



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Институт автоматики и информационных технологий
Кафедра «Электронные системы и информационная
безопасность»

О Т Ч Е Т

о выполнении лабораторной работы №2

«Классы в “Python”»

по дисциплине «Технологии и методы программирования»

Преподаватель	07.10.2023			Панфилова И.Е.
	(должность)	(подпись)	(дата)	(инициалы, фамилия)
Преподаватель	07.10.2023			Трофимычев И.И.
	(должность)	(подпись)	(дата)	(инициалы, фамилия)
Студент	18.11.2023			Марченко И.К.
	2-ИАИТ-102	(подпись)	(дата)	(инициалы, фамилия)
	(группа)			

Цель работы: изучение классов в “Python”.



2. Садовник и помидоры

Классовая структура

Предлагаем создать следующую классовую структуру:

Есть **Помидор** со следующими характеристиками:

1. Индекс
2. Стадия зрелости(стадии: Отсутствует, Цветение, Зеленый, Красный)

Помидор может:

1. Расти (переходить на следующую стадию созревания)
2. Предоставлять информацию о своей зрелости

Есть **Куст с помидорами**, который:

1. Содержит список томатов, которые на нем растут

И может:

1. Расти вместе с томатами
2. Предоставлять информацию о зрелости всех томатов
3. Предоставлять урожай

И также есть **Садовник**, который имеет:

1. Имя
2. Растение, за которым он ухаживает

И может:

1. Ухаживать за растением
2. Сбирать с него урожай

Задание

Класс **Tomato**:

1. Создайте класс **Tomato**
2. Создайте статическое свойство **states**, которое будет содержать все стадии созревания помидора
3. Создайте метод **__init__()**, внутри которого будут определены два динамических protected свойства: 1) **_index** - передается параметром и 2) **_state** - принимает первое значение из словаря **states**
4. Создайте метод **grow()**, который будет переводить томат на следующую стадию созревания
5. Создайте метод **is_ripe()**, который будет проверять, что томат созрел (достиг последней стадии созревания)

Класс **TomatoBush**

1. Создайте класс **TomatoBush**
2. Определите метод **__init__()**, который будет принимать в качестве параметра количество томатов и на его основе будет создавать список объектов класса **Tomato**. Данный список будет храниться внутри динамического свойства **tomatoes**.
3. Создайте метод **grow_all()**, который будет переводить все объекты из списка томатов на следующий этап созревания
4. Создайте метод **all_are_ripe()**, который будет возвращать **True**, если все томаты из списка стали спелыми
5. Создайте метод **give_away_all()**, который будет чистить список томатов после сбора урожая

Класс **Gardener**

1. Создайте класс **Gardener**
2. Создайте метод **__init__()**, внутри которого будут определены два динамических свойства: 1) **name** - передается параметром, является публичным и 2) **_plant** - принимает объект класса **TomatoBush**, является **protected**
3. Создайте метод **work()**, который заставляет садовника работать, что позволяет растению становиться более зрелым
4. Создайте метод **harvest()**, который проверяет, все ли плоды созрели. Если все - садовник собирает урожай. Если нет - метод печатает предупреждение.
5. Создайте статический метод **knowledge_base()**, который выведет в консоль справку по садоводству.

Тесты:

1. Вызовите справку по садоводству
2. Создайте объекты классов **TomatoBush** и **Gardener**
3. Используя объект класса **Gardener**, поухаживайте за кустом с помидорами
4. Попробуйте собрать урожай
5. Если томаты еще не дозрели, продолжайте ухаживать за ними
6. Соберите урожай

Выполнение работы

```
1  # Создание класса томатов
2  class Tomato:
3      states = {0: 'Отсутствует(-ют)', 1: 'Цветёт(-ут)', 2: 'Зеленый(-ые)', 3: 'Красный(-ые)'} # статическое свойство
4
5      def __init__(self, index): # создание томатов
6          self._index = index
7          self._state = 0
8
9      def grow(self): # переводит на след стадию
10         if self._state < 3:
11             self._state += 1
12             self._print_state()
13
14         def is_ripe(self): # проверка на зрелость
15             if self._state == 3:
16                 return True
17             else:
18                 return False
19
20         def _print_state(self): # Информация для меня, потому что я рукожоп
21             print(f'Помидор(-ы) {self._index} сейчас {Tomato.states[self._state]}')
22
23
24  # создание томатного куста
25  class TomatoBush:
26      def __init__(self, num): # количество томатов на кусте
27          self.tomatoes = [Tomato(index) for index in range(1, num)]
28
29      def grow_all(self): # томаты зреют
30          for tomato in self.tomatoes:
31              tomato.grow()
32
33      def all_are_ripe(self): # проверка все ли томаты созрели
34          return all([tomato.is_ripe() for tomato in self.tomatoes])
35
36      def give_away_all(self): # сбор урожая
37          self.tomatoes = []
38          print("Собираем урожай!")
39
40
41  class Gardener:
42      def __init__(self, name, plant):
43          self.name = name
44          self._plant = plant
45
46      def work(self): # садовник работает
47          print(self.name + " работает...")
48          self._plant.grow_all()
49          print(self.name + " закончил работать")
50
51      def harvest(self): # проверка плодов на зрелость и сбор урожая
52          print(self.name + " собирает урожай")
53          if self._plant.all_are_ripe():
54              self._plant.give_away_all()
55              print("Урожай Собран!")
56              return True
57          else:
58              print("!!!Не все томаты созрели!!!")
59              return False
60
61
62      @staticmethod
63      def knowlenge_base():
64          print("Справка по садоводству: ")
65          print("За томатами нужно ухаживать: поливать, удобрять, а также своевременно собирать урожай")
66
```

Тесты:

```

67
68 # Тесты
69 if __name__ == '__main__':
70
71     # 1
72     Gardener.knowledge_base()
73
74     # 2
75     bush = TomatoBush(2)
76     gardener = Gardener("Иван", bush)
77
78     # 3
79     gardener.work()
80
81     # 4
82     gardener.harvest()
83
84     # 5, 6
85     print('Цирк цикловой:')
86     while not gardener.harvest():
87         gardener.work()
88

```

Вывод на консоли:

```

Справка по садоводству:
За томатами нужно ухаживать: поливать, удобрять, а также своевременно собирать урожай
Иван работает...
Помидор(-ы) 1 сейчас Цветёт(-ут)
Иван закончил работать
Иван собирает урожай
!!!Не все томаты созрели!!!
Цирк цикловой:
Иван собирает урожай
!!!Не все томаты созрели!!!
Иван работает...
Помидор(-ы) 1 сейчас Зеленый(-ые)
Иван закончил работать
Иван собирает урожай
!!!Не все томаты созрели!!!
Иван работает...
Помидор(-ы) 1 сейчас Красный(-ые)
Иван закончил работать
Иван собирает урожай
Собираем урожай!
Урожай Собран!
PS C:\Users\WinterKakao\Desktop\TiMP>

```

Вывод:

Мы научились работать с классами: создавать их, присваивать им параметры, а также создавать методы.