

Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

Ю.С. Белов, И.И. Ерохин

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ
НА PYTHON**

Методические указания к лабораторной работе
по дисциплине «Высокоуровневое программирование»

Калуга – 2019

УДК 004.62
ББК 32.972.1
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационные технологии».

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № ____ от «__» _____ 2019 г.

Зав. кафедрой ИУ4-КФ _____ к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ФНК протокол № ____ от «__» _____ 2019 г.

Председатель методической комиссии факультета ИУ-КФ _____ к.т.н., доцент М.Ю. Адкин

- Методической комиссией КФ МГТУ им.Н.Э. Баумана протокол № ____ от «__» _____ 2019 г.

Председатель методической комиссии КФ МГТУ им.Н.Э. Баумана _____ д.э.н., профессор О.Л. Перерва

Рецензент:

зав. кафедрой ИУ2-КФ _____ И.В. Чухраев
«Информационные системы и сети»,
к.т.н., доцент

Авторы
к.ф.-м.н., доцент кафедры ИУ4-КФ _____ Ю.С. Белов
асс. кафедры ИУ4-КФ _____ И.И.Ерохин

Аннотация

Методические указания к выполнению лабораторной работы по курсу «Высокоуровневое программирование» содержат общие сведения о работе с классами, объектами, методами, основными принципами объектно-ориентированного программирования и средствами их реализации на языке Python.

Предназначены для студентов 2-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2019 г.
© Ю.С. Белов, И.И. Ерохин, 2019 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
ЗАДАЧИ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	15
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ.....	21
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ.....	41
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ.....	42
ОСНОВНАЯ ЛИТЕРАТУРА.....	43
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	43

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Высокоуровневое программирование» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 2-го курса направления подготовки 09.03.04 «Программная инженерия», содержат сведения о классах, объектах, методах, основных принципах объектно-ориентированного программирования и средствах их реализации на языке Python.

Методические указания составлены для ознакомления студентов с основополагающими понятиями и принципами разработки программ, решение которых предполагает использование объектно-ориентированного подхода.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков процедурного программирования, разработки и отладки программ, овладение методами и средствами разработки и оформления технической документации.

Основными задачами выполнения лабораторной работы являются:

1. Изучить особенности создания классов;
2. Научиться создавать экземпляры классов;
3. Изучить типовые алгоритмы решения задач с использованием принципов объектно-ориентированного программирования.

Результатами работы являются:

1. Реализация разработанных алгоритмов на языке программирования Python;
2. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Объектно-ориентированное программирование (в дальнейшем ООП) — это предназначенная для снижения сложности и повышения надежности методика разработки программ, в которой основными понятиями являются классы и объекты.

В языке Python класс равносителен понятию тип данных. Он описывает группы однородных объектов: их характеристики и возможные действия. Например, тип `int` – это класс целых чисел. Числа 2, 25, 2019, -10 и т. д. – это конкретные объекты этого класса.

Python поддерживает создание классов и работу с ними в одной рабочей вкладке. Для объявления класса в Python используется ключевое слово `class`, после которого указывается идентификатор (имя класса) и ставится двоеточие. С новой строки начинается описание тела класса, в котором перечисляются атрибуты (переменные) и указываются методы. Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков. Рекомендуется использовать отступы, равные величине 4 пробелов или табуляции.

Переменные, определённые внутри класса (не метода), называются переменными экземпляра или полями (`fields`).

Метод — это именованный блок кода, который объявляется внутри класса и может быть использован многократно. Для его объявления необходимо использовать зарезервированное слово `def`, после которого указывается название метода и в скобках его параметры. Отметим, что первым параметром каждого метода на языке Python всегда является `self` (ключевое слово, которое относится к данному классу). Оно необходимо, чтобы разграничить поля и параметры.

Приведём пример создания содержащего атрибуты и метод класса:

```

class Robot:
    x = 5    # integer
    y = 5.0  # real
    def forward(self, x, y):
        self.x += x
        self.y += y

```

Отметим, что классы принято называть с заглавной буквы, используя стиль UpperCamelCase, в то время как при именовании атрибутов и методов следует придерживаться стиля lowerCamelCase. В классе могут быть несколько переменных и методов.

Конкретный представитель класса называется объектом. Он наследует свойства и методы класса. В языке программирования Python объекты принято называть также экземплярами. Это связано с тем, что в нем все классы сами являются объектами класса type. Точно также как все модули являются объектами класса module. Поэтому во избежание путаницы объекты, созданные на основе обычных классов, называют экземплярами.

```

#Создаём объект класса Robot
wally = Robot()

```

Конструктор

Кроме обычных методов [классы](#) могут определять специальные методы, которые называются конструкторами. Они вызываются при создании нового объекта данного класса и выполняют инициализацию объекта. В отличие от стандартных методов, конструктор никогда ничего не возвращает.

Если в классе не определено ни одного конструктора, то для этого класса автоматически создается конструктор без параметров. Имя такого метода обычно регламентируется синтаксисом конкретного языка программирования. Так в Java имя конструктора класса совпадает с именем самого класса. В Python же роль конструктора играет метод `__init__()`.

Конструктор определяет действия, выполняемые при создании объекта класса, и является важной частью класса. Как правило, программисты стараются явно указать конструктор. Если явного конструктора нет, то Python автоматически создаст его для использования по умолчанию.

Определенный выше определенный класс Robot не имеет явно определённого конструктора. Поэтому для него автоматически создается конструктор по умолчанию, который мы можем использовать для создания класса объекта Robot.

Наличие пар знаков подчеркивания спереди и сзади в имени метода `__init__` говорит о том, что он принадлежит к группе методов перегрузки операторов. Если подобные методы определены в классе, то объекты могут участвовать в таких операциях как сложение, вычитание, вызываться как функции и др.

Рассмотрим пример использования `__init__()` в явно определённых конструкторах:

```
class Person:
    def __init__(self, name, surname):
        self.name = name
        self.surname = surname
p1 = Person("Иванов", "Иван")
print(p1.name, p1.surname)
```

Результат: Иванов Иван

При этом методы перегрузки операторов не надо вызывать по имени. Вызовом для них является сам факт участия объекта в определенной операции. В случае конструктора класса – это операция создания объекта. Так как объект создается в момент вызова класса по имени, то в этот момент вызывается метод `__init__()`, если он определен в классе.

Деструктор

Помимо конструктора объектов в языках программирования есть обратный ему метод – деструктор. Он вызывается, когда объект не создается, а уничтожается.

В языке программирования Python объект уничтожается, когда исчезают все связанные с ним переменные или им присваивается другое значение, в результате чего связь со старым объектом теряется. Удалить переменную можно с помощью команды языка `del`. В классах Python функцию деструктора выполняет метод `__del__()`. После окончания работы с объектом мы можем использовать оператор **`del`** для удаления его из памяти:

```
person = Person("Tom")
del person # удаление из памяти
```

Основные принципы ООП

Основными принципами структурирования, которые придают объектам новые свойства, в объектно-ориентированных языках являются: абстрагирование, наследование, инкапсуляция и полиморфизм.

Инкапсуляция

Инкапсуляция — ограничение доступа к составляющим объект компонентам (методам и переменным). Инкапсуляция делает некоторые из компонент доступными только внутри класса.

Однако инкапсуляция в Python работает лишь на уровне соглашения между программистами о том, какие атрибуты являются общедоступными, а какие — внутренними. Таким образом, одиночное подчеркивание в начале имени атрибута говорит о том, что переменная или метод не предназначен для использования вне методов класса, однако атрибут доступен по этому имени.

Двойное подчеркивание в начале имени атрибута даёт большую защиту: атрибут становится недоступным по этому имени, то есть мы сможем обратиться к нему только из того же класса, где он определён, но не сможем обратиться к нему вне этого класса.

Приведём пример создания методов с ограниченным доступом:

```
class Car:
    # при вызове данного метода будет ошибка
```

```

def _private(self):
    print("Это приватный метод!")
# при вызове данного метода будет ошибка
def __private(self):
    print("Это приватный метод!")
car = Car()
car._private()
car.__private()

```

Рекомендуется как можно больше ограничивать доступ к данным, чтобы защитить их от нежелательного доступа извне (как для получения значения, так и для его изменения). Использование различных модификаторов гарантирует, что данные не будут искажены или изменены не надлежащим образом.

Наследование - один из четырёх важнейших механизмов объектно-ориентированного программирования, позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса, доступ к которым не был ограничен двойным подчёркиванием, заимствуются новым классом.

Термин “родительский класс” равнозначен таким понятиям, как “базовый класс” и “суперкласс”. Класс-наследник, в свою очередь, можно называть подклассом, производным или дочерним классом.

Чтобы объявить один класс наследником от другого, необходимо при создании нового класса после имени указать в круглых скобках название суперкласса, от которого будут наследоваться поля и методы:

```

class Car (Vehicle):
    # методы и поля

```

Концепция наследования в программировании позволяет сократить время разработки, упростить процесс написания программы как сейчас, так и в будущем, когда заказчик захочет внести "небольшие правки" в проект. Правильно спроектированный класс это, кроме

прочего, гибкая структура, которую вы свободно сможете изменять, дополнять в будущем.

Полиморфизм

Полиморфизм в объектно-ориентированном программировании реализует возможность обработки данных, принадлежащих к разным классам, с помощью функции или метода, которые были унаследованы от базового класса. Результат работы одноименных методов может существенно различаться, поэтому в данном контексте под полиморфизмом понимается множество форм одного и того же слова – имени метода.

Когда несколько классов имеют одни и те же имена методов, но различные реализации для этих же методов, классы являются полиморфными. Приведём пример:

```
class Vehicle():
    def printType(self):
        print("Наземный вид транспорта")
    def printName(self):
        print("Транспортное средство")
class Car(Vehicle):
    def printName(self):
        print("Наземный вид транспорта")
    def printType(self):
        print("Автомобиль")
class Train(Vehicle):
    def printName(self):
        print("Наземный вид транспорта")
    def printType(self):
        print("Поезд")
car = Car()
car.printName()
car.printType()
train = Train()
train.printName()
train.printType()
```

Результат работы: Наземный вид транспорта
Автомобиль

Наземный вид транспорта

Поезд

Классы `Car` и `Train` являются наследниками от родительского класса `Vehicle`. Переопределив унаследованный метод `printName`, получилось, что два производных класса содержат метод с одним и тем же названием, но различным результатом работы. Это стало возможным благодаря полиморфизму. Таким образом, полиморфизм позволяет нам изменять функции в классах наследниках и переопределять методы.

Абстрагирование

Абстрагированием называется процесс выделения абстракций в предметной области задачи. Абстракция - это совокупность существенных характеристик некоторого объекта, которые отличают его от всех других видов объектов и, таким образом, четко определяют особенности данного объекта с точки зрения дальнейшего рассмотрения и анализа. В соответствии с определением применяемая абстракция реального предмета существенно зависит от решаемой задачи: в одном случае нас будет интересовать форма предмета, в другом вес, в третьем - материалы, из которых он сделан, в четвертом – закон движения предмета и т.д.

Абстрактным называется класс, который содержит один и более абстрактных методов. Абстрактным называется объявленный, но не реализованный метод. Предполагается, что класс-потомок должен переопределить все методы и реализовать их функциональность.

В Python отсутствует встроенная поддержка абстрактных классов, для этой цели используется модуль `abc` (Abstract Base Class), который включается в стандартную библиотеку `abc`, добавляющую в язык абстрактные базовые классы. Модуль `abc` содержит метакласс (`ABCMeta`) и декораторы (`@abstractmethod` и `@abstractproperty`).

Рассмотрим пример создания абстрактного класса, который содержит один абстрактный и один обычный метод:

```

from abc import ABC, abstractmethod
class Vehicle(ABC):
    # обычный неабстрактный метод
    def reportChanges(self):
        print("Координаты машины были изменены")
    # абстрактный метод
    @abstractmethod
    def move(self):
        pass

```

Производный класс обязан переопределить и реализовать все методы или свойства, объявленные с использованием декораторов `@abstractmethod` и `@abstractproperty`, которые имеются в базовом абстрактном классе. Также следует учитывать, что если класс имеет хотя бы один абстрактный метод, то данный класс должен быть определен как абстрактный.

В приведённом выше примере также используется оператор `pass`, равноценный отсутствию операции. В ходе исполнения данного оператора ничего не происходит, поэтому он может использоваться в качестве заглушки в тех местах, где это синтаксически необходимо, например: в инструкциях, где тело является обязательным, таких как `def`, `except` и другие.

Зачастую `pass` используется там, где код пока ещё не появился, но планируется. Кроме этого, иногда, его используют при отладке, разместив на строчке с ним точку останова.

Композиция

Особенностью объектно-ориентированного программирования является возможность реализовывать композиционный подход, заключающийся в том, что определяется класс-контейнер (агрегатор), который включает в себя вызовы других классов. В результате получается, что при создании объекта класса-контейнера, также создаются объекты включенных в него классов.

```

class WinDoor:
    def __init__(self, x, y):
        self.square = x * y
class Room:
    def __init__(self, x, y, z):

```

```

        self.square = 2 * z * (x + y)
        self.wd = []

    def addWD(self, w, h):
        self.wd.append(WinDoor(w, h))

    def workSurface(self):
        new_square = self.square
        for i in self.wd:
            new_square -= i.square
        return new_square

r1 = Room(6, 3, 2.7)
print(r1.square)    # выведет 48.6
r1.addWD(1, 1)
r1.addWD(1, 1)
r1.addWD(1, 2)
print(r1.workSurface())    # выведет 44.6

```

Таким образом, объект класс WinDoor создаётся при создании объекта класса Room.

ЗАДАЧИ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Рассмотрим выполнение лабораторной работы на примере нескольких задач.

Первая задача: создайте класс `Car`, содержащий метод `location` для изменения расстояния от машины до парковки, метод `printLocation` - для печати расстояния до парковки и метод `speed`, печатающий сообщение о том, что скорость автомобиля = 60 км/ч. Создайте два класса-наследника: `Audi` и `BMW`, для каждого задайте расстояние, на котором они находятся относительно парковки. Для автомобиля `BMW` переопределите метод `speed`, установив скорость = 70 км/ч. Сместите каждый из автомобилей на произвольное расстояние, используя метод `location`. Выведите информацию о скорости и удалённости от парковки каждого автомобиля.

Для начала работы необходимо создать вкладку в новом проекте (в качестве примера используется среда `PyCharm`). Теперь сделаем импорт необходимых библиотек и модулей.

```
from __future__ import print_function
import random
```

Псевдомодуль `__future__` программисты могут использовать для включения новых языковых функций, которые не совместимы с текущим интерпретатором. В дальнейшем он будет необходим для изменения функции вывода сообщения. Модуль `random`, в свою очередь, позволяет генерировать случайные числа.

Теперь опишем поля и методы родительского класса `Car`:

```
class Car:
    loc = 0.0
    def __init__(self, loc):
        self.loc = loc
    def location (self):
        self.loc += (random.randint(30, 60))
```



```
def speed (self):
    print("скорость автомобиля 60 км/ч")

def printLocation(self):
    print(self.loc, end = ' ')
    print("км до парковки")
```

Обратим внимание, что при создании класса был явно указан конструктор, использующий ключевое слово self, с помощью которого были разграничены переменные класса (поля) и переменные метода, имеющие одинаковые названия.

Теперь создадим первый класс-наследник Audi. Данный класс будет наследовать методы и поля родительского класса, не переопределяя их и не добавляя новых. Так как в Python существует синтаксическое требование: блоки кода после if, except, def, class не могут быть пустыми, то используем оператор pass:

```
class Audi(Car):
    pass
```

Создадим второй класс-наследник BMW, который будет переопределять метод speed:

```
class BMW(Car):
    def speed (self):
        print("скорость автомобиля 70 км/ч")
```

Для завершения выполнения задания лабораторной работы создадим объекты классов Audi и BMW с определёнными параметрами и выведем данные о них:

```
car1=Audi(25)
car2=BMW(5)
print("Ауди ", end = ': ') #изменение функции в
команде вывода
car1.printLocation()
```

```

print("BMW", end = ': ') #изменение функции в команде
вывода
car2.printLocation()
print()
print("Состояние авто после перемещения по городу")
car2.location()
car1.location()
print("Ауди ", end = ': ') #изменение функции в
команде вывода
car1.printLocation()
car1.speed()
print("BMW", end = ': ') #изменение функции в команде
вывода
car2.printLocation()
car2.speed()

```

По умолчанию функция print после завершения вывода сообщения переносит позицию курсора для вывода на экран на новую строку, однако иногда необходимо, чтобы несколько последовательно вызываемых методов выводили сообщение на одну и ту же строку. Подключив псевдомодуль `__future__`, поставить запятую после перечисления необходимых переменных и прописать `end = 'символ, которым будет заканчиваться вывод сообщения (\n по умолчанию)'`.

Результат: Марка автомобиля - Audi

Расстояние до парковки равно 49,24

Скорость автомобиля равна 60 км/ч

Марка автомобиля - BMW

Скорость автомобиля равна 60 км/ч

Расстояние до парковки равно 5,00

Скорость автомобиля равна 70 км/ч

Вторая задача: написать программу с использованием абстрактного класса для конвертации валюты. Пользователь должен задать кол-во денег определённой валюты (USD или EU). Программа должна выводить результат конвертации на экран.

Так как по условию необходимо разработать программу с использованием абстрактного класса, то объявим абстрактный класс Valute, определим методы и подключим необходимые модули:

```
from __future__ import print_function
from abc import ABC, abstractmethod
class Valute(ABC):
    @abstractmethod
    def getRub(self):
        pass

    @abstractmethod
    def printData(self):
        pass
```

Создадим класс-наследник Convert, который будет переопределять и реализовывать все унаследованные абстрактные методы.

```
class Convert(Valute):
    Rub=0
    def __init__(self, USD, EU):
        self.USD = USD
        self.EU = EU
    def getRub(self):
        if (self.USD > 0):
            self.Rub = self.Rub + self.USD*65
        if (self.EU > 0):
            self.Rub = (self.Rub + self.EU*75)

    def printData(self):
        print("USD = ", + self.USD)
        print("EU = ", + self.EU)
        print("При конвертации в рубли получим ",
end=' : ')
        print(self.Rub)
```

Таким образом, в классе Convert были реализованы описанные в интерфейсе методы, а также добавлен метод printData(). Для завершения выполнения задания лабораторной работы создадим объект convert класса Convert и вызовем для него последовательно необходимые методы. Стоит отметить, что при создании объекта

необходимо указать количество валюты, которую необходимо перевести в рубли.

```
bag = Convert(3, 7)
bag.getRub()
bag.printData()
```

Результат: USD = 3

EU = 7

При конвертации в рубли получим : 720

Третья задача: написать программу, в которой создаётся массивы объектов таких классов, как: кальмар, дельфин, сом. Количество элементов каждого массива генерируется случайным образом от одного до четырёх.

Вывести на экран для каждого элемента массива тип созданной рыбы и её привычное место обитания (река/море/океан).

Для начала работы необходимо создать новую вкладку в проекте. Подключим необходимые библиотеки и опишем абстрактный класс Fish, который будет являться родительским:

```
import random
from abc import ABC, abstractmethod
class Fish(ABC):
    @abstractmethod
    def printData(self):
        pass
```

Создадим три класса-наследника, каждый из которых будет реализовывать объявленный абстрактный метод:

```
class Som(Fish):
    def printData(self):
        print("Сом, привычное место обитания - река")
class Dolphin(Fish):
    def printData(self):
        print("Дельфин, привычное место обитания -
```

```

море")
class Squid(Fish):
    def printData(self):
        print("Кальмар, привычное место обитания -
океан")

```

Создадим переменные, которые будут отвечать за количество элементов в классах Сом, Дельфин и Кальмар и присвоим им случайное значение от 1 до 4:

```

numS = (random.randint(1, 4))
numD = (random.randint(1, 4))
numSq = (random.randint(1, 4))

```

Для завершения выполнения задания лабораторной работы создадим массив объектов каждого класса и вызовем реализованные ранее методы:

```

print("Всего в заповеднике в заповеднике: ", +
      (numS+numD+numSq), "животных")
som = [Som() for each in range(numS)]
for i in range (0,numS):
    som[i].printData()

dolphin = [Dolphin() for each in range(numD)]
for i in range (0,numD):
    dolphin[i].printData()

squid = [Squid() for each in range(numSq)]
for i in range (0,numD):
    squid[i].printData()

```

Результат: Всего в заповеднике в заповеднике: 6 животных
 Сом, привычное место обитания - река
 Сом, привычное место обитания - река
 Дельфин, привычное место обитания - море
 Дельфин, привычное место обитания - море
 Кальмар, привычное место обитания - океан

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Вариант 1

Задача 1. Создайте класс Robot, содержащий метод movement для изменения координат при движении(x,y), метод printPosition - для печати координат и метод color - для вывода сообщения, о том, что цвет робота белый. Создайте два класса-наследника: Wally и Eva, задайте их координаты. Переопределите метод color для класса Wally, заменив цвет на жёлтый. Сместите каждого робота на произвольное расстояние, используя movement. Выведите информацию о координатах и цвете каждого робота.

Задача 2. Необходимо создать два абстрактных класса, первый из которых описывает какой-либо объект (школьник, студент, аспирант) и реализуется в подклассах, второй - должен называться как суперкласс Pupil и также реализовываться в трёх подклассах.

Задача 3. В компьютерной игре Minecraft присутствуют овцы и прирученные волки. У овец может быть до 8.0 единиц здоровья, а у волков до 20.0. Положение волка или овцы задается трехмерным радиус-вектором от начала координат. У волка присутствует способность атаковать овцу: при атаке у овцы отнимается $(h / h_f) d / r^2$ единиц здоровья, где

- h_f — константа полного здоровья волка,
- h — текущее здоровье волка,
- d — константа урона волка, равная 4-м,
- r — расстояние от волка до овцы.

Необходимо создать модель атаки волка на овцу, позволяющую рассчитать, сколько секунд понадобится волку для убийства овцы при их заданных состояниях. Использовать абстрактный суперкласс.

Задача 4. Создать класс Human и описать в нем имя, пол, возраст и рост человека, при этом возраст и рост можно указывать как в формате целого числа, дробного и также в виде строки (57, пятьдесят семь; 180, 1.80, метр восемьдесят). Описать метод print, выводящий на экран всю информацию о человеке.

Вариант 2

Задача 1. Создайте класс Animals и два класса-наследника Home и Wild. Распределите животных ЛЕВ, МЕДВЕДЬ, СОБАКА, КОШКА по данным классам и добавьте породу/вид. Создайте для каждого класса-наследника метод getBreed для вывода породы/вида животного.

Задача 2. Написать программу с использованием абстрактного класса для перевода скорости. Пользователь должен ввести скорость спутника (предусмотреть ввод в км/ч и м/с). Программа должна выводить исходные данные и их перевод в км/с.

Задача 3. Создать класс Car и описать в нем марку машины, ее номер и скорость. Скорость можно указать как в формате целого числа, так и строки (60, шестьдесят). Описать метод Enter, заполняющий все поля класса с клавиатуры.

Задача 4. Сильная и независимая женщина решила исследовать процесс кормления своих питомцев на компьютерной модели. У каждого питомца есть параметр *сытость*, задаваемый целым числом процентов в диапазоне (0, 100]. При симуляции кормления (метод *.eat()*) модель собаки должна печатать в консоль "Гав!", а модель

кошки "Мяу!". Кормление добавляет к сытости до 75% для любого животного, так, что в итоге сытость оказывается не более 100%. Необходимо реализовать такую модель, используя абстрактные классы, случайным образом сгенерировать массив питомцев и распечатать его состояние до и после кормления всех животных.

Вариант 3

Задача 1. Создайте класс Checkers, содержащий метод movement для изменения координат фигуры при перемещении (x,y), метод printPosition - для печати координат и метод color - для вывода сообщения, о том, что цвет фигуры белый. Создайте класс-наследник Queen, начальные координаты которого (0,0), переопределите метод printPosition таким образом, чтобы при выходе за границы игрового поля выводилось сообщение о невозможности хода. Создайте класс-наследник Pawn (0,0) класса Queen, переопределите метод color, изменив цвет фигуры на чёрный. Сместите фигуры на произвольное расстояние. Выведите информацию о координатах и цвете каждой из фигур

Задача 2. Написать программу с использованием абстрактного класса для вычисления синуса угла, заданного рядом Тейлора.

Задача 3. Создать класс MyWallet и описать в нем имя владельца счёта, номер карты и остаток денег на счёте, при этом остаток денег на счёте можно указывать как в формате дробного числа, так и строки (1000.00, тысяча рублей). Описать метод rest, определяющий, есть ли на счету деньги, и если есть, то сколько.

Задача 4. В очереди за бесплатными макбуками стоят студенты групп ИТД и ЭВМ. У каждого студента есть целочисленный балльный рейтинг, изначально равный нулю. Студенты могут сдавать лабораторные работы по программированию (метод

.makeProgramming()) и по электротехнике (метод *.makeElecEngineering()*). За работу по программированию студенту ЭВМ зачисляется случайным образом 3 или 4 балла, а за электротехнику 4 или 5. Для студентов ИТД наоборот. Необходимо случайным образом сгенерировать модель очереди студентов, провести несколько работ, и определить, сколько студентов ИТД и сколько студентов ЭВМ получают макбук. Макбуков раздается около 25% от количества всех студентов, и получают из лучшие по рейтингу. После каждого существенного шага следует выводить на экран состояния студентов, для реализации модели необходимо использовать абстрактные классы и оператор *instanceof*.

Вариант 4

Задача 1. Создайте класс *Geolocation*, содержащий метод *forward* для изменения координат при движении по оси *Ox* и *Oy*, метод *decline* - для изменения координат по осям *Ox*, *Oy*, *Oz*, метод *printPosition* – для печати координат. Создайте класс-наследник *aircraft* и определите на нём метод *wing*, печатающий сообщение о том, что у самолёта 2 крыла. Создайте класс-наследник *aircraft*, переопределяющий метод *wing*. Сместите самолёт и вертолёт на произвольные расстояния и выведите сообщение о количестве крыльев и координатах летательных аппаратов.

Задача 2. Создать 3 класса (базовый) и 2 предка, которые описывают некоторых работников с почасовой оплатой (один из предков) и фиксированной оплатой (второй предок). Описать в базовом классе абстрактный метод для расчета среднемесячной зарплаты. Для «почасовиков» формула для расчета такая: “среднемесячная зарплата = 20.8 * 8 * ставка в час”, для работников с фиксированной оплатой “среднемесячная зарплата = фиксированной месячной оплате”. Задание: упорядочить всю последовательность рабочих по убыванию среднемесячной зарплаты. При совпадении зарплаты –

упорядочить данные в алфавитном порядке по имени. Вывести идентификатор работника, имя и среднемесячную зарплату для всех элементов списка.

Задача 3. Создать класс Mail и описать в нем имя отправителя, имя получателя, место получения и место отправления, при этом место отправления и получения могут быть заданы индексами(целым числом), либо названием города(строкой). Описать метод travel, определяющий покинет ли письмо пределы города (места получения и отправления отличаются).

Задача 4. Требуется реализовать часть виртуальной машины языка Brainfuck: список из объектов-операций, имеющих метод .execute(), принимающий по ссылке массив памяти и номер активной ячейки. Операции в BF:

1. '>' — перейти к следующей ячейке;
2. '<' — перейти к предыдущей ячейке;
3. '+' — увеличить значение в текущей ячейке на 1;
4. '-' — уменьшить значение в текущей ячейке на 1;
5. '.' — напечатать значение из текущей ячейки,
6. ',' — ввести значение и сохранить в текущей ячейке;
7. '[' — если значение текущей ячейки ноль, перейти вперёд по тексту программы на ячейку, следующую за соответствующей ']' (с учётом вложенности);
8. ']' — если значение текущей ячейки не ноль, перейти назад по тексту программы на символ '[' (с учётом вложенности).

Применить абстрактный класс для задания операции.

Вариант 5

Задача 1. Создайте класс Parent, задайте параметр grow определите метод colorEye, выводящий предложение о зелёном цвете глаз, метод changeGrow – вычисляющий изменение роста, метод printGrow –

печатающий величину роста. Создайте два класса-наследника `Masha` и `Peter`, задайте параметр роста для них. В одном из данных классов переопределите метод `colorEye`, заменив цвет глаз на карий. Задайте параметры изменения роста для имеющихся классов. Выведите сообщение о росте и цвете глаз каждого класса.

Задача 2. Создать абстрактный класс, описывающие возможности игровых персонажей (бег, плавание, прыжки). Создать базовый класс персонажа для игры и два унаследованных класса персонажа, которые используют реализованные методы абстрактного класса для определения своих возможностей.

Задача 3. Создать класс `Holiday` и описать в нем название праздника, имя приглашенного и количество персон в этом приглашении. Описать метод `family`, определяющий может ли вся семья приглашенного (количество задается с клавиатуры) посетить данный праздник.

Задача 4. Для описания всех людей, находящихся в магазине, необходимо выделить подмножество работников и подмножество посетителей. У каждого человека есть полное имя, у сотрудников магазина имеет значение должность, а у посетителей возраст. Необходимо сгенерировать список людей в магазине и вызовом виртуального метода из абстрактного класса напечатать для сотрудников фамилию и должность, а для посетителей имя и возраст.

Вариант 6

Задача 1. Создайте класс `Factory`, задайте параметры координат `x`, `y`. Определите метод `geolocation`, печатающий координаты месторасположения. Создайте класс-наследник `Employee`, задайте его координаты. Создайте для него метод `ChangePosition` для изменения координат при движении, метод `department` – для вывода сообщения о

принадлежности работника к отделу продаж. Сместите Employee на произвольное расстояние. Выведите сообщение о координатах завода, координатах рабочего и отделе, где он работает, а также - о координатах расстояния между заводом и рабочим.

Задача 2. Реализовать при помощи абстрактного класса подсчет факториала числа n и двойного факториала числа n , выполнить проверку и вывести результат на экран.

Задача 3. Создать класс Donor и описать в нем имя реципиента, его группу крови, имя донора и его группу крови, группа крови может задаваться как строкой, так и целым числом. Описать метод blood, определяющий может ли человек быть донором для данного больного (одинаковая группа крови).

Задача 4. При генерации выписки реквизитов клиентов оператора фискальных данных для юридических лиц печатается название, ИНН и ОГРН, а для индивидуальных предпринимателей фамилия, ИНН и ОГРНИП. Требуется сгенерировать список клиентов ОФД и напечатать выписки реквизитов, используя виртуальный метод из абстрактного класса.

Вариант 7

Задача 1. Создайте класс Flower, определите метод color, выводящий сообщение о цвете цветка, метод components, подсчитывающий число листков и лепестков в цветке и метод printComponents, выводящий на экран число компонентов цветка. Создайте два класса наследника: Rose and Tulip. Задайте для них переменные, отвечающие за количество лепестков и цвет, а для класса Rose задайте ещё параметр количества шипов. Переопределите для класса Rose метод components, добавив в расчёты параметр количества

шипов. Выведите для каждого цветка его цвет и количество компонентов.

Задача 2. Создать абстрактного класса Pair с виртуальными арифметическими операциями(+, -, *, /) и реализовать произвольный класс Rational (класс рациональных чисел) и на примере дробей протестировать все арифметические операции. Вывести все операции и полученный результат на экран.

Задача 3. Создать класс Birthday и описать в нем имя человека, день его рождения, месяц и год, при этом месяц может задаваться, как номером(целым числом), так и словом(строкой). Описать метод print, выводящий дату рождения в формате dd.mm.yyyy.

Задача 4. Требуется рассчитать периметр и площадь фигур различной геометрической формы. Создать абстрактный класс Shape и реализовать в нем абстрактные методы, которые должны вычислять периметр и площадь заданной геометрической формы. Для данной задачи рассчитайте площадь квадрата, круга и прямоугольного треугольника.

Вариант 8

Задача 1. . Создайте класс Anima, задайте для него метод location, выводящий сообщение о том, что животное живёт в зоопарке, метод amountOfFeed, входная переменная которого отвечает за количество животных одного типа, а сам метод рассчитывает количество необходимого корма. Также создайте метод printAmountOfFeed, выводящий сообщение о количестве корма, который необходим животным. Создайте два класса наследника: Penguin и Giraffe. Введите и количество. Переопределите метод location для класса Penguin, изменив его место жительства на Арктику. Выведите для

каждого типа животных их место жительства и необходимое количество корма для их питания.

Задача 2. Написать программу с использованием абстрактного класса для определения средней оценки студента. Пользователь должен все текущие оценки и наименование предмета. Подсчитать предполагаемый средний балл и вывести результат на экран.

Задача 3. Создать класс Navigator и описать в нем номер автомобиля, место отправления и место назначения (можно задать как номером региона, так и названием города).

Описать метод travel, определяющий, покинет ли машина данный регион.

Задача 4. Требуется рассчитать объем фигур различной геометрической формы. Создать абстрактный класс Shape и реализовать в нём абстрактные методы, которые должны вычислять объем заданного тела вращения. Для данной задачи рассчитайте объем куба, шара и цилиндра.

Вариант 9

Задача 1. Создайте класс Student, определите метод amountOfHomework, подсчитывающий сумму необходимых к выполнению домашних заданий по трём предметам, получившееся число метод printAmountOfStudent выводит на экран; метод group, выводит на экран сообщение о том, что студент обучается в группе ИТД. Создайте два класса-наследника Kate и Peter. Для класса Peter переопределите метод amountOfHomework, убавив один входной параметр. Введите количество заданий по предметам, необходимых для выполнения каждым студентом. Выведите для каждого студента сумму необходимых к выполнению домашних заданий и его группу.

Задача 2. Создать класс, описывающий структуру комплексного числа. Используя абстрактный класс, найти и вывести на экран сопряженное ему число.

Задача 3. Создать класс Book и описать в нем имя автора книги и ее идентификация в библиотеке. Идентификация может быть кодом книги (8-значное число) или названием (строка).

Описать метод order, определяющий, должна ли книга стоять до или после данной (по номеру или по алфавиту)

Задача 4. Создать абстрактный базовый класс «Товар» с методами, которые:

- Выводят на экран информацию о товаре (Наименование, срок годности)

- Определяют, соответствует ли данный товар сроку годности на данный момент*

Сформировать массив из n товаров, вывести полную информацию о товарах из массива на экран, а также найти просроченный товар (на момент текущей даты).

* - для данной задачи используйте класс GregorianCalendar из библиотеки Java.

Вариант 10

Задача 1. Создайте класс Peoples, задайте ФИО. Создайте класс-наследник Students, заполните его ФИО. Создайте для него метод ChangeAestimate для изменения оценок студента, метод WriteAestimate – для вывода массива с оценками и ФИО. Задайте произвольные ФИО и оценки. Выведите список, состоящий из трёх студентов.

Задача 2. Написать программу с использованием абстрактного класса для нахождения среднего арифметического заданного ряда целых чисел.

Задача 3. Создать класс Password и описать в нем ник человека на сайте и его пароль, при этом пароль может быть четырёхзначным пин-кодом (целое число, состоящее из 4 цифр) или полноценным паролем (строкой). Описать метод coincidence, определяющий, является ли введенное с клавиатуры n паролем для данного ника.

Задача 4. Создать абстрактный базовый класс «Triangle» для треугольника и вычислить площадь и периметр с помощью абстрактных методов. Поля данных должны включать две стороны и угол между ними. Определить производные классы: прямоугольный треугольник, равнобедренный треугольник, равносторонний треугольник с собственными функциями вычисления площади и периметра.

Вариант 11

Задача 1. Создайте класс Animals и два класса-наследника Home и Wild. Распределите животных ЛЕВ, МЕДВЕДЬ, СОБАКА, КОШКА по данным классам и добавьте породу/вид. Создайте для каждого класса-наследника метод GetBreed для вывода породы/вида животного.

Задача 2. Создать класс для описания понятия «вершина», содержащий в себе название горной вершины и ее высоту. С помощью абстрактного класса определить самую высокую и самую низкую вершины.

Задача 3. Создать класс Cards и описать в нем масть и значение карты(целое число или строка).

Описать метод определяющий, побьет ли эта карта девятку. (масть не учитывать)

Задача 4. Требуется рассчитать периметр и площадь фигур различной геометрической формы. Создать абстрактный класс Shape и реализовать в нем абстрактные методы, которые должны вычислять периметр и площадь заданной геометрической формы. Для данной задачи рассчитайте площадь квадрата, круга и прямоугольного треугольника.

Вариант 12

Задача 1. Создайте класс SunSystem и два класса наследника Planets и Satellites для которых определите диаметр и привязку спутника к планете. Опишите метод вывода планеты на экран GetPlanet. При выводе на экран планеты автоматически выводится ее диаметр и ее спутники.

Задача 2. Дан класс «Библиотека», содержащий информацию: название книги, автора и год издания. Используя абстрактный класс, определить названия книги и авторов, изданных в 1968 году.

Задача 3. Создать класс Student и описать в нем имя ученика, оценки по математике, физике и информатике. При этом оценки могут быть в формате числа(4) или строки(четыре). Описать метод mark, выводящий на экран, является ли ученик отличником, хорошистом или ни тем, ни другим.

Задача 4. Требуется рассчитать объем фигур различной геометрической формы. Создать абстрактный класс Shape и реализовать в нём абстрактные методы, которые должны вычислять объем заданного тела вращения. Для данной задачи рассчитайте объем куба, шара и цилиндра.

Вариант 13

Задача 1. Создайте класс Pond и два класса наследника Sea и Lake. Для каждого задайте размер водоёма и для класса Sea минимальный размер. На ввод с клавиатуры приходит название водоёма и его размер – в ответе программы содержится информация о типе водоёма (Sea или Lake соответственно).

Задача 2. Создать класс для описания рационального дробного числа, содержащий объекты ЧИСЛИТЕЛЬ и ЗНАМЕНАТЕЛЬ. Создать абстрактный класс, позволяющий определить равенство чисел.

Задача 3. Создать класс Mail и описать в нем имя получателя, количество писем и почтовое отделение. Почтовое отделение может быть задано номером или названием. Описать метод destination, определяющий, находится ли клиент в нужном отделении по введенному отделению.

Задача 4. В очереди за бесплатными макбуками стоят студенты групп ИТД и ЭВМ. У каждого студента есть целочисленный балльный рейтинг, изначально равный нулю. Студенты могут сдавать лабораторные работы по программированию (метод *.makeProgramming()*) и по электротехнике (метод *.makeElecEngineering()*). За работу по программированию студенту ЭВМ зачисляется случайным образом 3 или 4 балла, а за электротехнику 4 или 5. Для студентов ИТД наоборот. Необходимо случайным образом сгенерировать модель очереди студентов, провести несколько работ, и определить, сколько студентов ИТД и сколько студентов ЭВМ получают макбук. Макбуков раздается около 25% от количества всех студентов, и получают из лучшие по рейтингу. После каждого существенного шага следует выводить на экран состояния студентов, для реализации модели необходимо использовать абстрактные классы и оператор *instanceof*.

Вариант 14

Задача 1. Создайте класс Staff и два класса наследника Manager и Cashier. Определите график работы по дням недели для каждого из классов. Создайте метод GetWork для вывода графика работы каждого класса и метод TodayWork, принимающий на вход числовое значение дня недели и отвечающий работает ли сегодня персонал или нет.

Задача 2. Создать класс АВТОМОБИЛИСТЫ, содержащий не более 10 объектов, указав фамилию, марку (модель) автомобиля и его номер. Используя абстрактный класс поиска, вывести на экран фамилию владельца и номер машины конкретной марки.

Задача 3. Создать класс Dream и описать в нем имя человека и количество часов сна за последние сутки(восемь или 8). Описать метод normal, определяющий выспался ли человек(проспал более 8 часов) и если нет, то сколько часов сна ему еще нужно.

Задача 4. В компьютерной игре Minecraft присутствуют овцы и прирученные волки. У овец может быть до 8.0 единиц здоровья, а у волков до 20.0. Положение волка или овцы задается трехмерным радиус-вектором от начала координат. У волка присутствует способность атаковать овцу: при атаке у овцы отнимается $(h / h_f) d / r^2$ единиц здоровья, где

- h_f — константа полного здоровья волка,
- h — текущее здоровье волка,
- d — константа урона волка, равная 4-м,
- r — расстояние от волка до овцы.

Необходимо создать модель атаки волка на овцу, позволяющую рассчитать, сколько секунд понадобится волку для убийства овцы при их заданных состояниях. Использовать абстрактный суперкласс.

Вариант 15

Задача 1. Создайте класс Rooms и определите для него кол-во комнат в номере гостиницы. Распределите номера (от 1-го до 10-го) по кол-ву комнат в нём, создав классы наследники. Опишите метод GetRoom, получающий на вход число человек в семье, заезжающей в номер и выводящий число подходящего для семьи номера из расчёта 1 комната на одного человека.

Задача 2. Необходимо создать абстрактный класс, использующийся для вывода на экран информации о неуспевающих студентах (фамилия и группа). Информация о студентах находится в классе СТУДЕНТ: фамилия, группа, оценка за экзамен.

Задача 3. Создать класс Student и описать в нем 2 оценки за экзамен (число или строка) и 2 зачета(+ или -). Описать метод scholarship, определяющий, получит ли человек стипендию (оценки за экзамен не ниже 4 и получены зачеты).

Задача 4. Создать абстрактный базовый класс «Товар» с методами, которые:

- Выводят на экран информацию о товаре (Наименование, срок годности)
- Определяют, соответствует ли данный товар сроку годности на данный момент*

Сформировать массив из n товаров, вывести полную информацию о товарах из массива на экран,

а также найти просроченный товар (на момент текущей даты).

* - для данной задачи используйте класс GregorianCalendar из библиотеки Java.

Вариант 16

Задача 1. Создайте класс Checkers, содержащий метод movement для изменения координат фигуры при перемещении (x,y), метод printPosition - для печати координат и метод color - для вывода сообщения, о том, что цвет фигуры белый. Создайте класс-наследник Queen, начальные координаты которого (0,0), переопределите метод printPosition таким образом, чтобы при выходе за границы игрового поля выводилось сообщение о невозможности хода. Создайте класс-наследник Pawn (0,0) класса Queen, переопределите метод color, изменив цвет фигуры на чёрный. Сместите фигуры на произвольное расстояние. Выведите информацию о координатах и цвете каждой из фигур

Задача 2. Дан класс ЛЮДИ: пол, год рождения, цвет глаз. Используя абстрактный класс, вывести на экран количество кареглазых женщин 1980 года рождения.

Задача 3. Создать класс Mail и описать в нем имя отправителя, имя получателя, место получения и место отправления, при этом место отправления и получения могут быть заданы индексами(целым числом), либо названием города(строкой).

Описать метод travel, определяющий покинет ли письмо пределы города (места получения и отправления отличаются).

Задача 4. Создать абстрактный базовый класс «Triangle» для треугольника и вычислить площадь и периметр с помощью абстрактных методов. Поля данных должны включать две стороны и угол между ними. Определить производные классы: прямоугольный треугольник, равнобедренный треугольник, равносторонний треугольник с собственными функциями вычисления площади и периметра.

Вариант 17

Задача 1. Создайте класс `Student`, определите метод `amountOfHomework`, подсчитывающий сумму необходимых к выполнению домашних заданий по трём предметам, получившееся число метод `printAmountOfStudent` выводит на экран; метод `group`, выводит на экран сообщение о том, что студент обучается в группе ИТД. Создайте два класса-наследника `Kate` и `Peter`. Для класса `Peter` переопределите метод `amountOfHomework`, убавив один входной параметр. Введите количество заданий по предметам, необходимых для выполнения каждым студентом. Выведите для каждого студента сумму необходимых к выполнению домашних заданий и его группу.

Задача 2. Написать программу с использованием абстрактного класса для перевода чисел в двоичную систему счисления. Пользователь должен ввести десятичное число, а получить результат на экране в двоичном виде.

Задача 3. Создать класс `Book` и описать в нем имя автора книги и ее идентификация в библиотеке. Идентификация может быть кодом книги (8-значное число) или названием (строка).

Описать метод `order`, определяющий, должна ли книга стоять до или после данной (по номеру или по алфавиту)

Задача 4. Для описания всех людей, находящихся в магазине, необходимо выделить подмножество работников и подмножество посетителей. У каждого человека есть полное имя, у сотрудников магазина имеет значение должность, а у посетителей возраст. Необходимо сгенерировать список людей в магазине и вызовом виртуального метода из абстрактного класса напечатать для сотрудников фамилию и должность, а для посетителей имя и возраст.

Вариант 18

Задача 1. Создайте класс `SunSystem` и два класса наследника `Planets` и `Satellites` для которых определите диаметр и привязку спутника к планете. Опишите метод вывода планеты на экран `GetPlanet`. При выводе на экран планеты автоматически выводится ее диаметр и ее спутники.

Задача 2. Реализовать при помощи абстрактного класса поиск чисел Фибоначчи из заданного ряда целых чисел.

Задача 3. Создать класс `Car` и описать в нем марку машины, ее номер и скорость. Скорость можно указать как в формате целого числа, так и строки (60, шестьдесят). Описать метод `enter`, заполняющий все поля класса с клавиатуры.

Задача 4. Сильная и независимая женщина решила исследовать процесс кормления своих питомцев на компьютерной модели. У каждого питомца есть параметр *сытость*, задаваемый целым числом процентов в диапазоне (0, 100]. При симуляции кормления (метод `.eat()`) модель собаки должна печатать в консоль "Гав!", а модель кошки "Мяу!". Кормление добавляет к сытости до 75% для любого животного, так, что в итоге сытость оказывается не более 100%. Необходимо реализовать такую модель, используя абстрактные классы, случайным образом сгенерировать массив питомцев и распечатать его состояние до и после кормления всех животных.

Вариант 19

Задача 1. Создайте класс `Staff` и два класса наследника `Manager` и `Cashier`. Определите график работы по дням недели для каждого из классов. Создайте метод `GetWork` для вывода графика работы каждого класса и метод `TodayWork`, принимающий на вход числовое значение дня недели и отвечающий работает ли сегодня персонал или нет.

Задача 2. Написать программу с использованием абстрактного класса для вычисления синуса угла, заданного рядом Тейлора.

Задача 3. Создать класс Mail и описать в нем имя получателя, количество писем и почтовое отделение. Почтовое отделение может быть задано номером или названием.

Описать метод destination, определяющий, находится ли клиент в нужном отделении по введенному отделению.

Задача 4. Требуется реализовать часть виртуальной машины языка Brainfuck: список из объектов-операций, имеющих метод .execute(), принимающий по ссылке массив памяти и номер активной ячейки. Операции в BF:

1. '>' — перейти к следующей ячейке;
2. '<' — перейти к предыдущей ячейке;
3. '+' — увеличить значение в текущей ячейке на 1;
4. '-' — уменьшить значение в текущей ячейке на 1;
5. '.' — напечатать значение из текущей ячейки,
6. ';' — ввести значение и сохранить в текущей ячейке;
7. '[' — если значение текущей ячейки ноль, перейти вперёд по тексту программы на ячейку, следующую за соответствующей ']' (с учётом вложенности);
8. ']' — если значение текущей ячейки не ноль, перейти назад по тексту программы на символ '[' (с учётом вложенности).

Применить абстрактный класс для задания операции.

Вариант 20

Задача 1. Создайте класс Flower, определите метод color, выводящий сообщение о цвете цветка, метод components, подсчитывающий число листков и лепестков в цветке и метод printComponents, выводящий на экран число компонентов цветка. Создайте два класса наследника: Rose and Tulip. Задайте для них

переменные, отвечающие за количество лепестков и цвет, а для класса Rose задайте ещё параметр количества шипов. Переопределите для класса Rose метод components, добавив в расчёты параметр количества шипов. Выведите для каждого цветка его цвет и количество компонентов.

Задача 2. Дан класс «Библиотека», содержащий информацию: название книги, автора и год издания. Используя абстрактный класс, определить названия книги и авторов, изданных в 1968 году.

Задача 3. Создать класс Password и описать в нем ник человека на сайте и его пароль, при этом пароль может быть четырёхзначным пин-кодом (целое число, состоящее из 4 цифр) или полноценным паролем (строкой). Описать метод coincidence, определяющий, является ли введённое с клавиатуры n паролем для данного ника.

Задача 4. Требуется рассчитать объем фигур различной геометрической формы. Создать абстрактный класс Shape и реализовать в нём абстрактные методы, которые должны вычислять объем заданного тела вращения. Для данной задачи рассчитайте объем куба, шара и цилиндра.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение термину объектно-ориентированное программирование.
2. Раскройте значения следующих терминов: класс, объект.
3. Опишите назначение и процесс вызова явно и неявно определённых конструкторов и деструкторов.
4. Запишите фрагмент кода программы на языке Java для создания класса Car с методом printLocation и создайте объект данного класса.
5. Изобразите схематично многоуровневую иерархию наследования.
6. Опишите специфику принципа полиморфизм.
7. Расшифруйте обозначение abc и обоснуйте необходимость использования данного модуля.
8. Приведите пример использования ключевого слова this при создании метода.
9. Раскройте необходимость ограничения доступа к данным.
10. Объясните функцию запрета наследования.
11. Опишите преимущества использования абстрактных классов.
12. Запишите фрагмент кода программы на языке Java для перебора всех элементов массива и подсчёта их среднего арифметического.
13. Опишите назначение каждого из четырёх основных принципов ООП.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета. Отчет на защиту предоставляется в печатном виде.

Порядок выполнения:

1. Изучить теоретический материал.
2. Получить вариант у преподавателя.
3. Разработать программы согласно варианту.
4. Выполнить тестирование программ.
5. Продемонстрировать работу программ преподавателю.
6. Оформить отчет.
7. Защитить выполненную работу у преподавателя.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Васильев А.Н. Python на примерах [Электронный ресурс]: практический курс по программированию/ Васильев А.Н.— Электрон. текстовые данные.— СПб.: Наука и Техника, 2017.— 432 с.— Режим доступа: <http://www.iprbookshop.ru/73043.html>
2. Буйначев С.К. Основы программирования на языке Python [Электронный ресурс]: учебное пособие/ Буйначев С.К., Боклаг Н.Ю. — Электрон. текстовые данные.— Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2014.— 92 с.— Режим доступа: <http://www.iprbookshop.ru/66183.html>

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

4. Хахаев, И.А. Практикум по алгоритмизации и программированию на Python [Электронный ресурс] : учебное пособие / И.А. Хахаев. — Электрон. дан. — Москва : , 2016. — 178 с. — Режим доступа: <https://e.lanbook.com/book/100377>
5. Сузи Р.А. Язык программирования Python [Электронный ресурс]/ Сузи Р.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 350 с. — Режим доступа: <http://www.iprbookshop.ru/52211.html>

Электронные ресурсы:

5. Электронно-библиотечная система <http://www.iprbookshop.ru>
6. Электронно-библиотечная система <http://e.lanbook.com>
7. Руководство по языку программирования Java [Электронный ресурс]//Сайт о программировании “METANIT”: сайт – Режим доступа: <https://metanit.com/python/tutorial/1.1.php>

