

## კლასების მოვალეობების განსაზღვრა პროგრამული სისტემის დაპროექტებისას

თეიმურაზ სუხიაშვილი, ირაკლი შურღაია  
საქართველოს ტექნიკური უნივერსიტეტი

### რეზიუმე

ობიექტორიენტირებული მიდგომით პროგრამული სისტემის დაპროექტების ძირითადი დანიშნულებაა მოთხოვნათა რეალიზების მექანიზმების ფორმირება, რომელიც გულისხმობს კლასების შექმნას და მათი მოვალეობების (ატრიბუტების, ოპერაციების) განსაზღვრას. კლასების იდენტიფიკაციისა და მოვალეობების რაციონალური განაწილებისათვის გამოვდივართ სისტემის ქცევის თავისებურებიდან. რაც გულისხმობს თითოეული მოთხოვნით განსაზღვრული მოქმედებების განაწილებას ობიექტებს შორის ანუ პრეცედენტის ქცევის განაწილებას მათში იდენტიფიცირებულ კლასებს შორის.

**საკვანძო სიტყვები:** UML დიაგრამები. ატრიბუტი. ოპერაცია. კლასი. შეტყობინება. აბსტრაქცია.

### 1. შესავალი

პროგრამული სისტემის დაპროექტება იწყება მოთხოვნების(პრეცედენტების) რეალიზების მექანიზმების ფორმირებით, რომელიც მოიცავს კლასების იდენტიფიკაციას, რომლებიც მონაწილეობენ პრეცედენტების მოვლენათა ნაკადების რეალიზაციაში, ქცევის, რომელიც რეალიზდება პრეცედენტის მიერ, განაწილებას კლასებს შორის (კლასების მოვალეობების განსაზღვრა), კლასების მოვალეობების საფუძველზე ატრიბუტებისა და ოპერაციების განსაზღვრას.

პრეცედენტის მოვლენათა ნაკადში ძირითადად სამი ტიპის კლასებს გამოავლენენ:

- **მოსაზღვრე კლასები (Boundary)** - შუამავლები გარე ობიექტებსა და სისტემას შორის ურთიერთქმედებისას. როგორც წესი, ყოველი წყვილისათვის „მოქმედი პირი – გამოყენებითი შემთხვევა“ განისაზღვრება ერთი მოსაზღვრე კლასი;

- **კლასები – არსი (Entity)** – დასამუშავებელი სისტემის ძირითადი აბსტრაქციებია (ცნებები). კლასი-არსის გამოვლენის წყაროა აბსტრაქციები, რომლებიც იქმნება პრეცედენტის მოვლენათა ნაკადის აღწერისას;

- **მმართველი კლასები (Control)** – უზრუნველყოფს ობიექტების ქცევის კოორდინაციას სისტემაში. როგორც წესი, ყოველი პრეცედენტისათვის განისაზღვრება ერთი მმართველი კლასი.

ყველა ისინი ერთობლივად ქმნიან ანალიზის კლასებს, რომლებიც გამოხატავენ სისტემისადმი ფუნქციონალურ მოთხოვნებს და ახდენს საგნობრივი სფეროს მოდელირებას. ანალიზის კლასთა ერთობლიობა სისტემის საწყისი კონცეპტუალური მოდელია.

გამოყოფილი სამი ტიპის კლასების დანიშნულებიდან გამომდინარე, შესაძლებელია მათ შორის მოვალეობათა დახასიათება:

- მოსაზღვრე კლასები პასუხისმგებელია სისტემის ურთიერთქმედებაზე გარე სამყაროსთან (მოქმედი პირები);

- კლასი არსები პასუხს აგებს მონაცემთა შენახვასა და მანიპულირებაზე;

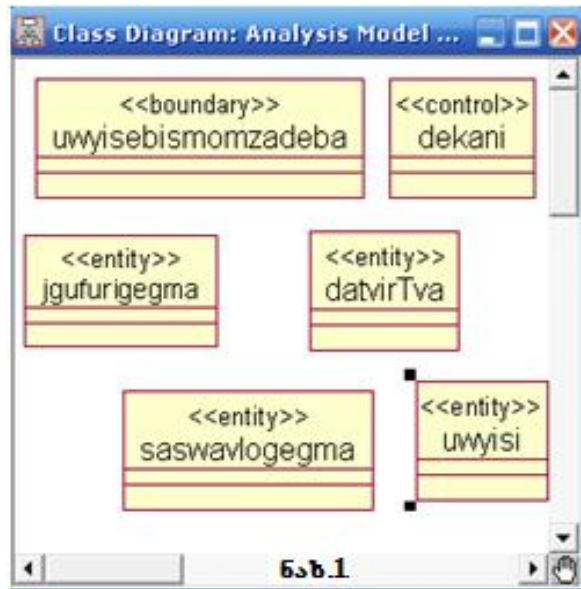
- მმართველი კლასები კოორდინაციას უწევს პრეცედენტების მოვლენათა ნაკადებს.

მოვალეობების უფრო დეტალური განაწილება (კლასების ოპერაციების სახით) სრულდება ურთიერთქმედების დიაგრამებით.

### 2. ძირითადი ნაწილი

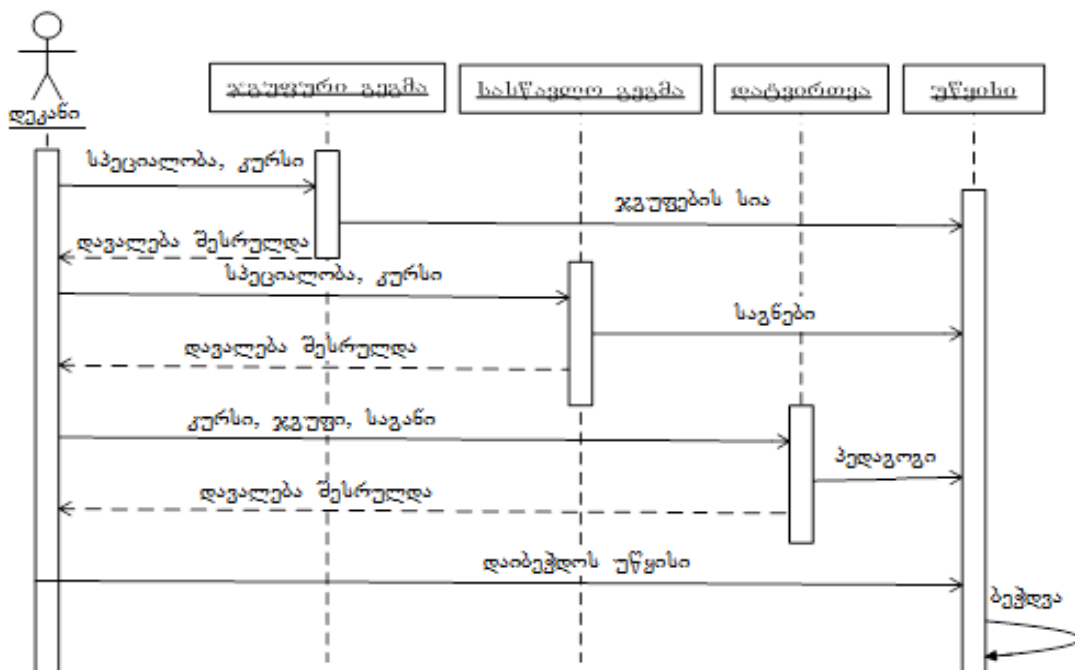
მოთხოვნის (პრეცედენტის) ქცევა გამოისახება ურთიერთქმედების დიაგრამით. პირველ რიგში აივება დიაგრამა, რომელიც აღწერს მონაცემთა ძირითად ნაკადს. ყოველი ალტერნატიული ნაკადისათვის იგება ცალკე დიაგრამა.

მაგალითად, განვიხილოთ „უწყისების მომზადების“ მოთხოვნის ანალიზი. ყოველ პრეცედენტში პროცესი შესაძლებელია სხვადასხვა სცენარით განვითარდეს, რომლებიც მოვლენათა ნაკადებით არის განსაზღვრული. ამ სცენარების შესწავლის საფუძველზე დგება ანალიზის კლასები, რომლებიც მოყვანილია 1-ელ ნახაზზე.



შემდეგ ეტაპზე ვახდენთ პრეცედენტის მიერ რეალიზებადი ქცევის განაწილებას კლასებს შორის. მას წარმოვადგენთ ურთიერთქმედების დიაგრამის სახით.

მე-2 ნახაზზე მოცემულია მიმდევრობის დიაგრამა მართვის ძირითადი ნაკადისათვის – უწყისის ყველა პარამეტრი დგინდება. ამ სურათზე ადამიანის ფიგურით გამოსახულია სისტემის მომხმარებელი, ხოლო დანარჩენი პროცესში მონაწილე ობიექტებია. ამ დიაგრამაზე ჩანს თუ რა მიმდევრობით ხდება შეტყობინებების გადაგზავნა ობიექტებს შორის სცენარის შესასრულებლად.



ნახ.2

როგორც სქემიდან ჩანს, დეკანი (მომხმარებელი) უთითებს სასურველ სპეციალობას და კურსს, რომელიც იცვლება 1-დან 4-მდე მოსაზღვრე კლასით „უწყისების მომზადება“.

აღნიშნული პარამეტრების საფუძველზე მმართველი ობიექტი **დეკანი** ობიექტიდან **ჯგუფური გეგმა** ადგენს ჯგუფის ნომრებს (N) და სტუდენტების სიას, რომლებიც მითითებულ სპეციალობასა და კურსზე ირიცხებიან. ამ მონაცემებს **ჯგუფური გეგმა** გადაუგზავნის ობიექტს **უწყისი** და ობიექტს **დეკანი** უბრუნებს შეტყობინებას, რომ მიღებული დავალება შეასრულა.

მიღებული შეტყობინების საფუძველზე მმართველი ობიექტი **დეკანი** ობიექტიდან **სასწავლო გეგმა** ადგენს დისციპლინებს (M), რომლებიც ისწავლება მითითებული სპეციალობის მოცემულ კურსზე, ამ მონაცემებს **სასწავლო გეგმა** გადაუგზავნის ობიექტს **უწყისი** და ობიექტს **დეკანი** უბრუნებს შეტყობინებას, რომ მიღებული დავალება შეასრულა.

მიღებული შეტყობინების საფუძველზე მმართველი ობიექტი **დეკანი** ობიექტიდან **დატვირთვა** ადგენს გამომცდელ პედაგოგს, რომელიც ასწავლის მოცემულ საგანს მითითებული სპეციალობის მოცემულ კურსზე, მოცემულ ჯგუფში. ამ მონაცემს ობიექტი **დატვირთვა** გადაუგზავნის ობიექტს **უწყისი** და ობიექტს **დეკანი** უბრუნებს შეტყობინებას, რომ მიღებული დავალება შეასრულა.

მიღებული შეტყობინების საფუძველზე მმართველი ობიექტი **დეკანი** აძლევს მითითებას ობიექტს **უწყისი**, რომლის საფუძველზე იქმნება დოკუმენტი **უწყისი** მოცემული სპეციალობის, კურსის, ჯგუფისა და დისციპლინისათვის. აღნიშნულ პროცესს მმართველი ობიექტი (დეკანი) იმეორებს დისციპლინების რაოდენობის (M-ჯერ), ჯგუფების რაოდენობის (N-ჯერ) და ბოლოს კურსების(4) მიხედვით.

ამ სცენარში მონაწილე სხვა ობიექტებთან ერთად ჩანს მმართველი (მენეჯერი) ობიექტი. ასეთ ობიექტებს ევალება მხოლოდ მართვა. ისინი უმეტესად კლიენტი ობიექტების როლში გამოდიან, ხოლო ისეთი ობიექტები როგორცაა დოკუმენტი - სერვერის როლში.

თპერაციები, რომელსაც ავალებს შესასრულებლად მმართველი (მენეჯერი) ობიექტი სხვა ობიექტებს, ფაქტობრივად განსაზღვრავს მათ მოვალეობას. ჩვენ მაგალითში, ობიექტს „ჯგუფური გეგმა“ ევალება ჯგუფის ნომრის და სიის დადგენა, ობიექტს „სასწავლო გეგმა“ საგნების, რომლებიც ისწავლება, ობიექტს „დატვირთვა“ პედაგოგის და ბოლოს „უწყისი“ ობიექტის მოვალეობაა დოკუმენტის (უწყისის) ფორმირება.

### 3. დასკვნა

კლასთა მოვალეობების დადგენა სისტემის ქცევის საფუძველზე საშუალებას გვაძლევს უფრო რაციონალურად გადავანაწილოთ მოვალეობები, შევქმნათ კლასები მოვალეობებით, რომლითაც სრულად და ეფექტურად სრულდება თითოეული მოთხოვნის და მთლიანად სისტემისადმი დაკისრებული ფუნქციები.

**ლიტერატურა:**

1. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. Серия “Объектно- ориентированные технологии в программировании”. Москва, 2004.
2. სუხიაშვილი თ.. პროგრამული სისტემის დამუშავების CASE საშუალებები. ელ-სახ., სტუ, „ტექნიკური უნივერსიტეტი“, 2013
3. სუხიაშვილი თ., რეალიზების მექანიზმების მოდელირება ობიექტ-ორიენტირებული პროექტირებისას. სტუ-ს შრ.კრ. № 1(2), 2007. გვ. 82-85.

**OPREDEDENIYE OF DUTIES OF CLASSES AT DESIGN OF PROGRAM SYSTEM**

Sukhiashvili Temur, Shurgaia Irakli  
Georgian Technical University

**Summary**

The main naenacheniye at design of program system by object-oriented approach formation of mechanisms of implementation of requirements that means creation of classes and definition of their duties (attributes, operation). For identification of classes and rational distribution of duties between them we proceed from features of behavior of system. That means distribution of actions defined for each requirement between objects i.e. distribution of behavior of precedent between the classes identified in it.

**ОПРЕДЕЛЕНИЕ ОБЯЗАННОСТЕЙ КЛАССОВ ПРИ ПРОЕКТИРОВАНИИ ПРОГРАММНОЙ СИСТЕМЫ**

Сухиашвили Т., Шургая И.  
Грузинский Технический Университет

**Резюме**

Основное назначение при проектирования программной системы объектно-ориентированным подходом - формирование механизмов реализации требований, что подразумевает создание классов и определение их обязанностей(атрибутов, операции). Для идентификации классов и рационального распределения обязанностей между ними исходим из особенностей поведения системы. Это подразумевает распределение между объектами действий, определенных каждым требованием, т.е. распределение поведения прецедента между идентифицированными в нем классами.