

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი
ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი
კომპიუტერულ მეცნიერებათა დეპარტამენტი

ბექა ჭელიძე

**სატელეკომუნიკაციო კომპანიების ქსელში არსებული
აპარატურული უზრუნველყოფების აღწერა შპს „ახალი ქსელების“
მაგალითზე**

სამაგისტრო პროგრამა: საინფორმაციო სისტემები

სამაგისტრო ნაშრომი შესრულებულია საინფორმაციო სისტემებში მეცნიერების
მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი: **ბექან ღვაზერიძე,**

ფიზ.-მათ. მეცნიერებათა კანდიდატი,
ასოცირებული პროფესორი

თბილისი

2016

ანოტაცია

ნაშრომი ეხება პრობლემატიკას, რომელიც გულისხმობს ქსელური მოწყობილობების აღწერას, როგორც ფიზიკურად, ასევე ფინანსურად. სამომხმარებლო ბაზარზე არსებობს უამრავი პროგრამული პაკეტი, რომლებიც არცთუ იშვიათად გამოიყენებიან მსგავსი აღწერის ჩასატარებლად. აღვნიშნოთ ისიც, რომ ასეთი პროგრამული პაკეტების გადარჩევა და კონკრეტული კომპანიისთვის კონკრეტულ ამოცანაზე მორგება არ არის ადვილი და ცალკე ამოცანას წარმოადგენს, ვინაიდან შესაძლოა საჭირო გახდეს ამა თუ იმ პროგრამული პაკეტის გადაკეთება, რათა სრულად დააკმაყოფილოს კომპანიის მოთხოვნა, რისთვისაც საჭიროა დიდი დრო და რესურსები. საბოლოო მიზანია, შეიქმნას თანამედროვე პროგრამული უზრუნველყოფა, რომელიც სრულად დააკმაყოფილებს კომპანიის მოთხოვნებს, შეძლებს აპარატურული უზრუნველყოფის სრულად აღწერას და იქნება მარტივი და გასაგები მომხმარებლებისთვის.

ამოცანის განსახორციელებლად დეტალურად იქნა შესწავლილი კომპანიის მოთხოვნა. მოხდა პრობლემატიკის ცალსახა ფორმირება და ყველა შესაძლო სიტუაციის მაქსიმალურად გათვალისწინება. კომპანიის წარმომადგენლებთან ერთად შეიქმნა სისტემის მოდელი, ასევე დაზუსტდა, თუ პროგრამირების რომელი ენებისა და ტექნოლოგიების გამოყენება მოხდებოდა ამოცანის რეალიზებისას.

გამოყენებულია პროგრამირების შემდეგი ენები და ტექნოლოგიები: javascript, java, css, html, jquery, pl/sql. ასევე გამოყენებულია oracle- ის მონაცემთა ბაზა.

Annotation

The work involves the problem, which includes a description of the network devices, both physically and financially. There are many software packages on the market, which are often used to describe network devices. It should be noted that, it is not easy to select a desired software to solve current problem, as it may be required to remake the soft for fully meet the Requirements. All this requires time and resources. The goal is to create modern software, which fully describes the network devices and will be simple and comprehensible for customers.

For implementation the task there was studied in detail the customer's demand. There was strictly formalized existent problem to implement the task. After all there was established model of the system.

სარჩევი

შესავალი.....	5
თავი I ძირითადი მოთხოვნები და სისტემის აღწერა	6
ძირითადი მოთხოვნები	6
ახალი აპარატურის მიღებისა და აღრიცხვის მოდული.....	7
ძირითადი საწყობის მოდული	8
TRANSFER	9
აბონენტთა მომსახურების ცენტრების მოდული	10
აპარატურის ტექნიკური მხარდაჭერის მოდული.....	11
რეპორტირება	12
უპირატესობები.....	13
თავი II ტექნოლოგიების აღწერა.....	14
Oracle PL-SQL	14
პროგრამირების ენა Java.....	15
ჯავა და ინტერნეტი.....	16
აპლეტები.....	16
ბაიტ-კოდი.....	17
განსაკუთრებული თვისებები	17
დასკვნა	18
გამოყენებული ლიტერატურა	19

შესავალი

კომპიუტერული ტექნიკის სწრაფმა განვითარებამ, სხვადასხვა საკომუნიკაციო არხების შექმნამ და დანერგვამ ჩვენს დროში აუცილებელი გახადა ელექტრონული და ციფრული ტექნოლოგიების გამოყენება თითქმის ყველა სფეროში. ამ ყველაფერს მოყვა სატელეკომუნიკაციო კომპანიების შექმნა, რომლებშიც იგლულისხმება როგორც სატელეფონო ასევე ინტერნეტის მომწოდებელი კომპანიები (პროვაიდერები). ისინი აღჭურვილნი არიან შესაბამისი აპარატურული უზრუნველყოფებით, რომლებიც უზრუნველყოფენ სერვისის მიწოდებას თითოეულ აბონენტამდე. დღითიდღე ასეთი კომპანიების მიერ შემოთავაზებული სერვისების რაოდენობა და ხარისხი იზრდება და ჩვენი ცხოვრების განუყოფელი ნაწილი ხდება. ასევე იზრდება აპარატურული უზრუნველყოფების რაოდენობები მათი სხვადასხვა დანიშნულების მიხედვით. სწორედ ასეთმა განვითარებამ გახადა აუცილებელი ისეთი პროგრამული უზრუნველყოფის შექმნა, რომელიც კონკრეტული პროფილის მქონე კომპანიისთვის შემღებს მის ბალანსზე არსებული ქსელური აპარატურის და მათი თანმხლები დეტალების აღწერას.

მიზანი იყო შემქნილიყო მოქნილი, თანამედროვე და მარტივად მოდიფიცირებადი ინფორმაციული სისტემა, რომელიც ზუსტად უპასუხებდა თანამედროვე გამოწვევებსა და მოთხოვნებს, ასევე მონაცემთა დაცვას და მთლიანობას. ეს უკანასკნელი კი სისტემის შექმნის დროს წამოჭრილ პრობლემებს შორის ერთ-ერთი მნიშვნელოვანია.

იქიდან გამომდინარე, რომ ამოცანა უნდა ყოფილიყო მორგებული კონკრეტული პროფილის მქონე კომპანიაზე და მის მოთხოვნებზე საჭირო გახდა მოთხოვნების მაქსიმალურად დეტალური შესწავლა. ამოცანის კვლევა შეიძლება დაიყოს შემდეგ ეტაპებად:

- 1) მოთხოვნების დეტალურად შესწავლა და მათი გაანალიზება, თუ რამდენად თანხვედრაშია ისინი ერთმანეთთან
- 2) ამოცანის დასმა ვერბალურ დონეზე.
- 3) კონსულტაციები დამკვეთებთან დასმული ამოცანის დასაზუსტებლად
- 4) ყველა შესაძლო უზუსტობის მაქსიმალურად გათვალისწინება.
- 5) სისტემის მოდელის შექმნა .
- 6) ამოცანის მოთხოვნებიდან და პროექტის ბიუჯეტიდან გამომდინარე ხელსაყრელი პროგრამირების ენებისა და ტექნოლოგიების არჩევა.

კვლევებისა და ჩატარებული სამუშაოების შედეგად შეიქმნა თანამედროვე და მოქნილი აღრიცხვის სისტემა, რომლის ინტეგრირება მოხდა უკვე არსებულ მონაცემებზე.

თავი I ძირითადი მოთხოვნები და სისტემის აღწერა

ძირითადი მოთხოვნები

- სისტემა იყოს მარტივად კონფიგურირებადი და მოდიფიცირებადი.
- შესაძლებელი იყოს სისტემის მორგება უკვე არსებული მონაცემების სტრუქტურაზე.
- შესაძლებელი იყოს არსებული ინფორმაციის სისტემაში შეტანა(იმპორტი) რაიმე წინასწარ განსაზღვრული ფორმატით (xml, csv, xls,xlsx.)
- მოხდეს კომპანიის ბალანსზე არსებული ქსელური აპარატურის სრული აღწერა, როგორც ფინანსურად, ასევე ტექნიკურად.
- მოხდეს თითოეული „რგოლის“ მკაცრი კონტროლი რომელსაც აპარატურა გაივლის.
- შედგეს ყველა საჭირო დოკუმენტი, როგორც ელექტრონულად, ასევე ბეჭდვითი ფორმით, რომლებიც საჭიროა აპარატურის აღრიცხვისთვის.
- შედგენილი დოკუმენტისა და მოწყობილობის სტატუსის მიხედვით კონკრეტულ აპარატურულ უზუნველყოფაზე პასუხისმგებლობა დაეკისროს შესაბამის პიროვნებას.
- მუშაობა შესაძლებელი იყოს ერთ ძირითად და რამდენიმე ქვესაწყობთან.
- შესაძლებელი იყოს დროის ნებისმიერ მომენტში მიმდინარე მდგომარეობის დაფიქსირება .
- პროცესის სრულად გასააზრებლად, სასურველდროის მონაკვეთში, შესაძლებელი იყოს ნახვა, თუ რომელი ობიექტიდან, რომელ ობიექტს, რა რაოდენობის აპარატურა გადაეცა.
- დროის ნებისმიერ მომენტში საწყობ(ებ)ის სრული ნაშთის ნახვა.
- აპარატურის გადაცემისას ცალსახად განისაზღვროს როგორც გამცემი, ასევე მიმღები პიროვნება.
- პროგრამის მომხმარებლები გაერთიანდნენ პროგრამულ როლებში (რომლებიც, თავის მხრივ გაერთიანებულია როლთა ჯგუფებში) . მათი უფლებები უნდა იყოს მკაცრად გამიჯნული ერთმანეთისგან.

ახალი აპარატურის მიღებისა და აღრიცხვის მოდული

ახალი აპარატურის მიღებისა და აღრიცხვის მოდული წარმოადგენს ელექტრონული აღრიცხვის დოკუმენტს (უწყისი). პროგრამის ამ მოდულში ხდება იმ უწყისების ელექტრონული აღრიცხვა, რომლითაც კომპანიამ მიიღო სხვადასხვა ტიპის ახალი აპარატურა. ძირითად ინფორმაციას წარმოადგენს: უწყისის ნომერი, დოკუმენტის შედგენის თარიღი, კომპანიის დასახელება, მომწოდებელი კომპანიის დასახელება, მიღებული აპარატურების ტიპების ჩამონათვალი და ტიპების მიხედვით აპარატურის რაოდენობა. საგარანტიო მომსახურების შემთხვევაში ხდება საგარანტიო თარიღისა და პირობების დაფიქსირება. პროგრამის მომხმარებელს აქვს ახალი ინვოისის დამატების, არსებულის რედაქტირების (თუ ინვოისი დახურული არ არის) წაშლის და ინვოისის დახურვის უფლება. დოკუმენტი ითვლება სისტემაში შეტანილად მას შემდეგ, რაც მომხმარებელი მას დახურავს. (იხ. სურათი#1)

ინვოისები

ინვოისები				
დამატება წაშლა ინვოისის დახურვა ინვოისის ბეჭედი				
ინვოისი #	თარიღი	კომპანია	დახურული	
1 inv_001	2016-01-12	ახალი ქველბი	☒	
2 beqa	2016-01-26	ახალი ქველბი	☒	
3 tel_050	2016-01-26	ახალი ქველბი	☒	
4 om_0001	2016-02-03	ახალი ქველბი	☒	
5 inv_0000123	2016-01-25	ახალი ქველბი	☒	
6 el-022016	2016-01-14	ახალი ქველბი	☒	
7 wwwwwwwww	2016-01-26	ახალი ქველბი	☒	
8 test27amvari	2016-01-27	ახალი ქველბი	☒	
9 test_007	2016-01-28	ახალი ქველბი	☒	
10 Test_002	2016-02-03	ახალი ქველბი	☒	
11 inv79879	2016-01-13	ახალი ქველბი	☒	
12 inv2222	2016-01-13	ახალი ქველბი	☒	
13 el-022000	2016-01-14	ახალი ქველბი	☒	
14 Test_001	2016-01-18	ახალი ქველბი	☒	
15 dadadadad	2016-01-04	ახალი ქველბი	☐	
16 INV_001	2016-01-18	ახალი ქველბი	☒	
17 testbeqam001	2016-01-23	ახალი ქველბი	☒	
18 tel_51	2016-01-28	ახალი ქველბი	☒	
19 EL_000001	2016-01-27	ახალი ქველბი	☒	
20. el_0000	2016-01-20	ახალი ქველბი	☒	

ინვოისის დეტალები				
დამატება წაშლა				
მოდელი	პროდუქტი	რაოდენობა		
-- ველი --				

ახალი ინვოისი
ინვოისი #:
თარიღი:
კომპანია:
შენახვა

დეტალური ინფორმაციის დამატება
მოდელი:
პროდუქტი:
რაოდენობა:
შენახვა

სურათი#1

ძირითადი საწყობის მოდული

ძირითადი საწყობის მოდულში ხდება მოწყობილობების თავდაპირველი აღრიცხვა და მათი შემდგომი გადანაწილება ქვესაწყობებზე ან შესაბამის პიროვნებებზე. თავდაპირველი აღრიცხვის დროს ხდება წინასწარ შედგენილი ელექტრონული უწყისის საფუძველზე ყველა იმ მოწყობილობის სისტემაში დამატება რომელიც კომპანიამ შეიძინა. მკაცრად არის განსაზღვრული უწყისის მიხედვით აპარატურის ტიპი და რაოდენობა. სისტემაში მოწყობილობის დამატება გულისხმობს მის მაკ მისამართისა და სერიული ნომრის უნიკალური წყვილის პროგრამაში შეყვანას.(იხ.სურათი#2)

საწყისი გვერდი

კონტაქტები

გახეილ

ახალი მოწყობილობების დამატება

უწყისი #:

000012

კომპანია:

ახალი ქსელები

ინვოისის #:

ინვოისი el-022016 : თარიღი 2016-01-14 00:00:00

პროდუქტის სრული დასახელება	მოწყობილობის ტიპი	რაოდენობა
F20 10/100	INBOX F20	50

მოწყობილობის ძებნა

S/N ან MAC

ძებნა

S/N	MAC	სტატუსი
-----	-----	---------

შესატანი მოწყობილობების რაოდენობა: 47

სერიული ნომერი

MAC მისამართი

შეწესება

სურათი#2

TRANSFER

პროგრამის ეს ნაწილი გამოიყენება მხოლოდ ძირითად ან ქვესაწყობებში. ქვემოთხსენებული მოიცავს შემდეგ ოპერაციებს: ძირითადი საწყობიდან ქვესაწყობზე აპარატურის გადაცემა, როგორც ძირითადი საწყობიდან ასევე ქვესაწყობიდან აპარატურის გაცემა, ქვესაწყობებიდან ძირითად საწყობში აპარატურის დაბრუნება, როგორც ძირითადი ასევე ქვესაწყობიდან აპარატურის ტექნიკური მხარდაჭერის ჯგუფში გადაცემა და დაბრუნება. საწყობზე დაქვემდებარებული ზონიდან აპარატურის დაბრუნება, აპარატურის კომპანიიდან კომპანიაზე გადაცემა (თუ პროგრამა ერთზე მეტ კომპანიას ემსახურება) და აბონენტთა მომსახურების ცენტრებიდან აპარატურის დაბრუნება. (იხ.სურათი #3)

საწყისი გვერდი

კომპანია

გასვლა

Transfer

ოპერაცია:

დაბრუნება საწყობში

სერვისცენტრი:

დაბრუნება საწყობში

გადემა

გადემა ქუთაისში

დაბრუნება ქუთაისიდან

კომპანიის შეცვლა

დაბრუნება სერვის ცენტრიდან

დაბრუნება ტექნიკური მხარდაჭერის განყოფილებიდან

დაბრუნება ზონიდან

გასცა:

Nothing selected

მოწყობილობის ძეგნა

სხვა მოწყობილობები

დასახელება

რაოდენობა

დამატება

Transfer Amounts

Process

დასახელება

მოწყობილობები

	S/N
1	x

სურათი #3

აბონენტთა მომსახურების ცენტრების მოდული

პროგრამის ეს ნაწილი ეკუთვნის მომსახურების ცენტრებს, უშუალოდ კომპანიის სააბონენტო მხარეს. მოდული იყოფა სამ ნაწილად : კომპანიის აბონენტზე აპარატურის გაცემა, აბონენტისგან აპარატურის დაბრუნება და დაზიანებული აპარატურის ჩანაცვლება.

პირველ შემთხვევაში ხდება კომპანიის აბონენტზე ხდება რაიმე სახის მოწყობილობ(ებ)ის გაცემა თანხმლები დეტალებით. დგება შესაბამისი დოკუმენტი(აქტი). ასევე, ფიქსირდება გაცემული მოწყობილობისა და მისი დეტალების ღირებულებები.

მერე შემთხვევაში პროგრამა გვიჩვენებს თუ რა ტიპის მოწყობილობა ჰქონდა აბონენტს. მოწყობილობის დაბრუნების დროს ხდება თითოეული დეტალის შემოწმება. აბრუნებს თუ არა აბონენტი მოწყობილობის პაკეტს სრულიად. დაუბრუნებლობის შემთხვევაში ხდება აბონენტზე შესაბამისი თანხის დარიცხვა.

მესამე შემთხვევაში ხდება დაზიანებული მოწყობილობის ან მისი დეტალის ახლით ჩანაცვლება. ასევე ხდება მოწყობილობების შესაბამისი ღირებულების გადაანგარიშება (იხ. სურათი#4)

აქტი მიღება-ჩაბარების და დაზიანებული მოწყობილობების შეცვლის შესახებ

ოპერაცია					
☒ აბონენტზე მოწყობილობების გაცემა	თარიღი: 16-06-2016				
☐ აბონენტისგან მოწყობილობების მიღება	ინსტალატორი: ადამია პაატა				
☐ აბონენტი არ არსებობს					
☐ დაზიანებული მოწყობილობების შეცვლა					
მოწყობილობები					
Serial	MAC	დასახელება	ფასი	მიღებულია	შენიშვნა
გატარებული მოწყობილობები					
	SIN	MAC	დასახელება		
1	x				

ქმედებები	
აბონენტი	
ტელეფონი:	
გადამხდელი:	
აბონენტი:	
მისამართი:	
კონტაქტი:	
სტატუსი:	
შიფრი:	
საიდენტიფიკაციო კოდი:	
დადგენის თარიღი:	
პასპორტი:	
მოწყობილობის ტიპი	

აპარატურის ტექნიკური მხარდაჭერის მოდული

მოდული მოიცავს აბონენტთა მომსახურების ცენტრებიდან მოწყობილობების ტექნიკური მხარდაჭერის განყოფილებაში გადაცემას, სადაც ხდება თითოეული მოწყობილობის ტესტირება და დადგენა იმისა, ვარგისია თუ არა შემდგომი გამოყენებისთვის. თუ მოწყობილობა ვარგისია, ხდება მისი საწყობში დაბრუნება და შემდეგ მისი კვლავ გამოყენება. ხოლო თუ მოწყობილობა დაზიანებულია, ხდება მისი ძირითად საწყობში დაბრუნება და ჩამოწერა დასკვნის საფუძველზე.

(იხ. სურათი#5)

 საწყისი გვერდი

მოდემების შემოწმება

შემოწმება

S/N:

MAC:

სტატუსი:

დაზიანებული

შეწმება

გაუქმება

დაზიანებული

ვარგისია

სურათი#5

რეპორტიზი

რეპორტიინგი ერთ-ერთი ძირითადი და გამოყენებადი მოდულია მთელს სისტემაში. სწორედ ამ მოდულში ხდება მთელი სისტემის მუშაობის ხარისხის შეფასება. ასევე ისეთ კითხვებზე პასუხების გაცემა როგორიცაა: (იხ. სურათი#6)

- მოცემული დროის განმავლობაში რამდენი მოწყობილობის შესყიდვა მოხდა?
- რამდენი და რა ტიპის მოწყობილობაა დარჩენილი საწყობში?
- მოცემული დროისთვის რამდენი და რა ტიპის მოწყობილობა გაიცა საწყობიდან. მათ შორის რამდენი მოწყობილობის მიხედვით მოხდა აბონენტზე?
- მოცემული დროის განმავლობაში რამდენი მოწყობილობა გაიცა და დაბრუნდა საწყობში?
- რამდენი და რა ტიპის დაზიანებული მოწყობილობის შეცვლა მოხდა?

საწყისი გვერდი
Billing Developer

უწყისების ისტორია

უწყისები
გე მასშტაბი / სეროული ნომერი

კომპანია: ახალი ქსელები | პერიოდში: 2016-05-27 | ჩამატა: | გასცა:

სტატუსი: | ზონა: | რეგიონი: | Search

	ნომერი	მუდგენის თარიღი	სტატუსი	გასცა	ჩამატა	სერვისცენტრი	მიწოდება	კომპანია	ზონა	რეგიონი
1	000159	2016-03-02	ვადაცმა ტექნიკური მხარდაჭერის განყოფილებაში	ats43122	TL უცვლელი	ქეთოსი	დახურულია			ქეთოსი
2	000668	2016-05-27	ვადაცმა ტექნიკური მხარდაჭერის განყოფილებაში	ats43122	TL უცვლელი	ქეთოსი	დახურულია			ქეთოსი
3	000618	2016-05-20	ვადაცმა		თირნიკ ფარმაცევტი		დახურულია	ახალი ქსელები	ჭანიკასი	თბილისი
4	000381	2016-04-08	ფარმაცეპს სერვის ცენტრიდან	რესკა ვეპასლი	საწყობი	ისანი	დახურულია			თბილისი
5	000545	2016-05-10	ფარმაცეპს სერვის ცენტრიდან	რესკა ვეპასლი	საწყობი	ისანი	დახურულია			თბილისი
6	000745	2016-06-07	ფარმაცეპს საწყობი	ხატი მუშყედინი	საწყობი	სამგორი	დახურულია			თბილისი
7	000544	2016-05-10	ფარმაცეპს საწყობი	ხატი მუშყედინი	საწყობი	სამგორი	დახურულია			თბილისი
8	000268	2016-03-24	ფარმაცეპს სერვის ცენტრიდან	ხატი მუშყედინი	საწყობი	სამგორი	დახურულია			თბილისი
9	000332	2016-04-01	ფარმაცეპს სერვის ცენტრიდან	ნონა ვეპასლი	საწყობი	სამგორი	დახურულია			თბილისი
10	000541	2016-05-10	ფარმაცეპს საწყობი	ანა კუდშვილი	საწყობი	ელდამი - ქსელები	დახურულია			თბილისი

10
H
<
ვერდი 1
64-დან
>
H
O

ჩანაწერების გადახედვა: 1-დან 10-მდე (ჯამად 640)

სურათი#6

უპირატესობები

ჩამოვთვალოთ ამ სისტემის რამდენიმე უპირატესობა რომლითაც განსხვავდება სხვა მსგავსი სისტემებისგან. ხშირად არის შემთხვევა როცა გარკვეული მიზეზების გამო შეიძლება არ გვქონდეს კავშირი ინტერნეტთან, შეიძლება იყოს ეს ადგილმდებარეობა ან ინტერნეტის არასაკმარისი სიჩქარე რის გამოც ვერ მოხერხდეს კავშირი ცენტრალურ სერვერთან. ამ შემთხვევაში პროგრამას შეუძლია იმუშაოს offline რეჟიმში. წინასწარ შეთანხმებულ დროს პროგრამა ამოწმებს კავშირს ინტერნეტთან და აკეთებს მონაცემების სინქრონიზაციას.

ასევე ხშირია შემთხვევა როცა სისტემის დანარგვა გვიწევს უკვე არსებულ რაიმე სახის ინფორმაციაზე რომელიც ინახება სხვა მონაცემთა ბაზაში ან უბრალოდ ფაილებში (.xlsx.xls.csv) ამ შემთხვევაში შესაძლებელია ინფორმაციის სისტემაში სტრუქტურირებულად შეტანა მოკლე დროში.

მიუხედავად იმისა რომ მაქსიმალურად არის დაცული მონაცემთა ბაზა არასწორი ან არასანქცირებული ინფორმაციის შეღწევიდან არსებობს ალბათობა რომ არასასურველი ან ბიზნესპროცესის ლოგიკიდან ამოვარდნილი ინფორმაცია მოხვდეს მონაცემთა ბაზაში. ამისთვის არსებობს სისტემაში არსებული მონაცემების ავტომატური მონიტორინგი, რომელიც წინასწარ განსაზღვრულ დროს შესრულდება და წინასწარ გაწერილი კრიტერიუმებით „გაასუფთავებს“ მონაცემებს.

არასანქცირებული ქმედებებისგან და შემდგომი გაურკვევლობებისგან თავის აცილებისთვის არსებობს ე.წ აუდიტ სქემები სადაც ინახება პროგრამის მომხმარებლის მიერ ჩატარებული ქმედებები და დრო როდესაც ეს განხორციელდა.

თავი II ტექნოლოგიების აღწერა

Oracle PL-SQL

მილიონობით პროგრამისტი და მონაცემთა ბაზის ადმინისტრატორი მსოფლიოს მასშტაბით რთული პროგრამული უზრუნველყოფების შესაქმნელად რომლებიც მუშაობენ დიდი მოცულობის მონაცემებთან იყენებენ პრდუქტს რომელიც შექმნილია კომპანია „ორაკლის“ მიერ. ორაკლის უდიდესი ნაწილი დაფუძნებულია და იყენებს პროგრამირების ენას PL/SQL, რომელიც წარმოადგენს სტრუქტურირებული ენის SQL ის პროცედურულ გაფართოებას.

PL/SQL არის ერთ-ერთი წარმატებული დაპროდუქტი რომელსაც ორაკლი გვთავაზობს. პროფესიონალი პროგრამისტები მას იყენებენ სხვადასხვა რთული ამოცანების გადასაჭრელად როგორცაა:

- ძირითადი ბიზნეს-ლოგიკის რეალიზება მონაცემთა ბაზის მხარეს სხვადასხვა პროცედურებისა თუ ტრიგერების დახმარებით.
- სხვადასხვა დოკუმენტების გენერაცია და მართვა მონაცემთა ბაზის მხარეს (მაგალიტად მონაცემების მიმოცვლა XML დოკუმენტით)
- ვებ გვერდის პირდაპირი კავშირი ორაკლის მონაცემთა ბაზასთან.
- რეალიზაცია და ავტომატიზაცია მონაცემთა ბაზის ადმინისტრატორისთვის განკუთვნილი სხვადასხვა ამოცანებისა, დაწყებული ქვედა დონის სისტემური უსაფრთხოებიდან და დამთავრებული ტრივიალური ამოცანებით

PL/SQL ენა შეიქმნა პროგრამირების ენა ADA ბაზაზე, რომელიც თავის მხრივ შეიქმნა ამერიკის შეერთებული შტატების თავდაცვის სამინისტროს დაკვეთით. ADA წარმოადგენს მაღალი დონის პროგრამირების ენას რომელიც იპყრობს დიდ ყურადღებას თავისი შესაძლებლობებიდან გამომდინარე. მას აქვს აბსტრაგირების მაღალი შესაძლებლობები, ასევე მონაცემების დაცვისა და უსაფრთხოების თანამედროვე მექანიზმები. ტავისი მოქნილი არქიტექტურის წყალობით PL/SQL გახდა ერთერთი უძლიერესი ენა რომელიც მოიცავს სხვადასხვა თანამედროვე პროცედურული ენების თითქმის ყველა შესაძლებლობას.

- მონაცემთა ტიპების ფართო არჩევანი, დაწყებული რიცხვითი ტიპებიდან დასრულებული რთული სტრუქტურირებული ტიპებით, როგორიც არის მაგალითად სხვადასხვა სახის კოლექციები, XMLType რომელიც გამოიყენება უშუალოდ xml დოკუმენტებთან სამუშაოდ.
- ნათელი და მკაფიო ბლოკური სტრუქტურა, რომელიც გასაგებს და მარტივს ხდის ენასთან მუშაობას.
- პირობითი, ციკლური და რიგითი მართვის სხვადასხვა სტრუქტურები. ასევე case პირობების გამოყენებით სამი სხვადასხვა ტიპის ციკლები.
- მონაცემთა ბაზაში რაიმე მოვლენის აღძვრისას წარმოქმნილი გამონაკლისი შეთხვევების დამუშავების მოქნილი შესაძლებლობები.

- განმეორებითი გამოყენებისთვის შექმნილი ელემენტები როგორებიცაა, პროცედურები, ფუნქციები, ტრიგერები, ობიექტური ტიპები (ისეთივე როგორიც კლასები ობიექტზე ორიენტირებული პროგრამირების შემთხვევაში), პაკეტები (ჯგუფი ლოგიკურად დაკავშირებული პროცედურების, ფუნქციების სხვადასხვა ტიპებისა და ცვლადების)

PL/SQL მჭიდრო კავშირშია oracle sql თან. მისი ინსტრუქციები შეიძლება პირდაპირ შესრულდეს პროცედურული პროგრამიდან რაიმე შუალედური ინტერფეისების გამოყენების გარეშე, მაგალითად როგორიცაა JDBC ან ODBC. ასევე შესაძლებელია პირიქითაც.

პროგრამირების ენა Java

ჯავას განვითარებას საფუძველი ჩაეყარა 1991 წელს ჯეიმს გოსლინგის და მისი რამდენიმე თანამშრომლის მიერ, რომლებიც იმ დროისთვის მუშაობდნენ კომპანიაში „Sun Microsystems“. პირველი მუშა ვერსიის შექმნას დასჭირდა 18 თვე. თავდაპირველად პროგრამირების ახალმა ენამ მიიღო დასახელება Oak (მუხა). მაგრამ 1995 წელს გადაარქვეს სახელი და ეწოდა Java.

როგორი უცნაურიც არ უნდა იყოს, Java-ს შექმნის მოტივი იყო პლატფორმისგან დამოუკიდებლობა. უნდა ყოფილიყო შესაძლებელი მაქსიმალურად მოქნილი პროგრამული უზრუნველყოფების შექმნა, რომლებიც გამოიყენებდა ჩადგმულ ბაიტკოდს. მისი გამოყენება შესაძლებელი უნდა ყოფილიყო ისეთ ელექტრონულ მოწყობილობებში როგორიცაა: დისტანციური მართვის ხელსაწყოები, მიკროტალღური ღუმელები და სხვა. არ არის რთული მისახვედრი, რომ ასეთი ტიპის მოწყობილობებში გამოიყენება სხვადასხვა სახის მიკროპროცესორები. მათთან სამუშაოდ პროგრამირების ისეთი ენების გამოყენება როგორიცაა C ან C++ რთულია, რადგან მათზე დაწერილი პროგრამები უნდა დაკომპილირდეს სხვადასხვა კონკრეტული პლატფორმებისთვის. პრაქტიკულად C++ ენაზე დაწერილი პროგრამა შეიძლება დაკომპილირდეს ყველა პლატფორმისთვის, მაგრამ, ამისთვის საჭიროა შესაბამისი კომპილატორი, რომლის შექმნის ხარჯებიც არც თუ ისე ცოტაა და საჭიროებს დიდ დროს.

ზუსტად მაშინ, როდესაც ჩამოყალიბდა java-ს ძირითადი მახასიათებლები და პრინციპები, გამოჩნდა მეორე მნიშვნელოვანი ფაქტორი, რომელიც ითამაშებდა გადამწყვეტ როლს ამ ახალშექმნილი პროგრამირების ენის მომავალში. მეორე ძირითად ფაქტორში იგულისხმება მსოფლიო ქსელი (WWW, ინტერნეტი). იმ შემთხვევაში, თუ ჯავა ვერ უპასუხებდა ამ გამოწვევას ის დარჩებოდა როგორც საჭირო და მძლავრი პროგრამირების ენა, რომელიც გამოიყენება მხოლოდ ელექტრონული მოწყობილობებისთვის, მაგრამ იქნებოდა ბევრისთვის შეუმჩნეველი.

გარკვეული დროის განმავლობაში მულტიპლატფორმულობა უკანა პლანზე გადავიდა, რადგან დიდი ყურადღება ექცეოდა გამოთვლით ტექნიკას, რომელიც თავის მხრივ სამ დიდ ჯგუფად (პლატფორმად) გაიყო (Intel, Macintosh, UNIX). შესაბამისად, ძირითადად, იყვნენ პროგრამისტები, რომლებიც კონკრეტულად ერთ-ერთ პლატფორმაზე მუშაობდნენ. ინტერნეტის განვითარებასთან ერთად მულტიპლატფორმულობამ დაიბრუნა თავისი აქტუალურობა და ისევ დადგა დღის წესრიგში. იქიდან გამომდინარე, რომ ინტერნეტში ჩართულია უამრავი კომპიუტერი სხვადასხვა ოპერაციული

სისტემით და პროცესორით, საჭირო გახდა ერთი და იგივე პროგრამული უზრუნველყოფის განსხვავებულ სისტემაზე მუშაობა.

როგორც ვიცით, ჯავა თავისი თვისებებიდან გამომდინარე წარმოადგენს C, C++ ენების გაგრძელებას. ჯავას შემქმნელებმა იცოდნენ, რომ C ენის ცნობილი სინტაქსის და C++ ის ობიექტზე ორიენტირებულობის გამოყენება გახდიდა მას მიმზიდველს და ადვილად აღსაქმელს უკვე მოქმედი და გამოცდილი C/C++ პროგრამისტებისთვის.

როგორც აღინიშნა, პროგრამირების ენის განვითარება დამოკიდებულია ორ ძირითად ფაქტორზე: ადაპტაცია, ცვლილებები თანამედროვე გარემოში და პროგრამირების ახალი იდეების განხორციელება. ასევე ჯავამ გარკვეულწილად შეცვალა მიდგომა პროგრამის წერისადმი. კერძოდ, მან უფრო გააფართოვა და სრულყოფილი გახადა ობიექტზე ორიენტირებული მიდგომა, C++ ის გამოყენებით მან ხელი შეუწყო სხვადასხვა ბიბლიოთეკების დამუშავებას და ასევე მრავალნაკადიანობას, რომლებითაც გამარტივდა ინტერნეტთან წვდომა და მუშაობა. პროგრამირების ენა ჯავას ასეთი წარმატება განპირობებულია არა მისი ცალცალკე აღებული ტექნიკური მახასიათებლებით, არამედ მათი გაერთიანებით მიღებული ერთიანი ძლიერი კომპლექტით.

ჯავა და ინტერნეტი

ინტერნეტი დაეხმარა ჯავას, რომ გადმოსულიყო პროგრამირების ენებს შორის წამყვან პოზიციზე. თავისმხრივ, ჯავამ მოიპოვა ძლიერი გავლენა ინტერნეტზე. ჯავამ არა მხოლოდ გაამარტივა ვებ პროგრამების შექმნა, არამედ, ხელი შეუწყო ახალი გამოყენებითი ტიპის პროგრამების (აპლეტების) შექმნას. მათ გაამარტივეს ქსელური ნაკადების გამოყენება. ამასთანავე, ჯავამ შესაძლებელი გახადა ორი საკმაოდ რთული პრობლემის გადაჭრა, რომლებიც დაკავშირებულია ინტერნეტთან და ქსელურ პროგრამირებასთან: პორტაბელურობა და უსაფრთხოება.

აპლეტები

აპლეტი არის გამოყენებადი პროგრამა, რომელიც შექმნილია ინტერნეტში მონაცემების მიმოცვლისთვის და თავსებადია ვებ ბრაუზერთან. თუ აბონენტი გადავა ისეთ ლინკზე, რომელიც შეიცავს აპლეტს, იგი ავტომატურად ჩამოიტვირთება და გაეშვება ვებ ბრაუზერში. აპლეტები, როგორც წესი, იქმნებიან როგორც მცირე პროგრამები. ისინი ემსახურებიან მონაცემების წარმოდგენას სერვერიდან, მომხმარებლის ქმედების ლოკალურად გაანალიზებას და დამუშავებას სერვერზე გაგზავნის გარეშე. ასევე ისეთ მარტივ ქმედებებს როგორცაა, მაგალითად, სესხის პროცენტის დათვლა. აპლეტი საშუალებას იძლევა რიგი ფუნქციებისა სერვერის მხრიდან გადავიტანოთ კლიენტის მხარეზე.

აპლეტების გამოჩენამ გარკვეულწილად შეცვალა ვებ პროგრამირების მიდგომა. სერვერისა და კლიენტის მხარეებს შორის გადაეცემა ორი მსხვილი კატეგორიის ობიექტები: პასიური მონაცემები და დინამიური აქტიური პროგრამები. მაგალითად, ელექტრონულ ფოსტაში შეტყობინების წაკითხვა გულისხმობს პასიური მონაცემის გადაცემას. თვითონ აპლეტი წარმოადგენს დინამიურ შერულებად პროგრამას. ქსელური დინამიური პროგრამებისთვის სერიოზულ პრობლემას

წარმოადგენს უსაფრთხოებისა და პორტაბელურობის საკითხი. იქიდან გამომდინარე, რომ არ ვიცით თუ რამდენად დაცულია ან რა გარემოში უწევს მუშაობა კლიენტის კომპიუტერს, ასევე თუ რა სახის არქიტექტურის მქონეა ის და რომელი ოპერაციული სისტემა უყენია.

ბაიტ-კოდი

ჯავას ძირითადი განსაკუთრებულობა, რომელიც საშუალებას იძლევა გადაიჭრას ზემოთ ჩამოთვლილი პრობლემები, არის ის რომ კომპილატორი იძლევა არა შესრულებად კოდს-ბაიტ-კოდს. ბაიტ-კოდს იყენებს ჯავას ვირტუალური მანქანა (JVM- Java Virtual Machine). ვირტუალური მანქანის პირველი ვერსია შექმნილი იყო როგორც ბაიტ-კოდის ინტერპრეტატორი. ჯავა პროგრამის გარდაქმნა ბაიტ-კოდად მნიშვნელოვნად ამარტივებს მის შესრულებას სხვადასხვა გარემოში. სხვადასხვა პლატფორმისთვის საჭიროა მხოლოდ ვირტუალური მანქანის რეალიზება. თუ სისტემაში რეალიზებულია ჯავას ვირტუალური მანქანა მაშინ ამ პლატფორმაზე შესაძლებელია ჯავაზე შესრულებული ნებისმიერი პროგრამის რეალიზება. ყველა ვირტუალური მანქანა სხვადასხვა პლატფორმაზე, იშვიათი გამონაკლისების გარდა, ახდენს ერთი და იგივე ბაიტ-კოდის რეალიზაციას. თუ ჯავაზე შესრულებული პროგრამა დაკომპილირდა რომელიმე პლატფორმაზე დამოკიდებულ კოდში, მაშინ სხვადასხვა სისტემისთვის უნდა არსებობდეს სხვადასხვა ვერსია ერთი და იმავე კოდისა. ცხადია რომ ასეთი გადაწყვეტილება მიუღებელია.

ის ფაქტი, რომ ჯავაზე დაწერილი პროგრამა სრულდება ვირტუალურ მანქანაზე ასევე ზრდის მის უსაფრთხოებას. იქიდან გამომდინარე, რომ პროგრამას მართავს ვირტუალური მანქანა, მას შეუძლია იზოლაცია გაუკეთოს შესრულებად კოდს მთელი სისტემისგან, შესაბამისად, კოდი დაცულია. ასეთ შემთხვევაში პროგრამა ჯერ კომპილირდება შუალედურ ფორმაში და შემდეგ ინტერპრეტირდება ვირტუალური მანქანის მიერ, ამიტომ პროგრამის შესრულების დრო გაცილებით დიდია ვიდრე იმ შემთხვევაში, როდესაც პროგრამული კოდი კომპილირდება პირდაპირ შესრულებად კოდში. თუმცა ბაიტ-კოდი იმდენად ოპტიმიზირებულია, რომ მისი შესრულება უფრო მალე ხდება ვიდრე ეს მოსალოდნელი იყო.

ჯავას პირველი ვერსიის გამოსვლიდან მალევე შეიქმნა ტექნოლოგია „HotSpot“, რომელიც წარმოადგენს დინამიურ კომპილატორს ბაიტ-კოდისთვის. თუ დინამიური კომპილატორი შედის ვირტუალური მანქანის შემადგენლობაში, მაშინ არჩეული ბაიტ-კოდის ფრაგმენტი კომპილირდება შესრულებად კოდში ნაწილ-ნაწილ. გასათვალისწინებელია ისიც, რომ მთლიანი ჯავა პროგრამის შესრულებად კოდში ერთდროულად კომპილაცია მიზანშეუწონელია, რადგან ჯავა აწარმოებს სხვადასხვა შემოწმებებს რომლებიც სრულდება მხოლოდ შესრულების დროს. დინამიური კომპილატორი კოდის ინტერპრეტირებას ახდენს შესრულებისას, იმ თანმიმდევრობით, რომელსაც ის საჭიროდ ჩათვლის. უფრო მეტიც, დინამიური კომპილაციის პრონციპი ხელს უწყობს პროგრამის პროდუქტიულობის ზრდას. მთავარი ის არის, რომ დინამიური კომპილაციისას ბაიტ-კოდის მთავარი უპირატესობები, პორტაბელურობა და უსაფრთხოება ხელუხლებელი რჩება.

განსაკუთრებული თვისებები

ძირითადი ფაქტორი, რომლებმაც ჯავას შექმნა და მისი შემდგომი განვითარება განაპირობა არის პორტაბელურობა და უსაფრთხოება. მაგრამ არსებობს რამდენიმე ფაქტორი რომლებმაც დიდი

როლი ითამაშეს ჯავას საბოლოო ვერსიების შექმნაში. ჯავას დეველოპერების ჯგუფმა შეაჯამა და წარმოადგინა ძირითადი მახასიათებლების შემდეგი სია:

- სიმარტივე.
- უსაფრთხოება.
- პორტაბელურობა.
- ობიექტზე ორიენტირებულობა.
- საიმედოობა.
- მრავალნაკადიანობა.
- არქიტექტურულად დამოუკიდებლობა.
- მაღალი წარმადობა.
- განაწილება.
- დინამიურობა.

დასკვნა

სამაგისტრო ნაშრომის მიზანი იყო, მომხმარებლისთვის უკეთ გაეცნო დასმული პრობლემა, რაც გულისხმობს სატელეკომუნიკაციო კომპანიებისთვის სასაქონლო აღწერას, ასევე კვლევისა და ამოცანის ფორმულირების პროცესი, პრობლემის გადაჭრის გზები და ტექნოლოგიები.

კვლევის პროცესში, თავდაპირველად ჩამოყალიბდა მოთხოვნა კომპანიის წარმომადგენლების მიერ. მოთხოვნა მდგომარეობდა შემდეგში: უნდა შექმნილიყო მოქნილი მარტივად კონფიგურირებადი და მომხმარებლისთვის მარტივად გამოყენებადი პროგრამული უზრუნველყოფა, რომელიც შეძლებდა ბალანსზე არსებული აპარატურული უზრუნველყოფის აღწერას. კვლევის პროცესში ერთ-ერთ პრობლემას წარმოადგენდა ამოცანის დეტალიზაცია და მისი სრული მოძიება, რადგან პროგრამა მორგებული უნდა ყოფილიყო სხვადასხვა პროფილის მქონე თანამშრომლებზე, რომლებიც განსხვავებულ ჭრილში იყენებენ შექმნილ პროგრამულ უზრუნველყოფას. თითოეული რგოლისთვის შემუშავდა ამოცანის მოდელი, გამოიყო ის ნაწილები რომლებიც თანაკვეთაში იყო სხვადასხვა პროფილის თანამშრომლებისთვის. ყოველივე ამის შემდეგ შეიქმნა ამოცანის მოდელი. ასევე დიდ პრობლემას წარმოადგენდა ამოცანის რეალიზების პროცესში წამოჭრილი უზუსტობების აღმოფხვრა და ახალი პროგრამული უზრუნველყოფის მორგება უკვე არსებულ ინფორმაციაზე სტრუქტურის დარღვევის გარეშე.

საბოლოოდ მივიღეთ პროგრამული უზრუნველყოფა, რომელიც აღწერს კომპანიის ბალანსზე არსებულ ქსელურ მოწყობილობებს როგორც ფიზიკურად, ასევე ფინანსურად, არის მოქნილი და დროის ნებისმიერ მომენტში გვაჩვენებს სრულ სურათს. რა რაოდენობისა და ღირებულების მოწყობილობები აქვს შეძენილი კომპანიას. გვაჩვენებს საწყობების სრულ ნაშთს. ასევე არის მარტივად დანერგვადი და მომხმარებლისთვის იოლად გამოყენებადი.

გამოყენებული ლიტერატურა

- 1) Фейерштейн С., Прибыл Б. - Oracle PL-SQL для профессионалов. 6-е издание 2015
- 2) Jean Francois Oracle 11g SQL Tuning Full Course
- 3) Herbert Schidt Java 8 Ninth Edition