Національна металургійна академія України

Спеціальність 7.0804 01 – Інформаційні управляючі системи та технології.
Навчальний предмет Об'єктно-орієнтовані мови програмування

Лабораторная работа № 5. Вариант 2010-1.

Написать программу на языке C++, реализующую иерархию классов собирателей водорослей (Collector). Собиратели бывают следующих типов:

Дельфин (Dolphin) Стартовая скорость не зависит от оплаты (её не платят) и составляет 15 кг/час. Коэффициент уставания – 0.1.

Водолаз (Diver) Стартовая скорость кусочно-линейно зависит от оплаты: в диапазоне 0–120 грн пропорциональна оплате, коэффициент пропорциональности – 0.3 кг/грн. При дальнейшем росте оплаты стартовая скорость остаётся постоянной. Коэффициент уставания – 0.3.

Доброволец (Volunteer) Стартовая скорость зависит от оплаты таким образом: $v = 10 \cdot (1 - \exp(-0.2x))$, где v – скорость в кг/час, x – оплата в грн/час. Коэффициент уставания – 0.4.

Киборг (Cyborg) Стартовая скорость не зависит от оплаты и составляет 7 кг/час. Коэффициент уставания – 0.01, но после 4 часов работы киборг выключается из-за истощения топливного элемента. Имя для всех киборгов одно (IC1000).

В программе завести массив (или другой стандартный контейнер по выбору) указателей на собирателей, заполнить указателями на конкретных собирателей разных типов. Для измерения добычи для каждого объекта по указателю вызывается функция **work** без аргументов, возвращающая искомый показатель. Перед работой каждый сборщик получает (или игнорирует) оплату с помощью функции **pay**. После каждого часа работы скорость падает пропорционально коэффициенту уставания. Например, при начальной скорости 10 кг/час и коэффициенте уставания 0.15 в первый час будет собрано 10 кг, во второй – 8.5 кг, в третий – 7.225 кг. и т.д.

Вывести почасовую добычу всех созданных собирателей.

Предоставить возможность определить для каждого типа количество собирателей и полное количество всех собирателей.

Программа не должна завершаться аварийно, приводить к утечке или порче памяти. Объекты любого из созданных классов должны позволять корректную передачу в произвольную функцию по значению, а также возврат по значению из функции.

Національна металургійна академія України

Спеціальність 7.0804 01 – Інформаційні управляючі системи та технології.
Навчальний предмет Об'єктно-орієнтовані мови програмування

Лабораторная работа № 5. Вариант 2010-2.

Написать программу на языке C++, реализующую иерархию классов банковских счетов (BankAccont).

Должны быть смоделировано поведение следующих видов счетов:

Простой вклад Каждый месяц на остаток начисляется постоянный процент, определяемый банком (по умолчанию – 6%).

Шестимесячный вклад В конце каждого шестого месяца на остаток начисляется процент для шестимесячного вклада (по умолчанию – 32%). В случае досрочного снятия любой суммы срок отсчитывается от момента снятия.

Анонимный случайный вклад Каждый месяц на остаток начисляется процент в указанных пределах, (по умолчанию – от 2% до 8%). В качестве имени вкладчика используется порядковый его номер.

Вклад с премией Каждый месяц на остаток начисляется постоянный процент, определяемый банком (по умолчанию – 5%). При этом, если в течении 4 месяцев не было снятия денег, начисляется премия определённого размера (по умолчанию – 3%).

Любой вклад должен позволять довложение средств в произвольный момент времени, снятие средств в пределах остатка, получения справки о состоянии счета. Любой тип вклада должен позволять узнать и изменить свои условия. Любой вклад должен хранить имя вкладчика и номер счета.

Считать (и реализовать в программе), что для единообразного доступа к вкладам существует массив (или другой стандартный контейнер по выбору) указателей на базовый класс, в котором хранятся указатели на реальный объекты-вклады. Для подсчета процентов для каждого указателя в конце месяца вызывается функция **calc** без аргументов. Вклады могут создаваться в разные месяцы, причем каждый сам хранит все необходимую информацию о временных интервалах.

Предусмотреть возможность вывода количества вкладов как каждого типа в отдельности, так и всех вкладов.

Программа не должна завершаться аварийно, приводить к утечке или порче памяти. Объекты любого из созданных классов должны позволять корректную передачу в произвольную функцию по значению, а также возврат по значению из функции.

Національна металургійна академія України

Спеціальність 7.0804 01 – Інформаційні управляючі системи та технології.
Навчальний предмет Об'єктно-орієнтовані мови програмування

Лабораторная работа № 5. Вариант 2010-3.

Написать программу на языке C++, моделирующую иерархию классов животных (Animal), движущихся с известной им цели, находящейся на заданном расстоянии (по умолчанию 100 км). Все животные имеют имена.

В программе завести массив (или другой стандартный контейнер по выбору) указателей на животных, заполнить указателями на конкретных животных разных видов.

В начале каждого часа их кормят (feed), причём каждое животное съедает не более максимально допустимого количества еды для данного вида $m_{\max}$. Потом все животные бегут (run) в течении часа. Текущая скорость животного v зависит от номинальной скорости v_{nom} и усталости f следующим образом: $v = v_{nom} \cdot (1 - f)$. Считать, что усталость больше 1 не бывает.

Виды животных:

Конь (Horse) ест не более 6 кг. Номинальная скорость 22 км/ч. Усталость на следующий час определяется только количеством съеденной пищи m $f = 1 - (m/m_{\max})$.

Крокодил (Crocodile) ест не более 12 кг. Номинальная скорость 6 км/ч. Усталость определяется как накопившейся усталостью f_{old}, так и количеством съеденной пищи: $f = 1 + 0.2 \cdot f_{old} - 0.6 \cdot (m/m_{\max})$.

Верблюд (Camel) ест не более 12 кг. Номинальная скорость 14 км/ч. Усталость определяется только скоростью на предыдущем этапе: $f = 0.4 \cdot (v/v_{nom})$.

БТР (APC) ест не более 150 кг. Номинальная скорость 30 км/ч. Усталость равна нулю, если съел не менее 50 кг, и единице в противном случае. В качестве имени используется строка "БТР-хххх", где хххх – четырёхсимвольное десятичное представление номера БТР по порядку

В начале пути все животные не уставшие ($f = 0$). Для всех животных почасово вывести пройденный путь и текущую усталость. Животных кормить случайным количество пищи от 0.5 до $1.2 \cdot m_{\max}$.

Программа не должна завершаться аварийно, приводить к утечке или порче памяти. Объекты любого из созданных классов должны позволять корректную передачу в произвольную функцию по значению, а также возврат по значению из функции.

Національна металургійна академія України

Спеціальність 7.0804 01 – Інформаційні управляючі системи та технології.
Навчальний предмет Об'єктно-орієнтовані мови програмування

Лабораторная работа № 5. Вариант 2010-4.

Написать программу на языке c++, моделирующую иерархию классов рыбаков (Fisher), добывающих рыбу. Все рыбаки имеют имена. Улов y (yield) определяется удачей $l \in (0; 1)$ (luck) и типом рыбака.

Типы рыбаков:

Любитель (Amateur) Улов прямо пропорционален удаче:

$$y = A \cdot l.$$

Максимальный улов для любого любителя $A = 20$ кг.

Профессионал (Profi) Улов зависит от удачи линейно:

$$y = A \cdot (0.5 + l).$$

Максимальный улов для любого профессионала $A = 50$ кг.

Браконьер (Poacher) Первоначальный улов не зависит от удачи и равен максимальному. Но с вероятностью

$$p = 1 - 0.4 \cdot l$$

весь улов конфискует рыбнадзор. Максимальный улов для любого браконьера $A = 400$ кг.

В программе завести массив (или другой стандартный контейнер по выбору) указателей на рыбаков, заполнить указателями на конкретных рыбаков разных типов. Каждый рыбак выходит на рыбалку 5 раз. Для каждого рыбака и каждой рыбалки сгенерировать случайно значение удачи. Подсчитать и вывести для каждого рыбака улов за каждую рыбалку и суммарный улов.

Программа не должна завершаться аварийно, приводить к утечке или порче памяти. Объекты любого из созданных классов должны позволять корректную передачу в произвольную функцию по значению, а также возврат по значению из функции.

Національна металургійна академія України

Спеціальність 7.0804 01 – Інформаційні управляючі системи та технології.
Навчальний предмет Об'єктно-орієнтовані мови програмування

Лабораторная работа № 5. Вариант 2010-5.

Написать программу на языке C++, реализующую иерархию классов, описывающую зависимость тока, проходящего через радиоэлектронный элемент (Element), от приложенного напряжения. У каждого элемента есть обозначение (например: "D7-G", "MLT-0.5" ...). Для снятия вольт-амперной характеристики (ВАХ) на каждый элемент одинаково подают напряжение (voltage), а затем измеряют ток (current).

В программе завести массив (или другой стандартный контейнер по выбору) указателей на элементы, заполнить указателями на элементы различных типов.

Виды элементов:

Резистор Характеризуется сопротивлением R. Протекающий ток определяется законом Ома:

$$I = \frac{U}{R}.$$

Диод Характеризуется максимальным прямым напряжением $U_{\max}$ и максимальным выпрямленным током $I_{\max}$. В первом приближении, при подаче напряжения в прямом направлении, ток зависит следующим образом:

$$I = I_{\max} \left(\frac{U}{U_{\max}} \right)^2.$$

В обратном направлении диод ток практически не пропускает.

Предохранитель Характеризуется сопротивлением R и максимальным током $I_{\max}$. Если ток по модулю не превышает $I_{\max}$, то этот элемент ведет себя как резистор. При превышении величины предельного тока, предохранитель сгорает, и перестаёт проводить ток.

Заполнить контейнер указателями на элементы разных типов, и снять вольт-амперные характреристики.

Программа не должна завершаться аварийно, приводить к утечке или порче памяти. Объекты любого из созданных классов должны позволять корректную передачу в произвольную функцию по значению, а также возврат по значению из функции.
