Український католицький університет
Кафедра комп’ютерних наук та інформаційних технологій
Білет № __ по курсу Основи програмування
Прізвище студента _____ Оцінка _____

Задача	Кількість балів	Тематика	Обов’язкові складові та їх оцінювання				
			Стиль програмування		Перевірка працездатності	Оцінка відповідності завданню	
			Код	Документ-тація		Алгоритм	Проходження тестів
1	10	Вирішення завдання з використанням ООП. Розробка простої ієрархії класів із забезпеченням коректного представлення об’єктів на екрані.	1	1	1 doctest	4	3
2	10	Вирішення завдання з використанням ООП. Розробка класів та методів для зміни стану об’єктів.	1	1	1 doctest	4	3
3	10	Вирішення завдання з використанням ООП та необхідністю розробки (модифікації) структур даних.	1	1	2 unittest	4	2

ЗАДАЧА 1

Розробити класи Стілець та Меблі. При реалізації класів потрібно використовувати елементи об’єктно-орієнтованого програмування. Наприклад, функцію super() або, якщо потрібно, перезавантаження методів. Реалізація цих класів повинна забезпечувати коректне виконання наступного коду:

```
furniture1 = Furniture("empire", "bedroom")
assert(furniture1.style == "empire")
assert(furniture1.assign == "bedroom")
assert(str(furniture1) == "<furniture style is empire>")
# A Chair is a kind of Furniture
chair1 = Chairt("empire", "bedroom", "armchair")
assert(chair1.tipe == "armchair")
assert(isinstance(chair1, Furtiture))
assert(str(chair1) == "<furniture style is empire>")
```

ЗАДАЧА 2

Напишіть програму, яка буде імітувати аудиторію в якій знаходяться стільчики та ослінчики. Спочатку в аудиторії знаходяться тільки парти і її можна змоделювати як список певної довжини (кількість парт) . Елементами цього списку є кортежі (парти). Кожен кортеж у цьому списку повинен містити один з наступних варіантів:

- два об’єкти Стільчик або два об’єкти Ослінчик;
- один об’єкт Стільчик а інший None;
- один об’єкт Ослінчик а інший None;
- два None.

На кожному кроці, випадковим чином в аудиторію заносять стільчик або ослінчик. Стільчики та ослінчики розставляють за парти і це змінює стан об’єкту Аудиторія. Стільчик (ослінчик) ставлять за парту – зі списку обирається кортеж і в одному з його елементів None замінюється на об’єкт Стільчик (Ослінчик). Правила розміщення стільчиків та ослінчиків за партами наступні:

- разом за однією партою можуть стояти тільки стільчики або тільки ослінчики;
- якщо за партою вже стоять два стільчики (ослінчики) то потрібно шукати вільне місце;
- якщо за партою вже стоїть стільчик (ослінчик) то для ослінчика (стільчика) потрібно шукати вільне місце.

Якщо в аудиторії вже немає місця то повинен збуджуватися відповідний виняток. Якщо для стільчика (ослінчика) не можна знайти місце за партою в аудиторії то повинен збуджуватися відповідний виняток.

ЗАДАЧА 3

Реалізувати метод replacement (data, item) в класі LinkedQueue. Цей метод повинен замінити елемент черги з вказаним значенням data на інший елемент item або повернути False, якщо такий елемент відсутній в черзі. (При вирішенні цієї задачі потрібно використовувати надану реалізацію черги).

Український католицький університет
Кафедра комп’ютерних наук та інформаційних технологій
Білет № __ по курсу Основи програмування
Прізвище студента _____ Оцінка _____

Задача	Кількість балів	Тематика	Обов’язкові складові та їх оцінювання				
			Стиль програмування		Перевірка праце-здатності	Оцінка відповідності завданню	
			Код	Документ-тація		Алгоритм	Проходження тестів
1	10	Вирішення завдання з використанням ООП. Розробка простої ієрархії класів із забезпеченням коректного представлення об’єктів на екрані.	1	1	1 doctest	4	3
2	10	Вирішення завдання з використанням ООП. Розробка класів та методів для зміни стану об’єктів.	1	1	1 doctest	4	3
3	10	Вирішення завдання з використанням ООП та необхідністю розробки (модифікації) структур даних.	1	1	2 unittest	4	2

ЗАДАЧА 1

Розробити класи Котик та Тварина. При реалізації класу потрібно використовувати елементи об'єктно-орієнтованого програмування. Наприклад, функцію super() або, якщо потрібно, перезавантаження методів. Реалізація цих класів повинна забезпечувати коректне виконання наступного коду:

```
animal1 = Animal("chordata", "mammalia")
assert(animal1.phylum == "chordata")
assert(animal1.clas == "mammalia")
assert(str(animal1) == "<animal class is mammalia>")
animal2 = Animal("chordata", "birds")
assert(not (animal1 == animal2))
# A Cat is a kind of Animal
cat1 = Cat("chordata", "mammalia", "felis")
assert(cat1.genus == "felis")
assert(isinstance(cat1, Animal))
assert(str(cat1) == "<animal class is mammalia>")
```

ЗАДАЧА 2

Напишіть програму, яка буде імітувати карусель на якій катаються котики та горобчики. Карусель можна змоделювати як список певної довжини елементами якого є кортежі (пари). Кожен кортеж у цьому списку повинен містити:

- два об'єкти Котик або два об'єкти Горобчик;
- один об'єкт Котик а інший None;
- один об'єкт Горобчик а інший об'єкт None ;
- два None.

Спочатку всі горобчики та котики сідають на карусель – встановлюється початковий стан об'єкту Карусель. Котик (Горобчик) сідає на карусель – зі списку обирається кортеж і в одному з його елементів None замінюється на об'єкт Котик (Горобчик). Правила розміщення тварин на каруселі:

- разом можуть сидіти тільки котики або горобчики.
- якщо пара сидінь вже зайнята то котик (горобчик) шукає вільне місце
- якщо там вже сидить горобчик (котик) то котик (горобчик) шукає вільне місце.

На кожному кроці, прилітає ще випадкова кількість горобчиків і їх потрібно також розмістити на каруселі. Якщо на каруселі вже немає вільних місць то повинен збуджуватися відповідний виняток. Якщо котик (горобчик) не можуть знайти вільне місце то повинен збуджуватися відповідний виняток.

ЗАДАЧА 3

Реалізувати метод concatenate(Q2) в класі LinkedQueue. Цей метод повинен додати всі елементи черги Q2, яка є об'єктом LinkedQueue в кінець черги. Після цієї операції черга Q2 повинна залишитися пустою. (При вирішенні цієї задачі потрібно використовувати надану реалізацію черги).