Poznan University of Technology
Objects and classes

Jakub Piotr Hamerliński

Objects and classes

Agenda

01. Object
02. Class
03. Naming conventions
04. Styling
05. Task

> *"There's a lot of beauty in ordinary things. Isn't that kind of the point?"* Pam Halpert

Object

“An object is the equivalent of the quanta from which the universe is constructed”

David West

“They are living organisms, with their own behavior, properties and a life cycle.”

Yegor Bugayenko

Objects and classes

Object example

```
Dog loyalFriend;
loyalFriend.breed = "Akita Inu";
loyalFriend.name = "Hachikō";
```

Objects and classes

Object - Responsibility

A responsibility is a service that an object must provide.
An object is everything capable to provide a limited set of services.
The only way to create an application is to compose objects.
The responsibility of an object is known also as the interface that the object implements.
Object does only what it is intended to do.

Class

Objects and classes

Class

Classes are the units of understanding.
We define the world in terms of objects associated to some class.
Classes define objects of its kind.

Objects and classes

Class example

```
class Dog {
public:
  string breed;
  string name;
  string favSnack;
  int age;
};
```

Naming is hard

Objects and classes

Naming conventions

Manager. Controller. Helper. Handler. Writer. Reader. Converter. Validator. Router. Dispatcher. Observer. Listener. Sorter. Encoder. Decoder. This is the hall of shame for the class names. Do they appear in your code? You use open source libraries, right? In books on patterns? They're all off base. What features do they share? All of them end in "-er." What is wrong with it, then? They are not classes, and neither are the objects they create. Instead, they are simply groups of processes masquerading as classes.

Objects and classes

Naming conventions

There are several great articles, which show that "-er" ending is just bad.

01. Don’t Create Objects That End With -ER by Yegor Bugayenko
02. Your Coding Conventions Are Hurting You by Carlo Pescio
03. One of the Best Bits of Programming Advice I Ever Got by Travis Griggs

Objects and classes

Naming conventions example

Consider that you are my client and that I am an object. You hand me a bunch of numbers and instruct me to sort them. If I were a resident of the world of imperative programming, you would resolve them right away, and we would never speak again. Without even considering why you need them resolved, I will complete my task exactly as required. I would be a sorter who is unconcerned with your true intent:

```
std::list<int> outOfOrderNumbers = {7, 5, 16, 8};
Sorter sorter;
std::list<int> sortedNumbers = sorter.sort(outOfOrderNumbers);
int smallestNumber = sortedNumbers.front();
```

Objects and classes

Naming conventions example

The goal is to find the least number, as you can see above. This is not what you would anticipate from a reliable business associate who can assist you in juggling a lot of figures.

Instead, I would say to you, "Consider them sorted; what do you want to do next?," if I were a declarative programmer. You would then inform me that you urgently require the largest and smallest number. I would then respond, "No problem; here it is." I wouldn’t sort them all in order to get the smallest one back. Just going through one by one, I would choose the largest. Compared to sorting first and then choosing the top item on the list, this process is much faster.

Objects and classes

Naming conventions example

In other words, I would endeavor to carry out my own business without explicitly disobeying your directions. Compared to that necessary sorter, I would make a far more intelligent mate for you. And instead of being a process that sorts, I would turn into an actual thing that functions like a list of apples that has been sorted:

```
std::list<int> outOfOrderNumbers = {7, 5, 16, 8};
SortedNumbers sorted(outOfOrderNumbers);
int smallestNumber = sorted.value(0);
```

Objects and classes

Naming conventions example

Pay close attention to how the sorter and sorted names differ from one another.

Returning to class names now. Your class name instantly becomes a dumb imperative executor of your will when you add the "-er" suffix. You don’t give it any room to reason or improvise. As you sort, manage, control, print, write, combine, concatenate, etc., you want it to do exactly what you want.

A living entity that resists being instructed what to do is an object. By revealing behavior in accordance with its contract(s), also known as interfaces in Java and C# or protocols in Swift, it aspires to be an equal partner with other objects.

Philosophically speaking, the “-er” suffix is a sign of disrespect toward the poor object.

"The name of an object should tell us what this object is, not what it does, just like we name objects in real life: book instead of page aggregator, cup instead of water holder, T-shirt instead of body dresser."

Yegor Bugayenko

Styling

Objects and classes

Styling

Great place to start is Google C++ Style Guide

Tasks

Objects and classes

Tasks

Tasks

01. Write a program in C++ with few classes. Try to make them present the world around you.
02. Write a program in C++ using classes from exercise 1 which will demonstrate your ability to create objects.

Thank you

Feel free to reach me via LinkedIn

Fin