Forum:Hack Forums

Thread: OOP programming...

Krash39 - *October 22, 2019, 11:42 pm*
hi, i am relatively new in python, i am was programmer in 1990 with COBOL, RPG....and i dont understand very well OOP.....(i maden recently a python script with 1700 lines ? ? , but i do not aply oop, and i am sure that in oop that script was pretty short...)..

i ask for help in oop training.....class and methods, hierachy and all teory i know, but no to think in oop...

how may i think in oop???
perhaps some books that you know on it???

thank you...

ArviX - *October 23, 2019, 3:09 am*
The problem is that you know what oop is? What is it used for?What is ease?

vorticalbox - *October 23, 2019, 4:53 am*
OOP is just a way of programming that allows you to build classes from other classes and its not the only way! personally, I much prefer function programming where you break each part of the program into little functions that only do one task and then join the function to create larger ones.

Krash39 - *October 23, 2019, 11:38 am*
yes, i know that oop is object oriented programming, but i read in all books that python is an oop language and it is better program in oop thinking....for that reason, and because i perhaps want to learn oop method,

Goodies - *October 23, 2019, 3:45 pm*
(Oct 23, 2019 11:38 AM)Krash39 Wrote: *[CITE]* yes, i know that oop is object oriented programming, but i read in all books that python is an oop language and it is better program in oop thinking....for that reason, and because i perhaps want to learn oop method,
OOP means a lot of things, but in general it means that you have a template of a group of related functions (i.e. a class) which can be instantiated and subsequently called with a shared state. If you've ever used "class" in python, then congrats, you've done OOP.

The shared state is accessible through the "self" variable in all class methods. This is the essence of OOP.

MavenGradle - *October 23, 2019, 10:25 pm*
This man is used to programming in a very different style than you all are. He sure as hell gets the concept of OOP but fails to see the advantages or implementations of it. He is used to stateless/functional programming and punch-card like code. If you've programmed like that for the past 30 years it's normal to be confused.

Anyway OP, the best way to break this cycle is to understand the key diffences between the two types. This website explains those differences in a well formed way that will make sense to you:

https://www.educba.com/functional-programming-vs-oop/

Also, OP, are you a fan of RPG IV and free format code or is that not something you did either? If you did, you might find it easier to try Java instead to get to know OOP (and I'm not suggesting this because my name has a link to Java but simply because ILE RPG looks like Java and so the transition can be smoother).

Krash39 - *October 23, 2019, 11:39 pm*

(Oct 23, 2019 04:53 AM)vorticalbox Wrote: *[CITE]* OOP is just a way of programming that allows you to build classes from other classes and its not the only way! personally, I much prefer function programming where you break each part of the program into little functions that only do one task and then join the function to create larger ones.

(Oct 23, 2019 10:25 PM)MavenGradle Wrote: *[CITE]* This man is used to programming in a very different style than you all are. He sure as hell gets the concept of OOP but fails to see the advantages or implementations of it. He is used to stateless/functional programming and punch-card like code. If you've programmed like that for the past 30 years it's normal to be confused.

Anyway OP, the best way to break this cycle is to understand the key diffences between the two types. This website explains those differences in a well formed way that will make sense to you:

Also, OP, are you a fan of RPG IV and free format code or is that not something you did either? If you did, you might find it easier to try Java instead to get to know OOP (and I'm not suggesting this because my name has a link to Java but simply because ILE RPG looks like Java and so the transition can be smoother).

Thank God finally someone understands my situation.

At that time, everything was different.

When I went to work for that company, it was changing the system of punch cards with a system of 3 hard drives that was larger than a closet. I remember that there was an operator in the mainframe room and 3 young ladies recording data. The system had large magnetic tapes to make backup copies.

process thinking was totally different. Now, everything is at the moment. I remember that the systems that were there were chained processes that did chained things and were long processes, for example, the billing that the girls recorded passed them a series of chained programs (which lasted all afternoon and ended at night) in which all tasks were performed.

I remember that the operator went to work when we (the programmers) left and he stayed until the process (processes) ended. If the chained process failed (and failed many times), my boss had to go, and the next day he told us what had happened and how he could solve it urgently ...

Well, I'm rolling up, I'm sorry. What I mean by all this, is that the mentality that was at that time was not oriented at all to the thinking of today. That's why it's hard for me to think like now.

I was the one who started a little to do things that the user could do from the screens of the different departments of the company (modify things, etc ...)

and it still costs me, because all those languages are outdated because the processes are no longer done that way.

Thank you very much and forgive for all the roll that I have told you.

datguy.dev - *October 24, 2019, 10:49 am*

When an author writes a book, they can tell you exactly where events happen in it because they wrote it. However, given the readers are involved, the author should use means that enable the

accessibility of their readers instead. You should use OOP just as books have glossaries and indexes.

MavenGradle - *October 24, 2019, 10:56 am*

(Oct 23, 2019 11:39 PM)Krash39 Wrote: [CITE] Thank God finally someone understands my situation.
At that time, everything was different.
When I went to work for that company, it was changing the system of punch cards with a system of 3 hard drives that was larger than a closet. I remember that there was an operator in the mainframe room and 3 young ladies recording data. The system had large magnetic tapes to make backup copies.
process thinking was totally different. Now, everything is at the moment. I remember that the systems that were there were chained processes that did chained things and were long processes, for example, the billing that the girls recorded passed them a series of chained programs (which lasted all afternoon and ended at night) in which all tasks were performed.
I remember that the operator went to work when we (the programmers) left and he stayed until the process (processes) ended. If the chained process failed (and failed many times), my boss had to go, and the next day he told us what had happened and how he could solve it urgently ...

Well, I'm rolling up, I'm sorry. What I mean by all this, is that the mentality that was at that time was not oriented at all to the thinking of today. That's why it's hard for me to think like now.
I was the one who started a little to do things that the user could do from the screens of the different departments of the company (modify things, etc ...)

and it still costs me, because all those languages are outdated because the processes are no longer done that way.

Thank you very much and forgive for all the roll that I have told you.
No need to tell me buddy. I work with operators all day at work (I sit next to one at work). These kinds of things still exist in this world, day and age. Your skills are still very much required and looked after. In fact, you could probably make more money doing what you did 30 years ago than learning python. Just saying, if you're primary goal is to make money, IBM I Series / AS400 is still a thing.

To get back on topic, let us know if you have any further questions or require assistance. We're here to help.

(Oct 24, 2019 10:49 AM)datguy.dev Wrote: [CITE] When an author writes a book, they can tell you exactly where events happen in it because they wrote it. However, given the readers are involved, the author should use means that enable the accessibility of their readers instead. You should use OOP just as books have glossaries and indexes.
What the fuck does this have to do with the question OP asked? As far I know, the glossaries and indexes of books aren't polymorphic, abstract or encapsulated in nature. Can you even elaborate on this or are you just trolling?

datguy.dev - *October 24, 2019, 3:59 pm*

(Oct 24, 2019 10:56 AM)MavenGradle Wrote: [CITE] What the fugg does this have to do with the question OP asked? As far I know, the glossaries and indexes of books aren't polymorphic, abstract or encapsulated in nature. Can you even elaborate on this or are you just trolling?
Are you a fan of Halt and Catch Fire on AMC? There is this genius coder in that series that is stubborn about her ways. The methods she uses in her code were not standard and that caused serious issues down the road when it came to group work because no one else could follow her work.

The OP is having trouble understanding why OOP is a standard, just like comments. I thought that a simile would explain the reasoning easier that classes are foremost about organization and readability. Without glossaries, readers cannot effectively find content within the book. You see a class in some code, and you can backtrace it to its creation to figure out what is going on. That is just one aspect.

MavenGradle - *October 24, 2019, 4:18 pm*

(Oct 24, 2019 03:59 PM)datguy.dev Wrote: [CITE] Are you a fan of Halt and Catch Fire on AMC? There is this genius coder in that series that is stubborn about her ways. The methods she uses in her code were not standard and that caused serious issues down the road when it came to group work because no one else could follow her work.

The OP is having trouble understanding why OOP is a standard, just like comments. I thought that a simile would explain the reasoning easier that classes are foremost about organization and readability. Without glossaries, readers cannot effectively find content within the book. You see a class in some code, and you can backtrace it to its creation to figure out what is going on. That is just one aspect.
I see now. Truth be told every idiot can write complicated code. It's called spaghetti code that doesn't follow any of the design patters or standard feature sets that should be applied. Not only are you probably making things more complicated than they should be, you're also reducing maintainability not only for others but also for yourself which in turn can cause huge sink-holes in projects that might even cause some projects to fail entirely. All because of a 'genuis' idiot who believed he/she had something to prove.

Don't get me wrong, you see this everywhere, but any code quality control unit should quickly deny any such code in a code base and quite likely get the programmer in question fired if the offence is repeated. This is why tech leads exist. KISS. Keep it simple, stupid. Only shitty programmers who want to prove something write spaghetti code (or the ones who simply don't know any better).

This is in part what separates good programmers, from excellent ones.

datguy.dev - *October 24, 2019, 4:29 pm*

(Oct 24, 2019 04:18 PM)MavenGradle Wrote: [CITE] I see now. Truth be told every idiot can write complicated code. It's called spaghetti code that doesn't follow any of the design patters or standard feature sets that should be applied. Not only are you probably making things more complicated than they should be, you're also reducing maintainability not only for others but also for yourself which in turn can cause huge sink-holes in projects that might even cause some projects to fail entirely. All because of a 'genuis' idiot who believed he/she had something to prove.

Don't get me wrong, you see this everywhere, but any code quality control unit should quickly deny any such code in a code base and quite likely get the programmer in question fired if the offence is repeated. This is why tech leads exist. KISS. Keep it simple, stupid. Only shitty programmers who want to prove something write spaghetti code (or the ones who simply don't know any better).

This is in part what separates good programmers, from excellent ones.
I should note that she is a terminator now ... That is just what happens you code poorly. ?

...

Krash39 - *October 25, 2019, 12:00 am*

Thank you both for your thoughts. Perhaps this is already more than anything else part of programming philosophy than of OOP in question. I guess the "spaghetti" programming also existed 30 years ago. because it was also done, and I think it was even more noticeable than now, since modern programming languages give many more facilities to order the code than before.
Of course, if you want to mess it up you will mess it up and you will get bad spaghetti code. Maybe that's my problem, I don't know. My experience as a programmer was not very long (only 5 years and then I did not practice as a programmer again) and maybe I did not have the bases to write ordered code and write (because my training at that time was scarce) spaghetti code. That is why I may simply lack guidance for orderly programming.
I think that currently, when programmers are trained, they are printed at the same time that they are taught the language in question, a way to program to program in a style. I am increasingly convinced that I lack time and experience to (if I had continued working as a programmer) take good ways to program.

Pd. Really think you that i can do money with my low experience over 30 years ago??? ? ? ? ?

i am 50 years old now, and another life. i only want to learn how to do good code for hobby...and i read in all python books
presuming that it is an oop language in all of them....

nOrr - *October 25, 2019, 2:12 pm*
Python is not an OOP language. It is multi-paradigm and both supports and encourages other styles than OOP.
PS: To say that OOP is just about writing a class is the same as saying any object with four wheels and a seat is a race car.

datguy.dev - *October 26, 2019, 8:05 pm*
(Oct 25, 2019 12:00 AM)Krash39 Wrote: *[CITE]* Thank you both for your thoughts. Perhaps this is already more than anything else part of programming philosophy than of OOP in question. I guess the "spaghetti" programming also existed 30 years ago. because it was also done, and I think it was even more noticeable than now, since modern programming languages give many more facilities to order the code than before.
Of course, if you want to mess it up you will mess it up and you will get bad spaghetti code. Maybe that's my problem, I don't know. My experience as a programmer was not very long (only 5 years and then I did not practice as a programmer again) and maybe I did not have the bases to write ordered code and write (because my training at that time was scarce) spaghetti code. That is why I may simply lack guidance for orderly programming.
I think that currently, when programmers are trained, they are printed at the same time that they are taught the language in question, a way to program to program in a style. I am increasingly convinced that I lack time and experience to (if I had continued working as a programmer) take good ways to program.

Pd. Really think you that i can do money with my low experience over 30 years ago??? ? ? ? ?

i am 50 years old now, and another life. i only want to learn how to do good code for hobby...and i read in all python books
presuming that it is an oop language in all of them....
It's never too late to learn. Everyone has their own means of that. I'm a strong believer in a structured learning system. Learn the entire principles and take strides as in stepping stones. When I went to community college, (in my 20s) there were plenty of people in 40s-50s in my introductory courses and it wasn't abnormal at all. Actually, they were all around better students and easier to talk to. If anything, the younger crowd respected their drive. Maybe you can find some sort of course and get out there? Doesn't have to be college. Maybe an online coding club? Discord Python or

something.