Студента группы ИС-23 Галушкиной Дианы

Практическая работа №14.

Тема: составление программ с использованием ООП в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с ООП в IDE PyCharm Community.

Стиль выполнения программ: с применением стиля ООП.

Постановка задачи 1:

Создайте класс «Счетчик», который имеет атрибут текущего значения и методы для инкремента и декремента значения.

Текст программы 1:

```
# Создайте класс «Счетчик», который имеет атрибут текущего значения и методы для
# инкремента и декремента значения.
class Counter:
	def __init__(self):
		self.value = 0 # начальное значение счетчика
		def increment(self):
		self.value += 1
	def decrement(self):
		if self.value > 0:
			self.value -= 1
# Пример использования класса "Счетчик"
counter = Counter()
print(counter.value) # Вывод: 0
counter.increment()
counter.increment()
print(counter.value) # Вывод: 2
counter.decrement()
print(counter.value) # Вывод: 1
```

Постановка задачи 2:

Создайте класс "Автомобиль", который содержит информацию о марке, модели и годе выпуска. Создайте класс "Грузовик", который наследуется от класса "Автомобиль" и содержит информацию о грузоподъемности. Создайте класс "Легковой автомобиль", который наследуется от класса "Автомобиль" и содержит информацию о количестве пассажиров.

Текст программы 2:

```
# Создайте класс "Автомобиль", который содержит информацию о марке, модели и
# годе выпуска. Создайте класс "Грузовик", который наследуется от класса
# "Автомобиль" и содержит информацию о грузоподъемности. Создайте класс
# "Легковой автомобиль", который наследуется от класса "Автомобиль" и содержит
# информацию о количестве пассажиров.

class Automobile:
	def __init__(self, brand, model, year):
		self.brand = brand
		self.model = model
		self.year = year

class Truck(Automobile):
	def __init__(self, brand, model, year, payload_capacity):
		super().__init__(brand, model, year)
		self.payload_capacity = payload_capacity

class Car(Automobile):
	def __init__(self, brand, model, year, passenger_capacity):
		super().__init__(brand, model, year)
		self.passenger_capacity = passenger_capacity

# Пример использования классов
car1 = Car("Toyota", "Corolla", 2020, 5)
print(car1.brand) # Вывод: Toyota
print(car1.model) # Вывод: Corolla
print(car1.year) # Вывод: 2020
print(car1.passenger_capacity) # Вывод: 5

truck1 = Truck("Volvo", "FH16", 2019, 20000)
print(truck1.brand) # Вывод: Volvo
print(truck1.model) # Вывод: FH16
print(truck1.year) # Вывод: 2019
print(truck1.payload_capacity) # Вывод: 20000
```

Протокол работы программы 1:

```
0
2
1

Process finished with exit code 0
```

Протокол работы программы 2:

```
Toyota
```

Corolla
2020
5
Volvo
FH16
2019
20000

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составление программ с ООП в IDE PyCharm Community.