**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  |  |
| Герасимович Н.И. |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г

**Задание:**

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь.](https://docs.python.org/3/library/abc.html)
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь.](https://docs.python.org/3/library/functions.html" \l "property)
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля [math.](https://docs.python.org/3/library/math.html)
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
   * Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
   * Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов. Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
    * Прямоугольник синего цвета шириной N и высотой N.
    * Круг зеленого цвета радиусом N.
    * Квадрат красного цвета со стороной N.
    * Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Текст программы на Python:**

**figure.py**

import abc

class ЦветФигуры:

def \_\_init\_\_(self, цвет):

self.цвет = цвет

class ГеометрическаяФигура(abc.ABC):

@abc.abstractmethod

def площадь(self):

pass

файл rectangle.py

from figure import ГеометрическаяФигура, ЦветФигуры

class Прямоугольник(ГеометрическаяФигура):

\_НазваниеФигуры = "Прямоугольник"

def \_\_init\_\_(self, ширина, высота, цвет):

super().\_\_init\_\_()

self.ширина = ширина

self.высота = высота

self.цвет = ЦветФигуры(цвет)

def площадь(self):

return self.ширина \* self.высота

def \_\_repr\_\_(self):

return f"{self.\_НазваниеФигуры} ({self.ширина} x {self.высота}, {self.цвет.цвет}), площадь: {self.площадь()}"

файл square.py

from rectangle import Прямоугольник

class Квадрат(Прямоугольник):

\_НазваниеФигуры = "Квадрат"

def \_\_init\_\_(self, сторона, цвет):

super().\_\_init\_\_(сторона, сторона, цвет)

файл circle.py

import math

from figure import ГеометрическаяФигура, ЦветФигуры

class Круг(ГеометрическаяФигура):

\_НазваниеФигуры = "Круг"

def \_\_init\_\_(self, радиус, цвет):

super().\_\_init\_\_()

self.радиус = радиус

self.цвет = ЦветФигуры(цвет)

def площадь(self):

return math.pi \* self.радиус\*\*2

def \_\_repr\_\_(self):

return f"{self.\_НазваниеФигуры} (радиус: {self.радиус}, {self.цвет.цвет}), площадь: {self.площадь()}"

файл main.py

from rectangle import Прямоугольник

from circle import Круг

from square import Квадрат

# Замените N на желаемые значения

прямоугольник = Прямоугольник(5, 10, "синий")

круг = Круг(7, "зеленый")

квадрат = Квадрат(8, "красный")

print(прямоугольник)

print(круг)

print(квадрат)

**Текст кода на C:  
  
#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <math.h>**

**// Структура для цвета фигуры**

**typedef struct {**

**char\* цвет;**

**} ЦветФигуры;**

**// Функция для создания объекта ЦветФигуры**

**ЦветФигуры\* создатьЦветФигуры(const char\* цвет) {**

**ЦветФигуры\* новыйЦвет = malloc(sizeof(ЦветФигуры));**

**if (новыйЦвет == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**exit(1);**

**}**

**новыйЦвет->цвет = strdup(цвет); // Используем strdup для копирования строки**

**if (новыйЦвет->цвет == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**free(новыйЦвет); // Освобождаем уже выделенную память**

**exit(1);**

**}**

**return новыйЦвет;**

**}**

**// Структура для абстрактного класса ГеометрическаяФигура**

**typedef struct ГеометрическаяФигура ГеометрическаяФигура;**

**struct ГеометрическаяФигура {**

**void (\*площадь)(ГеометрическаяФигура\*, double\*);**

**char\* название;**

**ЦветФигуры\* цвет;**

**};**

**// Класс Прямоугольник**

**typedef struct {**

**ГеометрическаяФигура super;**

**double ширина;**

**double высота;**

**} Прямоугольник;**

**// Класс Квадрат**

**typedef struct {**

**Прямоугольник super;**

**} Квадрат;**

**// Класс Круг**

**typedef struct {**

**ГеометрическаяФигура super;**

**double радиус;**

**} Круг;**

**// Класс Треугольник**

**typedef struct {**

**ГеометрическаяФигура super;**

**double основание;**

**double высота;**

**} Треугольник;**

**// Класс Трапеция**

**typedef struct {**

**ГеометрическаяФигура super;**

**double основание1;**

**double основание2;**

**double высота;**

**} Трапеция;**

**// Функции для вычисления площадей**

**void площадьПрямоугольника(ГеометрическаяФигура\* self, double\* result) {**

**Прямоугольник\* прямоугольник = (Прямоугольник\*)self;**

**\*result = прямоугольник->ширина \* прямоугольник->высота;**

**}**

**void площадьКруга(ГеометрическаяФигура\* self, double\* result) {**

**Круг\* круг = (Круг\*)self;**

**\*result = M\_PI \* круг->радиус \* круг->радиус;**

**}**

**void площадьТреугольника(ГеометрическаяФигура\* self, double\* result) {**

**Треугольник\* треугольник = (Треугольник\*)self;**

**\*result = 0.5 \* треугольник->основание \* треугольник->высота;**

**}**

**void площадьТрапеции(ГеометрическаяФигура\* self, double\* result) {**

**Трапеция\* трапеция = (Трапеция\*)self;**

**\*result = 0.5 \* (трапеция->основание1 + трапеция->основание2) \* трапеция->высота;**

**}**

**// Функции для создания фигур**

**Прямоугольник\* создатьПрямоугольник(double ширина, double высота, const char\* цвет) {**

**Прямоугольник\* new\_прямоугольник = malloc(sizeof(Прямоугольник));**

**if (new\_прямоугольник == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**exit(1);**

**}**

**new\_прямоугольник->super.название = "Прямоугольник";**

**new\_прямоугольник->super.цвет = создатьЦветФигуры(цвет);**

**new\_прямоугольник->ширина = ширина;**

**new\_прямоугольник->высота = высота;**

**new\_прямоугольник->super.площадь = площадьПрямоугольника;**

**return new\_прямоугольник;**

**}**

**Квадрат\* создатьКвадрат(double сторона, const char\* цвет) {**

**Квадрат\* new\_квадрат = malloc(sizeof(Квадрат));**

**if (new\_квадрат == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**exit(1);**

**}**

**new\_квадрат->super.super.название = "Квадрат";**

**new\_квадрат->super.super.цвет = создатьЦветФигуры(цвет);**

**new\_квадрат->super.ширина = сторона;**

**new\_квадрат->super.высота = сторона;**

**new\_квадрат->super.super.площадь = площадьПрямоугольника;**

**return new\_квадрат;**

**}**

**Круг\* создатьКруг(double радиус, const char\* цвет) {**

**Круг\* new\_круг = malloc(sizeof(Круг));**

**if (new\_круг == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**exit(1);**

**}**

**new\_круг->super.название = "Круг";**

**new\_круг->super.цвет = создатьЦветФигуры(цвет);**

**new\_круг->радиус = радиус;**

**new\_круг->super.площадь = площадьКруга;**

**return new\_круг;**

**}  
  
Треугольник\* создатьТреугольник(double основание, double высота, const char\* цвет) {**

**Треугольник\* new\_треугольник = malloc(sizeof(Треугольник));**

**if (new\_треугольник == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**exit(1);**

**}**

**new\_треугольник->super.название = "Треугольник";**

**new\_треугольник->super.цвет = создатьЦветФигуры(цвет);**

**new\_треугольник->основание = основание;**

**new\_треугольник->высота = высота;**

**new\_треугольник->super.площадь = площадьТреугольника;**

**return new\_треугольник;**

**}**

**Трапеция\* создатьТрапецию(double основание1, double основание2, double высота, const char\* цвет) {**

**Трапеция\* new\_трапеция = malloc(sizeof(Трапеция));**

**if (new\_трапеция == NULL) {**

**fprintf(stderr, "Ошибка выделения памяти.\n");**

**exit(1);**

**}**

**new\_трапеция->super.название = "Трапеция";**

**new\_трапеция->super.цвет = создатьЦветФигуры(цвет);**

**new\_трапеция->основание1 = основание1;**

**new\_трапеция->основание2 = основание2;**

**new\_трапеция->высота = высота;**

**new\_трапеция->super.площадь = площадьТрапеции;**

**return new\_трапеция;**

**}**

**// Функция для вывода информации о фигуре**

**void вывестиФигуру(ГеометрическаяФигура\* фигура) {**

**double площадь;**

**фигура->площадь(фигура, &площадь);**

**printf("%s (", фигура->название);**

**if (strcmp(фигура->название, "Круг") == 0) {**

**Круг\* круг = (Круг\*)фигура;**

**printf("радиус: %.2f", круг->радиус);**

**} else if (strcmp(фигура->название, "Треугольник") == 0) {**

**Треугольник\* треугольник = (Треугольник\*)фигура;**

**printf("основание: %.2f, высота: %.2f", треугольник->основание, треугольник->высота);**

**} else if (strcmp(фигура->название, "Трапеция") == 0) {**

**Трапеция\* трапеция = (Трапеция\*)фигура;**

**printf("основание1: %.2f, основание2: %.2f, высота: %.2f",**

**трапеция->основание1, трапеция->основание2, трапеция->высота);**

**} else {**

**Прямоугольник\* прямоугольник = (Прямоугольник\*)фигура;**

**printf("%.2f x %.2f", прямоугольник->ширина, прямоугольник->высота);**

**}**

**printf(", %s), площадь: %.2f\n", фигура->цвет->цвет, площадь);**

**}**

**// Функция для освобождения памяти**

**void освободитьФигуру(ГеометрическаяФигура\* фигура) {**

**if (фигура == NULL) return;**

**освободитьЦветФигуры(фигура->цвет);**

**free(фигура->название); // Освобождаем строку с названием**

**free(фигура);**

**}**

**// Функция для освобождения памяти для цвета**

**void освободитьЦветФигуры(ЦветФигуры\* цвет) {**

**if (цвет != NULL) {**

**free(цвет->цвет);**

**free(цвет);**

**}**

**}**

**int main() {**

**// Замените N на желаемые значения**

**Прямоугольник\* прямоугольник = создатьПрямоугольник(6, 6, "синий");**

**Круг\* круг = создатьКруг(6, "зеленый");**

**Квадрат\* квадрат = создатьКвадрат(6, "красный");**

**Треугольник\* треугольник = создатьТреугольник(6, 6, "желтый");**

**вывестиФигуру((ГеометрическаяФигура\*)прямоугольник);**

**вывестиФигуру((ГеометрическаяФигура\*)круг);**

**вывестиФигуру((ГеометрическаяФигура\*)квадрат);**

**вывестиФигуру((ГеометрическаяФигура\*)треугольник);**

**// Освобождаем ресурсы**

**освободитьФигуру(прямоугольник);**

**освободитьФигуру(круг);**

**освободитьФигуру(квадрат);**

**освободитьФигуру(треугольник);**

**return 0;**

**}**