

Кажыкенова Б.К.
Карагандинский университет имени академика Е.А. Бокетова
магистрант 2 курса математического факультета
(Учитель информатики Назарбаев интеллектуальной школы физико-
математического направления г.Нур-Султан)
bahitgul88@gmail.com

ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ – ОДНА ИЗ ПАРАДИГМ ПРОГРАММИРОВАНИЯ, ОСНОВНЫХ НА ТЕХНОЛОГИИ ООП

ООП (Объектно-Ориентированное Программирование) стало неотъемлемой частью разработки многих современных проектов, но, не смотря на популярность, эта парадигма является далеко не единственной. *Объектно-ориентированное программирование* – это подход, при котором вся программа рассматривается как набор взаимодействующих друг с другом объектов. При этом нам важно знать их характеристики. У каждого объекта в системе есть свойства и поведение, как и у любого реального объекта.

(ООП) Объектно-ориентированное программирование—методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования [1].

Прежде всего стоит ответить, зачем? Объектно-ориентированная идеология разрабатывалась как попытка связать поведение сущности с её данными и спроектировать объекты реального мира и бизнес-процессов в программный код. Задумывалось, что такой код проще читать и понимать человеком, т. к. людям свойственно воспринимать окружающий мир как множество взаимодействующих между собой объектов, поддающихся определенной классификации. Удалось ли идеологам достичь цели, однозначно ответить сложно, но де-факто мы имеем массу проектов, в которых с программиста будут требовать ООП.

Как уже говорилось, в настоящее время объектно-ориентированное программирование является магистральным путем развития в индустрии информационных технологий.

Апологеты ООП говорят прежде всего о следующих достоинствах объектно-ориентированных систем: - возможности повторного использования; - упрощении создания сложных систем за счет использования абстракции; - удобстве сопровождения и модификации объектно-ориентированных программ.

В то же время у ООП есть и ряд довольно авторитетных критиков. К ним, в частности, относится такой известный в мире программирования человек, как Никлаус Вирт, создатель языков Паскаль, Модула и Оберон. Он утверждает, в частности, что ООП — не более чем тривиальная надстройка над структурным программированием и преувеличение его значимости, выражющееся в том числе во включении в языки программирования всех новых модных «объектно-ориентированных» средств, безусловно, вредит качеству разрабатываемого программного обеспечения. Н. Вирт очень удивлен тем вниманием, которое уделяется ныне ООП. Другой критик ООП — известный специалист по программированию Александр Степанов, который участвовал в создании C++ вместе с Бьерном Страуструпом, а впоследствии написал стандартную библиотеку шаблонов языка программирования C++ (STL). А. Степанов полностью разочаровался в парадигме ООП. Он пишет: «Я уверен, что ООП методологически неверна. Она начинает с построения классов. Это как если бы математики начинали бы с аксиом. Но реально никто не начинает с аксиом, все начинают с доказательств. Только когда найден набор подходящих доказательств, лишь тогда на этой основе выводится аксиома. То есть в математике вы заканчиваете аксиомой. То же самое и с программированием: сначала вы должны начинать развивать алгоритмы, и только в конце этой работы приходите к тому, что вы в состоянии сформулировать четкие и непротиворечивые интерфейсы. Именно из-за этой неразберихи в ООП так популярен рефакторинг — из-за ущербности парадигмы вы просто обречены на переписывание программы уже в тот самый момент, когда только задумали ее спроектировать в ООП-стиле». Патриарх свободно

распространяемого ПО Ричард Столлмэн также известен своим критическим отношением к ООП. Особенно любит он шутить насчет мифа о том, что ООП «ускоряет разработку программ»: «Как только ты сказал слово “объект”, можешь сразу забыть о 158 модульности» [2].

Критики, в частности, обращают внимание на следующие обстоятельства:

- Внедрение ООП требует дополнительных затрат на обучение программистов и накладных расходов при создании программ — от 12 до 75 % времени.
- Наследование — самая большая провокация в индустрии, ни в каком моделировании наследования не существует.
- θ Инкапсуляция и повторное использование возможны и без ООП при правильной модульной организации программной системы.

В то же время ряд наиболее сложных и успешно используемых программных систем созданы именно в рамках объектноориентированного подхода, его применяют такие гиганты индустрии, как Microsoft, IBM, Oracle [3].

Закрепляя базовые знания принципов объектно-ориентированного программирования можно научится следующему:

- Применять на практике принципы объектно-ориентированного программирования: абстракцию, инкапсуляцию, наследование, полиморфизм.
- Пользоваться классами, объектами, конструкторами, виртуальными функциями и многим другим.
- Представлять, что происходит в оперативной памяти компьютера в процессе выполнения программы.
- Писать программы на языке C#.
- Применять знания при работе с любым другим языком программирования.

Список использованной литературы:

1. Мартин Р. Часть II. Парадигмы программирования. // Чистая архитектура = Clean Architecture. — СПб: Питер, 2018. — ISBN 978-5-4461-0772-8.
2. Майер Б. Объектно-ориентированное конструирование программных систем. — М.: Русская редакция, 2005. — ISBN 5-7502-0255-0.
3. Усов Т. М. Введение в объектно-ориентированное программирование с примерами на C#. — 2019. — ISBN 978-5-600-02430-4.