

ПЕДАГОГИЧЕСКИЕ
ИССЛЕДОВАНИЯ

EDUCATIONAL AND
PEDAGOGICAL STUDIES

УДК 378.147

ОБУЧЕНИЕ
РАЗЛИЧНЫМ ПОДХОДАМ РЕШЕНИЯ
ЭКОНОМИЧЕСКОЙ ЗАДАЧИ
ПРИ ПРОГРАММИРОВАНИИ НА C++
В ВЫСШЕЙ ШКОЛЕ

Аглямзянова Г.Н., Гумерова Л.З.

В данной статье представлены два подхода к обучению решению одной экономической задачи: структурное программирование и объектно-ориентированное программирование.

Ключевые слова: объектно-ориентированное программирование; классы; объекты; методы.

TRAINING IN VARIOUS
APPROACHES TO SOLVING ECONOMIC PROBLEMS
IN C++PROGRAMMING IN HIGH SCHOOL

Aglyamzyanova G.N., Gumerova L.Z.

This article presents two approaches to learning to solve one economic problem: structural programming and object-oriented programming.

Keywords: object-oriented programming; classes; objects; methods.

Введение

Объектно-ориентированное программирование – особый, специфический способ организации программного кода. Принципы ООП

универсальны и не имеют отношения к какому-то определенному языку программирования. Чтобы проиллюстрировать объектно-ориентированный подход, рассмотрим небольшой пример. Рассмотрим постановку экономической задачи: вкладчик хочет положить определенную сумму на счет в банк под проценты. Необходимо определить сумму, которую он получит через указанный период времени. Нам для решения этой задачи необходимо написать программный код. Но вначале нужно определиться с алгоритмом расчетов. В их основе лежит формула, по которой вычисляется сумма (с учетом начисленных процентов) на банковском счету. Для конкретности предположим, что на депозит ложится сумма в M денежных единиц. Процентная ставка будет составлять величину в n процентов годовых, а денежный вклад размещается на период времени t , выражается в годах. Тогда, по истечении периода времени t , вкладчик снимет с депозита сумму $M \left(1 + \frac{n}{100}\right)^t$.

Для вычисления результата необходимо знать значение трех параметров: вносимую на депозит величину M , годовую ставку процента n и время размещения депозита t . Рассмотрим решение данной задачи без использования объектно-ориентированного программирования (ООП). Код представлен в листинге 1.

Листинг 1.

```
#include<iostream>
using namespace std;
// Функция для вычисления итоговой суммы депозита:
double result(double money, double time, double rate){
// Локальная переменная для записи результата функции:
double res=money;
// Локальная переменная для оператора цикла:
int i;
for (i=1;i<=time;i++){
res=res*(1+rate/100);
}
// Результат функции:
```

```
return res;
}
// Главная функция программы:
int main(){
// Значения переменных (первый вкладчик):
double sidorov_money=100;// депозит
double sidorov_rate=13; // процентная ставка
double sidorov_time=3; // время
// Значения переменных (второй вкладчик):
double petrova_money=90; // депозит
double petrova_rate=18; // процентная ставка
double petrova_time=4; // время
// Итоговая сумма депозита для первого вкладчика:
cout<<"Сидоров: "<<result(sidorov_money, sidorov_time, sidorov
_rate)<<endl;
// Итоговая сумма депозита для второго вкладчика:
cout<<"Петрова: "<<result(petrova_money, petrova_time, petrova_
rate)<<endl;
// Завершение программы:
return 0;
}
```

Результат выполнения данного программного кода выглядит следующим образом:

Сидоров: 144.29

Петрова: 174.49

Далее рассмотрим решение данной задачи с использованием ООП. Создаются объекты фактически так же, как объявляются переменные, только вместо идентификатора типа указывается имя класса. Так, в рассматриваемом примере командами `BankAccount sidorov` и `BankAccount petrova` создаются два объекта класса `BankAccount`: один объект называется `sidorov`, а другой объект называется `petrova`. Но создание объектов, как и в случае с переменными, означает лишь, что под них в памяти выделяется место. Это место нужно чем-то заполнить или, проще говоря, полям созданных объектов нужно присвоить значения. Значения полям

присваиваются так же просто, как и значения локальным переменным: слева от оператора присваивания указывается поле, а справа от оператора присваивания указывается присваиваемое полю значение. Правда, одно формальное отличие все же есть. Поскольку у разных объектов имеются поля с одинаковыми названиями, необходимо как-то различать поля разных объектов. Другими словами, если идет обращение к полю, то нужно указать к полю какого объекта направлено обращение. Для этого используется так называемый «точечный» синтаксис: сначала указывается имя объекта, и затем, через точку, имя поля, то есть в формате объект.поле. В таком же формате выполняется обращение к методам объекта: перед инструкцией вызова метода указывается имя объекта.

Имя объекта и имя метода разделяются точкой. Например, командой `sidorov.money=100` полю `money` объекта `sidorov` присваивается значение 100, а командой `petrova.money=90` полю `money` объекта `petrova` присваивается значение 90. Аналогично, для вызова метода `result()`из объекта `sidorov` используем инструкцию `sidorov.result()`, а для вызова метода `result()`из объекта `petrova` используем команду `petrova.result()`.

После того, как полям объектов `sidorov` и `petrova` присвоены значения, для каждого из этих объектов вычисляется итоговая сумма депозита. Для этого достаточно вызвать из соответствующего объекта метод `result()`.

Листинг 2.

```
#include <iostream>
using namespace std;
// Начало описания класса:
class BankAccount {
// Открытые члены класса:
public:
double money;
double rate;
int time;
// Метод для вычисления итоговой суммы депозита:
double result() {
// Локальная переменная для записи результата метода:
```

```
double res=money;
int i;
for (i=1;i<=time;i++){
res=res*(1+rate/100);
}
return res;
}; // Окончание описания класса
// Главная функция программы:
int main(){
BankAccount sidorov;
BankAccount petrova;
// Значения полей первого объекта:
sidorov.money=100;
sidorov.rate=13;
sidorov.time=3;
// Значения полей второго объекта:
petrova.money=90;
petrova.rate=18;
petrova.time=4;
// Итоговая сумма депозита для Сидорова:
cout<<"Сидоров: "<<sidorov.result()<<endl;
// Итоговая сумма депозита для Петровой:
cout<<"Петрова: "<<petrova.result()<<endl;
return 0;
}
```

Результат выполнения программы будет таким:

Сидоров: 144.29

Петрова: 174.49

Результаты в обоих случаях совпадают. Однако программы принципиально разные. Во первом случае совершенно не используются классы и объекты. Вместо этого описана функция `result ()` с тремя аргументами (начальный вклад, процентная ставка и время размещения вклада). Результатом функция возвращает значение итоговой суммы депозита.

В главной функции программы для каждого из вкладчиков определяется по три переменных. Эти переменные передаются аргументами функции `result()`.

Может показаться, что программа без классов и объектов проще и понятней. Но тут важно осознать, что при усложнении задачи, например, при увеличении количества вкладчиков, проявится гибкость и эффективность методов ООП.

Список литературы

1. Царев Р.Ю. Программирование на языке Си: учебное пособие / Р.Ю. Царев. Красноярск: СФУ, 2014. 108 с. – ISBN 978-5-7638-3006-4. – URL: <http://www.studentlibrary.ru/book/ISBN9785763830064.html> (дата обращения: 14.07.2020). Текст: электронный.
2. Ашарина И.В. Язык С++ и объектно-ориентированное программирование в С++. Лабораторный практикум: учебное пособие для вузов / Ж.Ф. Крупская; И.В. Ашарина. Москва: Горячая линия. Телеком, 2016. 232 с. – ISBN 978-5-9912-0464-4. – URL: <http://www.studentlibrary.ru/book/ISBN9785991204644.html> (дата обращения: 14.07.2020). Текст: электронный.
3. Солдатенко И.С. Практическое введение в язык программирования Си: учебное пособие / И.С. Солдатенко, И.В. Попов. Санкт-Петербург: Лань, 2018. 132 с. – ISBN 978-5-8114-3150-2. – URL: <https://e.lanbook.com/book/109619> (дата обращения: 14.07.2020). Текст: эл.

References

1. Carev R.Ju. Programmyrovanye na jazyyke Sy: uchebnoe posobyе / R.Ju. Carev. Krasnojarsk: SFU, 2014. 108 s. – ISBN 978-5-7638-3006-4. – URL: <http://www.studentlibrary.ru/book/ISBN9785763830064.html>
2. Asharyna Y.V. Jazyyk S++ y ob#ektно-oryentyrovannoe programmyrovanye v S++. Laboratornyyj praktikum: uchebnoe posobyе dlja vuzov / Zh.F. Krupskaja; Y.V. Asharyna. Moskva: Gorjachaja lynuja. Telekom, 2016. 232 s. – ISBN 978-5-9912-0464-4. – URL: <http://www.studentlibrary.ru/book/ISBN9785991204644.html>
3. Soldatenko Y.S. Prakticheskoе vvedenye v jazyyk programmyrovanyja Sy: uchebnoe posobyе / Y.S. Soldatenko, Y.V. Popov. Sankt-Peterburg:

Lan', 2018. 132 s. – ISBN 978-5-8114-3150-2. – URL: <https://e.lanbook.com/book/109619>

ДААННЫЕ ОБ АВТОРАХ

Аглямзянова Гульшат Накиповна, доцент кафедры информатики и вычислительной математики, кандидат физико-математических наук

Набережночелнинский государственный педагогический университет

*ул. Низаметдинова, 28, г. Набережные Челны, 423806, Россия
dina.airat@mail.ru*

Гумерова Лилия Зуфаровна, доцент кафедры системного анализа и информатики, кандидат педагогических наук

Казанский (Приволжский) федеральный университет

*ул. Кремлёвская, 18, г. Казань, 420008, Россия
gum9370@mail.ru*

DATA ABOUT THE AUTHORS

Aglyamzyanova Gulshat N., Associate Professor of the department of computer science and computational mathematics, PhD in Physics and Mathematics

Naberezhnye Chelny Pedagogical University

*28, Nizametdinov str., Naberezhnye Chelny, 423806, Russia
dina.airat@mail.ru*

SPIN-code: 9891-7381

ORCID: 0000-0002-8003-5427

Gumerova Liliya Z., Associate Professor of the Department of System Analysis and Informatics, PhD in Pedagogics

Kazan Federal University

*18, Kremlevskaya Str., Kazan, 420008, Russia
gum9370@mail.ru*

SPIN-code: 4343-2445

ORCID: 0000-0002-8865-7589