”The Making of Python”. Uppdaterad 13 januari 2003. Tillgänglig: <http://www.artima.com/intv/pythonP.html>. [Hämtad 9 mars 2015].
- [7] S. Riley (2004). *Game Programming with Python*. Hingham: Charles River Media, 470 sidor, ISBN: 978-1584502586.
- [8] G. Gromov. ”Roads and Crossroads of the Internet History”. Uppdaterad 2012. Tillgänglig: http://www.netvalley.com/cgi-bin/intval/net_history.pl?chapter=4.

[Hämtad 9 mars 2015].

- [9] G. van Rossum. "The History of Python". Uppdaterad 20 januari 2009. Tillgänglig: <http://python-history.blogspot.fi/2009/01/personal-history-part-1-cwi.html>. [Hämtad 9 mars 2015].
- [10] D. Parsons (2003). *Introductory Java*. New York City: Continuum, 320 sidor, ISBN: 978-0826461797.
- [11] J. P. Kasurinen (2009). *Python 3- ohjelmointi*. Jyväskylä: Docendo, 247 sidor, ISBN: 978-9510352731.
- [12] TechEmpower. "Round 9 Results". Uppdaterad 1 maj 2014. Tillgänglig: <https://www.techempower.com/benchmarks/#section=data-r9&hw=peak&test=json>. [Hämtad 10 mars 2015].
- [13] T. Ziadé (2008). *Expert Python Programming*. Birmingham: Packt Publishing, 372 sidor, ISBN: 978-1847194947.
- [14] K. Cole, R. McChesney och R. Raszka (2011). *Advanced Java EE Development for Rational Application Developer 7.5: Developers' Guidebook*. Ketchum: MC Press, 752 sidor, ISBN: 978-1931182317.
- [15] J. Juneau, J. Baker, V. Ng, L. Soto och F. Wierzbicki (2010). *The Definitive Guide to Jython: Python for the Java Platform*. New York City: Apress, 544 sidor, ISBN: 978-1430225270.
- [16] HotFrameworks. "Web Framework Rankings". Uppdaterad 2015. Tillgänglig: <http://hotframeworks.com/>. [Hämtad 15 mars 2015].
- [17] B. Millero och D. Ranum. "Freedom to succeed: a three course introductory sequence using Python and Java". *Journal of Computing Sciences in Colleges*, volym 22, första upplagan, oktober 2006.

- [18] K.K. Agarwal, A. Agarwal och M.E. Celebi. "Python puts a squeeze on Java for CS0 and beyond". *Journal of Computing Sciences in Colleges*, volym 23, sjätte upplagan, juni 2008.