

Привет, друзья!

Мы Вам раскроем одну тайну, которую Вам не доведется узнать даже в университетах. Вы наверняка видели многих, кто даже 5 лет проучившись в университете на программиста так и не понял, что такое ООП (Объектно Ориентированное Программирование).

Обычно научное определение ООП звучит так

ООП = полиморфизм+инкапсуляция+наследование

или так

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования

Но что это значит на самом деле непонятно. А мы Вам расскажем простым человеческим языком.

Теперь давайте забудем про программирование и подумаем о себе. О своем человеческом организме. У каждого из нас есть разные части тела и разные органы. Вопрос - полностью ли мы можем собой управлять? Вопрос не праздный - и если Вы подумаете - то поймете ответ - НЕТ. Например, Вы хотите поднять руку - это легко. Вы хотите ненадолго задержать дыхание - это легко. Вы хотите что-нибудь сказать - это легко. А вот теперь Вы хотите остановить свое сердце силой мысли - можете? Нет, не сможете. Или например, Вы хотите силой мысли перестать переваривать пищу в кишечнике - сможете? Нет. Или Вы захотите не чувствовать боль - тоже не сможете.

Почему мы так странно устроены? Почему что-то мы с собой можем сделать - а что-то нет? Дело в том, что в нас самих прописана защита от того, чтобы мы сами себя не сломали и не умерли из-за неаккуратности. Внутри нас есть заборы, которые преодолеть нам не дано. Правда есть такие люди - называются йоги - которые все-таки на время умеют преодолевать эти заборы: они ходят по раскаленным углям и им не больно, они могут задерживать дыхание на 15-20 минут, замедляя все процессы в организме настолько - что не умирают.

Так вот величайшая тайна состоит в том, что ООП - это наука о строительстве внутри собственной программы искусственных заборов. Заборов от самого себя, либо от других разработчиков - которые будут разрабатывать проект вместе с Вами. Вы сами себе запрещаете много чего делать внутри собственной программы - чтобы случайно ее не сломать. И наука о заборах - это 99% программирования. А 1% - это собственно само программирование без заборов.

В науку о заборах помимо способов строительства заборов входит и наука о суперспособностях для их преодоления в особых случаях (да-да - это те самые йоги, ходящие по углям).

ООП необходим для написания крупных проектов, особенно превышающих 100000 строк кода. Но для меньших программ ООП часто создает избыточные трудности и вообще мешает.

Так например, первую версию ядра операционной системы Linux ее создатель Линус Торвалдс написал без ООП на чистом языке СИ и она была как раз 100 000 строк кода.

Но впоследствии все ядро было переписано сообществом разработчиков на объекто-ориентированном языке C++. И сейчас ядро Linux превышает 18 000 000 строк кода.

Как мы и говорили в одной из наших статей:

само по себе программирование - это всего лишь 20 слов - и научиться этому может каждый и очень быстро. И в первые годы обучения Вы научитесь любые свои идеи выражать этими 20-ью словами. Но когда Вы задумаете свою it-компанию, когда задумаете запрограммировать грандиозный командный проект - ООП уберезет Вас от многих ошибок.

Теперь Вы знаете что такое ООП и самое главное - зачем оно нужно. Приходите изучать программирование вместе с нами - это весело!

მოგესალმებით, მეგობრებო!

ჩენ გაგიმხელთ ერთ საიდუმლოს, რომელსაც თქვენ უნივერსიტეტშიც კი ვერ გაიგებდით. თქვენ შეიძლება იცნობდეთ ისეთ ხალხს, რომლებმაც დაამთავრეს უმაღლესი პროგრამისგან სპეციალობაზე და მაინც ვერ გაიგეს რა არის ოოპ (ობიექტ-ორიენტირებული პროგრამირება).

ჩვეულებრივ მეცნიერული გამონათქვამი ოოპ ჟღერს ასე

ოოპ = პოლიმორფიზმი+ინკაფსულაცია+მემკვიდრეობა

ან ასე

ობიექტ-ორიენტირებული პროგრამირება (ოოპ) - პროგრამირების მეთოდოლოგია, დაარსებული პროგრამის წარმოსახვაზე შერწყმული ობიექტების სახით, რომლის ყოველივე ობიექტი წარმოადგენს გარკვეული კლასის ეგზემპლარს, კლასები კი ქმნიან მემკვიდრეობის იერარქიას

მაგრამ რას ნიშნავს ეს გაუგებარია. ჩვენ კი - მოგიყვებით უბრალო გასაგებ ენაზე

ახლა კი მოდით დავივიწყოთ პროგრამირება და ვიფიქროთ ჩვენ თავზე. ჩვენ ადამიანურ ორგანიზმზე. ყოველ ჩვენგანს გააჩნია სხეულის სხვადასხვა ნაწილი და სხვადასხვა ორგანოები. კითხვა: შეგიძლიათ მართოთ თქვენი ორგანიზმი? ეს სერიოზული კითხვაა და თუ დაფიქრდებით, გაიგებთ პასუხს -

არა არ შეგიძლიათ. მაგალითად, გინდათ აწიოთ ხელი - ეს ადვილია. გინდათ ცოტახნით შეიჩეროთ სუნთქვა - ეს ადვილია. გინდათ რალაცის თქმა - ესაც ადვილია. ახლა კი თქვენ გინდათ გაიჩეროთ გული გააზრების ძალით - შეგიძლიათ? არა, არ შეგიძლიათ. ან გინდათ შეწყვიტოთ კუჭში საჭმლის გადაამუშავება - შეძლებთ? არა. ან გინდათ შეწყვიტოთ ტკივილის გრძნობა - ამასაც ვერ შეძლებთ.

რატომ ვართ ასე უცნაურად მოწყობილი? რატომ შეგვიძლია რალაცა გავაკეთოთ და რალაცა არა? საქმე იმაშია, რომ ჩვენშიც წინასწარ გაწერილია ეს ყველაფერი, იმისთვის, რომ შემთხვევით არ ვავნოთ ჩვენ ორგანიზმს და არ დავილეპოთ. ჩვენ შიგნით არის ღობეები, რომლის გადალახვა საჭირო არ არის. თუმცა არსებობენ ისეთი ადამიანებიც, რომლებსაც შეუძლიათ ღროებით გადალახონ ეს ღობეები: ისინი დადიან ცხელ ნაკორჩხალზე და მათ არ სტკივთ, მათ შეუძლიათ შეიკავონ სუნთქვა 15-20 წუთით, შეანელონ თავიანთი ორგანიზმის პროცესები ისე, რომ არ კვდებიან.

ასე რომ უდიდესი საიდუმლო იმაშია, რომ ოოპ - ეს არის მეცნიერება საკუთარი პროგრამის შიგნით ხელოვნური ღობეების აგებისთვის. ღობეები თქვენგან ან სხვა დევლოპერებისგან - რომლებიც თქვენთან ერთად შეიმუშავებენ პროექტს. თქვენ უკრძალავთ საკუთარ თავს შეიგანოთ ცვლილება საკუთარ პროგრამაში - რომ შემთხვევით არ გააფუჭოთ იგი. ხოლო 1% - ეს არის თვითონ პროგრამირება ღობეებს გარეშ.

ღობეების მეცნიერებაში, მათი აგების გარდა შედის მეცნიერება მათი გადალახვის სუპერშესაძლებლობებზე (კი, კი, კი - ეს ზუსტად ის ადამიანები არიან ვინც ნაკორჩხალზე დადიან, ან იკავებენ სუნთქვას 15-20 წუთით).

ოპო აუცილებელია მსხვილი პროექტების დასაწერად, მითუმეტეს თუ კოდის სიგრძე აღემატება 100 000 სტრიქონს. მაგრამ პატარა პროგრამებისთვის ოპო ქმნის ნამეტან პრობლემებს და საერთოდაც უშლის.

მაგალითად, ოპერაციული სისტემა Linux-ის ბირთვის პირველი ვერსია - ლინუს ტორვალდსმა დაწერა ოპო-ს გარეშე, სუფთა C ენაზე და იგი ზუსტად 100 000 სტრიქონიანი კოდს წარმოადგენდა.

მაგრამ ამის შედეგად მთელი ბირთვი გადაწერილი იყო ობიექტ-ორიენტირებულ ენაზე, C++ -ზე. ახლა Linux-ის კოდში სტრიქონების რაოდენობა აღემატება 18 000 000.

როგორც უკვე ვილაპარაკეთ ერთ-ერთ ჩვენ სტატიაში

თვითონ პროგრამირება - ეს არის სულ რაღაც 20 სიტყვა - მისი სწავლა შეუძლია ყველას და ძალიან სწრაფად. პირველ სასწავლო წელში თქვენ ისწავლით ამ 20 სიტყვის გამოყენებით ნებისმიერი იდეის გამოსახვას. მაგრამ როდესაც ჩაიფიქრებთ საკუთარ IT-კომპანიას, როდესაც მოისურვებთ გრანდიოზული პროექტის დაპროგრამებას ჯგუფური მუშაობით - ოპო დაგიცავთ სხვადასხვა შეცდომებისგან.

ახლა თქვენ იცით რა არის ოპო და რაც მთავარია - იცით რისთვის არის საჭირო. მობრძანდით პროგრამირების სასწავლოდ ჩვენთან - ეს საინტერესოა!

Hello friends!

We will reveal to you one secret that you will not have to learn even in universities. You have probably seen many who, after 5 years of studying at a university for a programmer, did not understand what OOP is (Object Oriented Programming).

Usually the scientific definition of OOP is:

OOP = polymorphism + encapsulation + inheritance

or so

Object-Oriented Programming (OOP) is a programming methodology based on the representation of a program as a collection of objects, each of which is an instance of a particular class, and the classes form an inheritance hierarchy

But what this means is really incomprehensible. And we will tell you in simple human language.

Now let's forget about programming and think about yourself. About your human body. Each of us has different parts of the body and different organs. The question is - can we completely control ourselves? The question is not idle - and if you think - then you will understand the answer - NO. For example, you want to raise your hand - it's easy. You want to hold your breath for a while - it's easy. You want to say something - it's easy. And now you want to stop your heart with the power of thought - can you? No, you cannot. Or, for example, you want to stop digesting food in the intestines with the power of thought - can you? Not. Or you want not to feel pain - you cannot either.

Why are we so strangely arranged? Why can we do something with ourselves - but something not? The fact is that we ourselves have a protection against the fact that we do not break ourselves and die because of inaccuracy. There are fences inside of us that we cannot overcome. The truth is, there are such people - they are called yogis - who still know how to overcome these fences for a while: they walk on hot coals and it does not hurt them, they can hold their breath for 15-20 minutes, slowing down all processes in their body so much that they do not die.

So the greatest secret is that the OOP is the science of building artificial fences inside your own program. Fences from yourself. You yourself forbid a lot of things to

do inside your own program - so as not to accidentally break it. And the science of fences is 99% of programming. And 1% is actually self programming without fences.

The science of fences, in addition to the methods of building fences, also includes the science of superpowers for overcoming them in special cases (yes, these are the same yogis walking on coals).

OOP is necessary for writing large projects, especially those that exceed 100,000 lines of code. But for smaller programs, the OOP often creates excessive difficulties and makes these projects difficult at all.

For example, the first version of the Linux kernel was written by its creator, Linus Torvalds, without OOP in pure C language, and it was just 100,000 lines of code.

But later, the entire core was rewritten by the developer community in the object-oriented C ++ language. And now the Linux kernel exceeds 18,000,000 lines of code.

As we said in one of our articles:

Programming itself is just 20 words - and everyone can learn it very quickly. And in the first years of training you will learn to express any of your ideas with these 20 words. But when you conceive your IT company, when you plan to program a grandiose team project - OOP will save you from many mistakes.

Now you know what OOP is and most importantly - why it is needed. Come and learn programming with us - it's fun!