

Лабораторная работа № 3. Вариант 2020-1

Написать программу на языке C++, моделирующую поведение горнопроходческого комбайна (RoadHeader) путём создания соответствующего класса.

Каждый комбайн имеет уникальную метку (label) и номинальную производительность (performance, $p \in [1; 10]$ м/час). Состояние комбайна определяется уровнем заточки инструмента (sharpening, $s \in [0; 1]$). Изначально заточка максимальная ($s = 1$). Рабочее время измеряется в целых часах, при этом комбайн может или работать (work), или проходить техобслуживание (maintenance).

Когда комбайн работает, то за час он проходит расстояние

$$dl = p \cdot s,$$

а уровень заточки падает на величину

$$ds = s \cdot k_1.$$

За час техобслуживание координата комбайна не изменяется, а уровень заточки возрастает на величину

$$ds = (1 - s) \cdot k_2.$$

Величины k_1 и k_2 ($k_i \in (0; 1)$) одинаковы для всех комбайнов, но могут быть изменены, например, в результате разработки новых методик проходки и обслуживания.

В функции *main* провести тестирование работоспособности разработанного класса. Использование объектов данного класса (в том числе передача/приём по значению) не должно приводить к утечке памяти, неопределённому поведению и аварийному завершению программы. Длина имени объекта не ограничена. Ресурсами управлять самостоятельно. Обеспечить получение и вывод всей необходимой информации, а также изменение параметров, относящихся к классу в целом. Выбрать разумные величины исходных параметров. Не должно быть очевидных способов изменять состояние объекта минуя его интерфейс. Приветствуется логически обоснованное расширение функционала. Конструкторы перемещения, операторы присваивания и для вывода в поток — по желанию.

Навчальний предмет:

Об'єктно-орієнтоване програмування

Составитель:

Гуда А.И.

Лабораторная работа № 3. Вариант 2020-2

Написать программу на языке C++, моделирующую поведение крестьянина (Peasant) путём создания соответствующего класса.

Каждый крестьянин знает своё имя (name) и свой уровень умений (skill) $s \in (0; 1)$. Текущая работоспособность определяется мотивацией m . Изначально все максимально мотивированы ($m = 1$). Профсоюз крестьян устанавливает количество рабочих часов в сутки (t_w) для всех крестьян.

Крестьянин в каждые сутки может или рубить лес (chop), или заниматься личным огородом (farm).

Если он рубит лес, то выработка за сутки составит

$$dw = t_w \cdot s \cdot m,$$

skill увеличивается на величину

$$ds = k_1 \cdot (1 - s),$$

а мотивация уменьшается вдвое.

Если он занимается личным огородом, то мотивация увеличивается на величину

$$dm = k_2 \cdot (1 - m),$$

а все остальные величины не изменяются.

Величины k_1 и k_2 ($k_i \in (0; 1)$) одинаковы для всех крестьян, но могут быть изменены, например, в результате изменения погоды и прочих факторов.

В функции *main* провести тестирование работоспособности разработанного класса. Использование объектов данного класса (в том числе передача/приём по значению) не должно приводить к утечке памяти, неопределённому поведению и аварийному завершению программы. Длина имени объекта не ограничена. Ресурсами управлять самостоятельно. Обеспечить получение и вывод всей необходимой информации, а также изменение параметров, относящихся к классу в целом. Выбрать разумные величины исходных параметров. Не должно быть очевидных способов изменять состояние объекта минуя его интерфейс. Приветствуется логически обоснованное расширение функционала. Конструкторы перемещения, операторы присваивания и для вывода в поток — по желанию.

Навчальний предмет:
Составитель:

Об'єктно-орієнтоване програмування
Гуда А.И.

Лабораторная работа № 3. Вариант 2020-3

Написать программу на языке C++, моделирующую поведение дворника (Janitor), убирающего снег путём создания соответствующего класса.

Каждый дворник знает своё имя (name) и оборудован личной лопатой определённой вместимости (volume) $v \in (1; 10)$. Изначально каждый дворник бодр, его усталость (fatigue) нулевая ($f = 0$), а температура нормальная ($T = 36.6$). Метеослужба сообщает температуру воздуха на сегодня (T_a).

В течении часа дворник может или мести (sweep), или отдохнуть (rest).

Если дворник метёт, то убирает снег массой

$$dm = v \cdot (1 - f) \exp\left(-\frac{|T - 36.6|}{T_x}\right), T_x \in [2; 10],$$

а усталость увеличивается на величину

$$df = k_1 \cdot v \cdot (1 - f),$$

а температура увеличивается на 2 градуса.

Если дворник отдыхает, то усталость уменьшается в 3 раза, а температура изменяется на величину

$$dT = \frac{T_a - T}{T_y}, T_y = 50.$$

Величины k_1 и T_x ($k_1 \in (0; 1)$) одинаковы для всех дворников, но могут быть изменены, например, в результате применения передовых методов уборки и отдыха.

В функции *main* провести тестирование работоспособности разработанного класса. Использование объектов данного класса (в том числе передача/приём по значению) не должно приводить к утечке памяти, неопределённому поведению и аварийному завершению программы. Длина имени объекта не ограничена. Ресурсами управлять самостоятельно. Обеспечить получение и вывод всей необходимой информации, а также изменение параметров, относящихся к классу в целом. Выбрать разумные величины исходных параметров. Не должно быть очевидных способов изменять состояние объекта минуя его интерфейс. Приветствуется логически обоснованное расширение функционала. Конструкторы перемещения, операторы присваивания и для вывода в поток — по желанию.

Навчальний предмет:

Об'єктно-орієнтоване програмування

Составитель:

Гуда А.И.

Лабораторная работа № 3. Вариант 2020-4

Написать программу на языке C++, моделирующую поведение охотника (Hunter) путём создания соответствующего класса. Каждый охотник знает своё имя и обладает определёнными умениями (skill) $s \in (0; 1)$. Изначально каждый охотник полусыт ($b = 0.5$), и без добычи ($m = 0$). Имеющийся лес характеризуется плотностью добычи ρ .

Охотник каждый час может или охотиться (hunt), или есть (eat).

Если охотник охотится, то масса добытой дичи увеличивается на величину

$$d m = \rho \cdot s \cdot \exp\left(-\frac{(b - 0.5)^2}{0.2}\right),$$

а сытость уменьшается таким образом:

$$d b = -k_1 \cdot b.$$

Если охотник ест, то масса добытого уменьшается максимум на $d m = 2$ кг, но так, что бы $m \geq 0$, а сытость увеличивается на величину

$$d b = k_2 \cdot d m,$$

но не превышает единицу.

Величины k_1 и k_2 ($k_i \in (0; 1)$) одинаковы для всех охотников, но могут быть изменены, например, в результате разработки методов добычи и поглощение еды.

В функции *main* провести тестирование работоспособности разработанного класса. Использование объектов данного класса (в том числе передача/приём по значению) не должно приводить к утечке памяти, неопределённому поведению и аварийному завершению программы. Длина имени объекта не ограничена. Ресурсами управлять самостоятельно. Обеспечить получение и вывод всей необходимой информации, а также изменение параметров, относящихся к классу в целом. Выбрать разумные величины исходных параметров. Не должно быть очевидных способов изменять состояние объекта минуя его интерфейс. Приветствуется логически обоснованное расширение функционала. Конструкторы перемещения, операторы присваивания и для вывода в поток — по желанию.

Лабораторная работа № 3. Вариант 2020-5

Специальный вариант

Написать программу на языке C++, моделирующую поведение пьяницы (Drunk) путём создания соответствующего класса. Каждый любитель выпить имеет уникальное имя (сделать предположение, что моделируемый объект помнит своё имя и может его сообщить), а также свою норму выпитого d_0 (в литрах). Для всех пьяниц данного типа определена длина шага l_0 (в метрах) в состоянии номинального опьянения.

Пьяница прежде всего выпивает (drink) заданное количество пойла d , затем пытается идти, причём каждый шаг (step) происходит в произвольном направлении, а длина шага l описывается формулой:

$$l = l_0 \cdot \exp\left(-\frac{(d - d_0)^2}{10d_0^2}\right),$$

при этом эффективное кол-во выпитого уменьшается на величину

$$dd = 0.05d.$$

Пьяница делает n шагов (задать произвольно) от начала координат, после чего следует вывести его имя, величины d_0 , d , его текущие координаты (x, y) , а также расстояние (r) от него до начальной точки.

В функции *main* провести тестирование работоспособности разработанного класса. Использование объектов данного класса (в том числе передача/приём по значению) не должно приводить к утечке памяти, неопределённому поведению и аварийному завершению программы. Длина имени объекта не ограничена. Ресурсами управлять самостоятельно. Обеспечить получение и вывод всей необходимой информации, а также изменение параметров, относящихся к классу в целом. Выбрать разумные величины исходных параметров. Не должно быть очевидных способов изменять состояние объекта минуя его интерфейс. Приветствуется логически обоснованное расширение функционала. Конструкторы перемещения, операторы присваивания и для вывода в поток — по желанию.
