*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

|  |  |  |  |
| --- | --- | --- | --- |
| Gerb-BMSTU_01 | **«Московский государственный технический университет имени Н.Э. Баумана»** | |  |
| Факультет: | | ИУ «Информатика и системы управления» |
| Кафедра: | | ИУ-6 «Компьютерные системы и сети» |

**Отчет**

**по домашнему заданию № 15**

**«Объекты. Наследование.»**

вариант №4

**Дисциплина: Основы программирования**

|  |  |  |
| --- | --- | --- |
| Студент группы ИУ6-51 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Милков В. В. |
|  | (Подпись, дата) | (Фамилия И.О.) |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гукетлев Д. В. |
|  | (Подпись, дата) | (Фамилия И.О.) |

*Москва, 2016*

**Содержание**

Задание 3

Блок-схема алгоритма 4

[Код программы 5](#h.2et92p0)

[Скриншоты выполнения программы 7](#h.tyjcwt)

[Вывод 8](#h.3dy6vkm)

**Задание**

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных.

Объект – доска. Параметры: два целых числа – длина и ширина, Методы: инициализирующий, определения площади и возвращающие значения.

Объект – ящик. Параметры: длина, ширина и высота. Методы: инициализирующий, определения объема и вывода параметров.

**UML-схема классов**



**Код программы**

**program main;**

**{$mode objfpc}{$m+}**

**uses crt;**

**type**

**PlankClass = class**

**private**

**length, width: integer;**

**public**

**constructor create(l, w: integer);**

**function getArea(): integer;**

**end;**

**constructor PlankClass.create(l,w:integer);**

**begin**

**length:=l;**

**width:=w;**

**end;**

**function PlankClass.getArea() : integer;**

**begin**

**result := width \* length;**

**end;**

**type**

**BoxClass = class(PlankClass)**

**private**

**height: integer;**

**public**

**constructor createBox(l, w, h: integer);**

**function getVolume(): integer;**

**end;**

**constructor BoxClass.createBox(l, w, h:integer);**

**begin**

**length:=l;**

**width:=w;**

**height:=h;**

**end;**

**function BoxClass.getVolume() : integer;**

**begin**

**result := getArea() \* height;**

**end;**

**var**

**plank: PlankClass;**

**box: BoxClass;**

**begin**

**plank := PlankClass.create(10, 20);**

**box := BoxClass.createBox(10, 20, 30);**

**writeln('Area of the plank: ', plank.getArea());**

**writeln('Volume of the box: ', box.getVolume());**

**end.**

**Скриншоты выполнения программы**



**Вывод**

Наследование позволяет использовать библиотеки классов и развивать их (совершенствовать и модифицировать биб­лиотечные классы) в конкретной программе. Наследование позволяет создавать новые объекты, из­меняя или дополняя свойства прежних. Объект-наследник полу­чает все поля и методы предка, но может добавить собст­венные поля, добавить собственные методы или перекрыть своими методами одноименные унаследованные методы.

Принцип наследования решает проблему модификации свойств объекта и придает ООП в целом исключительную гибкость. При работе с объектами программист обычно подбирает объект, наиболее близкий по своим свойствам для решения конкретной задачи, и создает одного или нескольких потомков от него, которые «умеют» делать то, что не реализовано в родителе.