**ПРАКТИЧЕСКАЯ РАБОТА № 18-19**

Классы и объекты

***Цель работы:*** Овладение приемами разработки простейших классов на языке С#. Изучение механизма инкапсуляции на основе свойств.

|  |  |  |
| --- | --- | --- |
| №  п/п | ФИО | Номер задачи |
|  | Балицкая Анастасия | 1 |
|  | Вартик Максим | 2 |
|  | Вдовиченко Ярослав | 3 |
|  | Веряскин Иван | 4 |
|  | Гоцалюк Арзу | 5 |
|  | Градинар Илья | 6 |
|  | Гребенщиков Данил | 7 |
|  | Завтонев Роман | 8 |
|  | Захаров Станислав | 9 |
|  | Ильинский Никита | 10 |
|  | Каплин Егор | 11 |
|  | Коваленко Михаил | 1 |
|  | Колак Михаил | 2 |
|  | Левицкая Мария | 3 |
|  | Милюкова Екатерина | 4 |
|  | Никифорова Елизавета | 5 |
|  | Подолян Евгений | 7 |
|  | Попов Максим | 6 |
|  | Родичев Максим | 8 |
|  | Рознован Роман | 9 |
|  | Тизул Игорь | 5 |
|  | Тихончук София | 11 |
|  | Толоченко Павел | 10 |
|  | Фокша Кирилл | 1 |
|  | Чебан Евгений | 2 |
|  | Чебан Илья | 4 |
|  | Юраш Ренат | 3 |

Задания для самостоятельной работы

Вариант 1

1. Описать класс **Product** (*Товар*) с полями: наименование, цена, количество. Определить в нем:

* конструктор, принимающий поля класса;
* метод, выводящий информацию об объекте;
* метод, возвращающий стоимость товара.

Написать программу, использующую этот класс и методы.

1. Модернизировать программу из предыдущего задания, реализовав принцип инкапсуляции посредством обеспечения доступа к полям классов через свойства. В случае необходимости вложите в свойства дополнительную логику проверки значений.

Вариант 2

1. Описать класс **Тriangle** (*Треугольник*)с полями *a*, *b*, *c* – длины 3-х сторон. Определить в нем:

* конструктор, принимающий поля класса;
* метод, выводящий информацию об объекте;
* метод, выводящий тип треугольника по сторонам (разносторонний, равнобедренный, равносторонний).

Написать программу, использующую этот класс и методы.

1. Модернизировать программу из предыдущего задания, реализовав принцип инкапсуляции посредством обеспечения доступа к полям классов через свойства. В случае необходимости вложите в свойства дополнительную логику проверки значений.

Вариант 3

1. Описать класс **Parallelogram** (*Параллелограмм*) с полями *a*, *b* – стороны параллелограмма и α – угол между сторонами. Определить в нем:

* конструктор, принимающий поля класса;
* метод, выводящий информацию об объекте;
* метод, выводящий вид параллелограмма (квадрат, прямоугольник, ромб).

Написать программу, использующую этот класс и методы.

1. Модернизировать программу из предыдущего задания, реализовав принцип инкапсуляции посредством обеспечения доступа к полям классов через свойства. В случае необходимости вложите в свойства дополнительную логику проверки значений.

Вариант 4

1. Описать класс **circle** (*Окружность*) с полями *x*0, *y*0 – координаты центра, *R*– радиус. Определить в нем:

* конструктор, принимающий поля класса;
* метод, выводящий информацию об объекте;
* методы, определяющие, пересекается ли окружность с осями *Ox* и *Oy*.

Написать программу, использующую этот класс и методы.

1. Модернизировать программу из предыдущего задания, реализовав принцип инкапсуляции посредством обеспечения доступа к полям классов через свойства. В случае необходимости вложите в свойства дополнительную логику проверки значений.

Вариант 5

1. Описать класс **Rectangle** (*Прямоугольник*), описывающий прямоугольник со сторонами, параллельными осям координат, и заданный координатами двух его противоположных вершин. Определить в нем:

* конструктор, принимающий поля класса;
* метод, выводящий информацию об объекте;
* метод, проверяющий попадание заданной точки внутрь прямоугольника.

Написать программу, использующую этот класс и методы.

1. Модернизировать программу из предыдущего задания, реализовав принцип инкапсуляции посредством обеспечения доступа к полям классов через свойства. В случае необходимости вложите в свойства дополнительную логику проверки значений.

Вариант 6

1.Напишите первую строку определения класса Poodle (Пудель). Этот класс должен расширять класс Dog.

2. Взгляните на приведенные ниже определения классов: class Plant: def \_init (self, plant\_type): self.\_plant\_type = plant\_type def message(self): print("Я - планета.") class Tree(Plant): def init (self): Plant. init (self, 'дерево') def rnessage(self): print("Я - дерево.") Что покажут приведенные ниже инструкции с учетом этих определений классов? р = Plant ('саженец') t = Tree () р .rnessage () t.rnessage ()

3. Взгляните на приведенное ниже определение: class Beverage: def init (self, bev\_narne): self. bev narne = bev narne Напишите программный код для класса с именем Cola (Кока-кола), который является подклассом класса Beverage (Напиток). Метод \_ ini t \_ () класса Cola должен вызывать метод ini t () класса Beverage, передавая строковое значение 'кока-кола' в качестве аргумента.

Вариант 7

1. Классы Employee и ProductionWorker. Напишите класс Ernployee (Сотрудник), который содержит атрибуты приведенных ниже данных: • имя сотрудника; • номер сотрудника. Затем напишите класс ProductionWorker (Рабочий), который является подклассом класса Ernployee. Класс ProductionWorker должен содержать атрибуты приведенных ниже данных: • номер смены (целое число, к примеру, 1, 2 или 3); • ставка почасовой оплаты труда. Рабочий день разделен на две смены: дневную и вечернюю. Атрибут смены будет содержать целочисленное значение, представляющее смену, в которую сотрудник работает. Дневная смена является сменой 1, вечерняя смена - сменой

2. Напишите соответствующие методы получатели и методы-модификаторы для каждого класса. После того как эти классы будут написаны, напишите программу, которая создает объект класса ProductionWorker и предлагает пользователю ввести данные по каждому атрибуту данных этого объекта. Сохраните данные в объекте и примените в этом объекте методы получатели, чтобы получить эти данные и вывести их на экран.

Вариант8

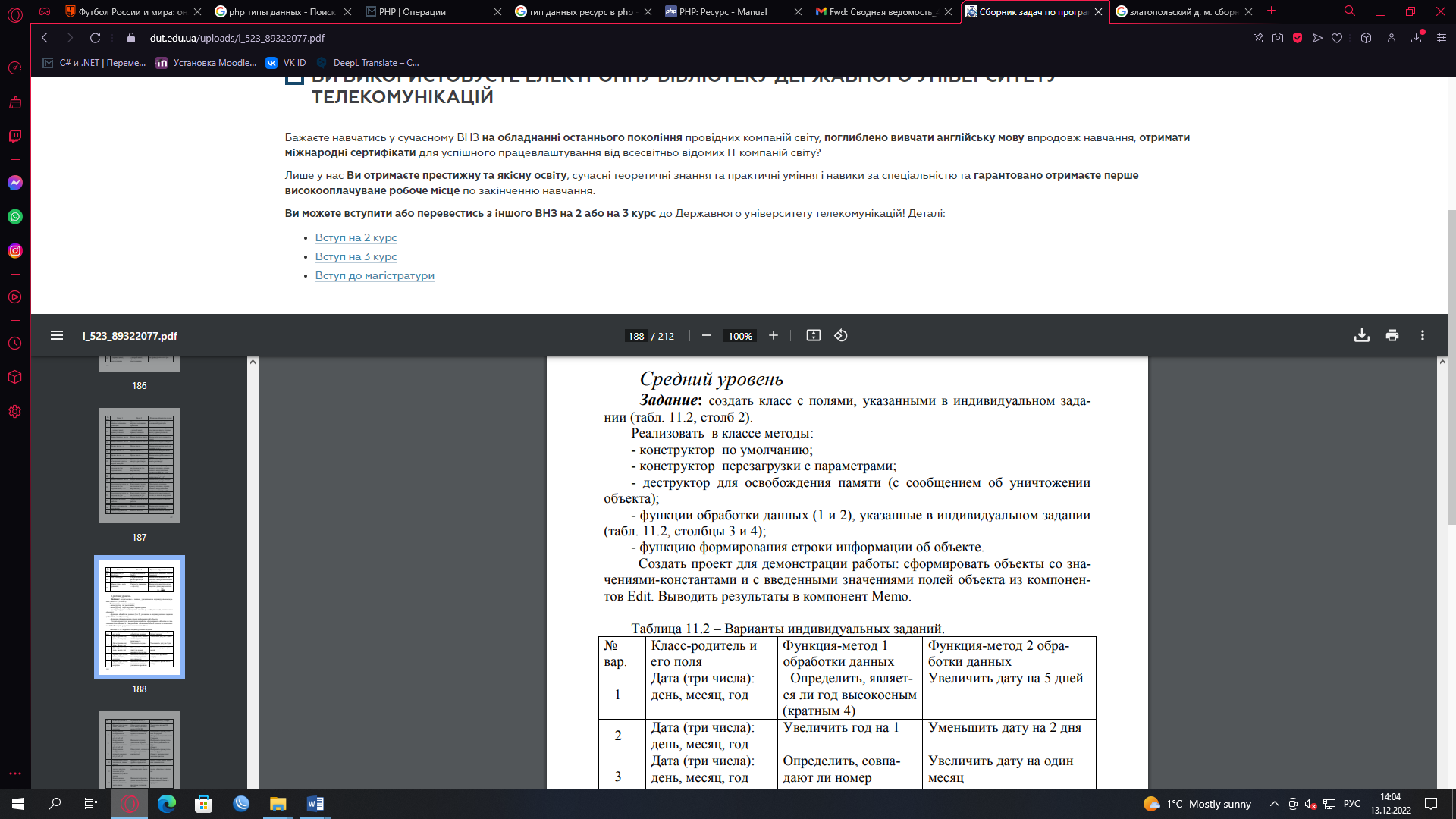
2. Класс ShiftSupervisor. На некой фабрике начальник смены является штатным сотрудником, который руководит сменой. В дополнение к фиксированному окладу начальник смены получает годовую премию за выполнение его сменой производственного плана. Напишите класс ShiftSupervisor (Начальник смены), который является подклассом класса Ernployee, созданного в задаче по программированию

1. Класс ShiftSupervisor должен содержать атрибут данных для годового оклада и атрибут данных для годовой производственной премии, которую заработал начальник смены. Продемонстрируйте класс, написав программу, которая применяет объект ShiftSupervisor.

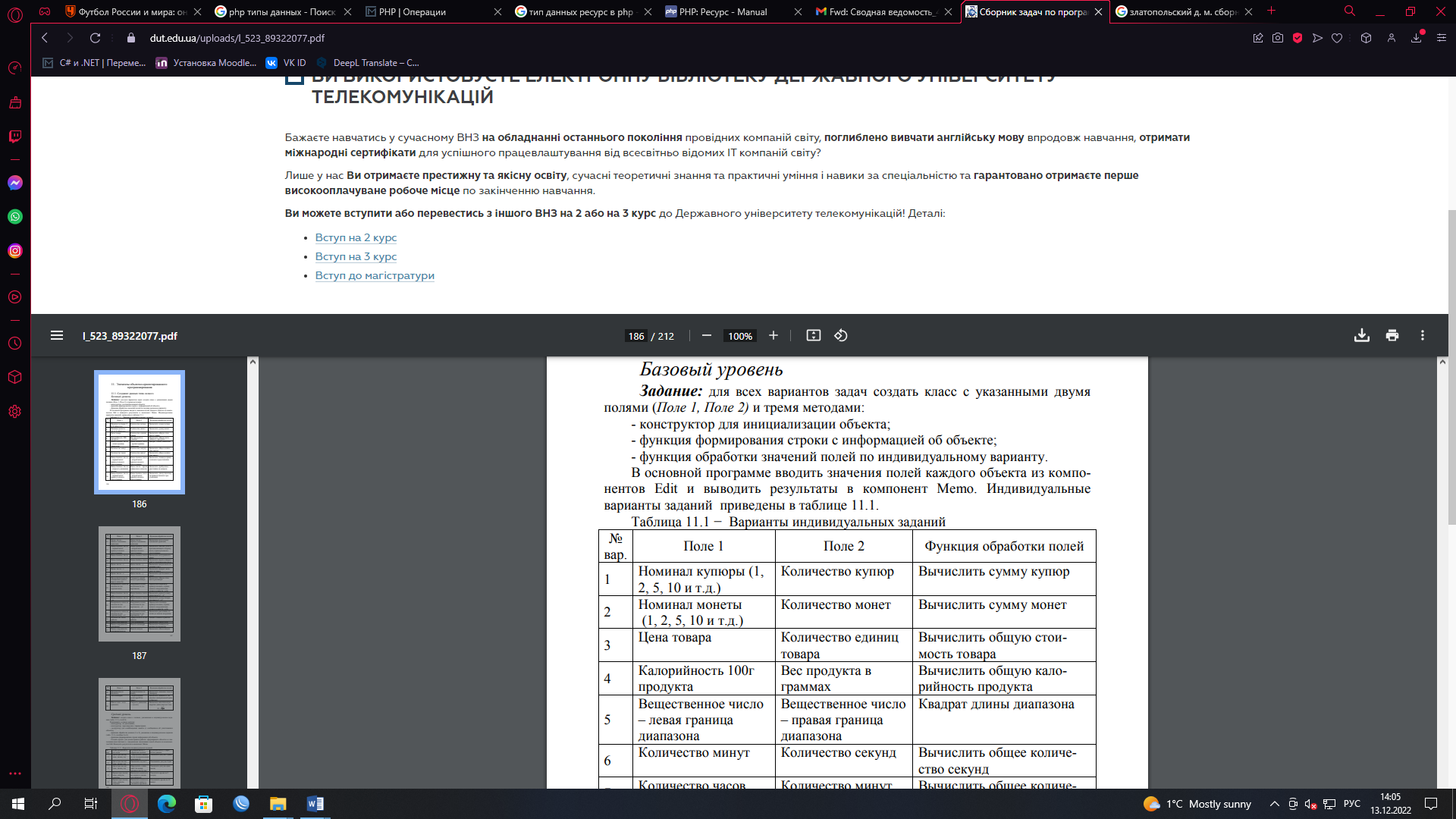
Вариант 9

3. Классы Person и Customer. Напишите класс Person с атрибутами данных для имени, адреса и телефонного номера человека. Затем напишите класс Customer (Клиент), который является подклассом класса Person. Класс Customer должен иметь атрибут данных для номера клиента и атрибут булевых данных, указывающий, хочет ли клиент быть в списке рассылки или нет. Продемонстрируйте экземпляр класса Custorner в простой программе.

Вариант 10



Вариант 11

.