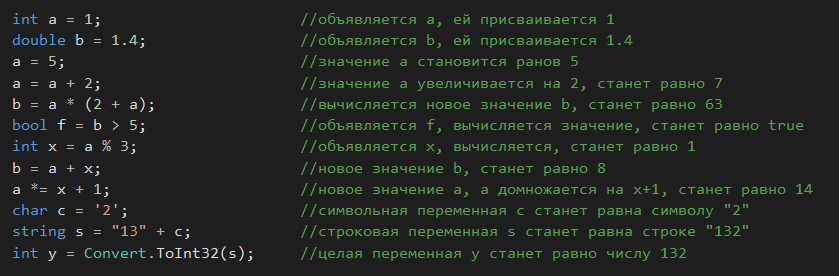
## Лабораторная работа №3. Оператор присваивания

#### Цель работы

Освоение применения оператора присваивания.

#### Справочный материал

Примеры оператора присваивания:



#### Порядок выполнения лабораторной работы

1. Создайте библиотеку классов, соответствующую своему варианту. У вас будут поля класса и соответствующие им свойства. Кроме того, у вас будет как минимум два метода, вычисляющих какое-либо значение или делающих что-то не менее полезное с помощью оператора присваивания. Необходимо четко следовать заданию, и не создавать лишних полей и методов. Типы данных вы должны подобрать самостоятельно (но сделать это логично).
2. Разработайте приложение, которое будет использовать эту библиотеку классов. У пользователя должна быть возможность ввести необходимые данные в текстовые поля и получить результат вычислений. Пока что подразумевается, что пользователь вводит корректные данные. Пользователь должен увидеть отображение всех полученных в ходе вычисления значений. Все вычисления должны производиться методами класса. При реализации приложения вы должны понимать, что какие-то из заданных вами параметров будут меняться от действий пользователя, и эти изменения должны быть отражены. Если для приложения необходима кнопка инициализации объектов, а о ней в задании не сказано, вы можете добавить ее.
3. На данном этапе не используйте условный оператор, даже если вы умеете им пользоваться и им проще. Пользуйтесь для вычислений преобразованиями типов и классом Math.
4. Проверьте работоспособность вашего приложения.
5. Доработайте дизайн приложения, поработав с цветом, графикой и т.п.

#### Примерные контрольные вопросы

1. Покажите, какие типы данных вы выбрали для полей вашего класса. Объясните, почему именно такие?
2. Значения каких типов данных возвращают ваши вычислительные методы? Объясните, почему именно такие?
3. Какие операции, методы библиотек вы использовали при программировании вычислений?
4. Можно ли в вашем вычислительном методе изменить форму записи выражения? Каким образом?
5. Покажите, какие методы вашего класса возвращают какие-либо значения. С помощью какого оператора это делается?
6. Как записать, что метод не должен возвращать никакого значения?
7. Что означает модификатор доступа public, записанный в заголовках ваших методов? Почему этот модификатор не поставлен возле полей класса?
8. Поменяйте работу приложения так, чтобы значения не вводились пользователем, а генерировались случайным образом.
9. 1. Опишите класс «Дракон», позволяющий хранить следующие сведения о драконе: максимально возможное число HP, текущее число HP, максимальный уровень атаки. Методы: нанести удар противнику, получить удар от противника, подлечиться на определенное число HP. Дракон сносит у противника случайное число HP, но в пределах своего максимального уровня атаки.
   2. Напишите приложение, в котором вы реализуете битву двух драконов. Каждый дракон ударяет противника при нажатии возле него кнопки. Однако, нельзя позволять дракону наносить два удара подряд. Также у каждого из драконов должна быть кнопка «подлечиться» (число HP для лечения задается пользователем). Оформите остаток жизни у каждого дракона в виде прогресс-баров.
   3. Добавьте кнопку, которая позволит вывести среднее арифметическое силы нанесенного удара у каждого дракона. Это значение подсчитывается во время каждого из ударов, затем после нажатия кнопки вся статистика сбрасывается.
10. 1. Опишите класс «Рыцарь», позволяющий хранить следующие сведения о рыцаре: максимально возможное число HP, текущее число HP, максимальный уровень атаки. Методы: нанести удар противнику, получить удар от противника, надеть артефакт, снять артефакт. Артефакт увеличивает максимальный уровень атаки рыцаря на некоторое число процентов. Рыцарь сносит у противника случайное число HP, но в пределах своего максимального уровня атаки.
    2. Напишите приложение, в котором вы реализуете сражение двух рыцарей. Каждый рыцарь ударяет противника при нажатии возле него кнопки. Однако, нельзя позволять наносить удары два раза подряд. Также у каждого из рыцарей должны быть кнопки «взять артефакт» (берется только один раз) и «снять артефакт» (возможно только если надет). Оформите остаток жизни у каждого рыцаря в виде прогресс-баров.
    3. Добавьте кнопку, которая позволит определить победителя (укажет номер рыцаря, у которого наибольший остаток HP).
11. 1. Опишите класс «Принцесса», позволяющий хранить следующие сведения о принцессе: имя, вес, рост, возраст. Методы: изменить вес принцессы на указанное число кг, посчитать индекс массы тела принцессы (по формуле Кеттле).
    2. Напишите приложение, в котором вы реализуете наблюдение за весом двух принцесс. Имя каждой принцессы задается пользователем и высвечивается. Нажимается кнопка, и после нажатия возле сведений о каждой из принцесс высвечивается фраза «Принцесса Такая-то изменила свой вес на столько-то килограмм!». Изменение генерируется случайным образом в заданном пользователем диапазоне (задается для каждой принцессы). Отрицательное изменение – похудение, положительное – поправка.
    3. Добавьте кнопку, которая позволит вычислить средний вес обеих принцесс.
12. 1. Опишите класс «Башня», позволяющий хранить следующие сведения о башне: название, максимально возможное число HP, текущее число HP, признак наличия принцессы. Методы: поселить принцессу в башне, убрать принцессу из башни, атаковать башню (снять HP). Число снятых HP зависит от уровня атаки персонажа, однако нельзя, чтобы остаток был меньше 0.
    2. Напишите приложение, в котором вы реализуете существование двух башен. Каждую из башен можно атаковать, туда можно вселить принцессу (по кнопке), забрать принцессу (по кнопке). Нельзя поселить в башню принцессу, если она уже там есть. И нельзя забрать, если ее нет. Оформите остаток HP у каждой из башен в виде прогресс-баров.
    3. Добавьте кнопку, которая позволит определить среднее арифметическое ударов, полученных башней. Данные накапливаются с каждым ударом. После кнопки статистика обнуляется.
13. 1. Опишите класс «Маг», позволяющий хранить следующие сведения о маге: имя, максимальное число HP, текущее число HP, максимальное количество маны, текущее количество маны. Методы: произнести заклинание (снимается часть маны), восстановить часть маны, атаковать (снять с мага часть HP), подлечиться.
    2. Напишите приложение, в котором вы реализуете битву двух магов. Имя каждого мага задается пользователем, все числовые данные – тоже. У каждого мага есть две кнопки для использования заклинания и кнопка для пополнения маны. Одна из кнопок действует на противника – снимает с него HP, другая – на самого мага (лечение). Каждый раз снимается мана (стоимость установите сами в приложении и передавайте методам как параметр). Маги делают свои ходы строго по очереди, пополнение маны также считается ходом. При атаке снимается случайное число HP, но не больше половины от остатка.
    3. Добавьте кнопку, которая позволит выдать число маны, израсходованной каждым магом в течение поединка. После нажатия кнопки статистика обнуляется.
14. 1. Опишите класс «Артефакт», позволяющий хранить информацию об игровом артефакте: название, воздействие на атаку (в процентах, может быть как плюс, так и минус), воздействие на HP (в процентах, плюс или минус), воздействие на защиту (в процентах, плюс или минус), цена. Методы: вернуть цену с учетом скидки (задается параметром в процентах), вернуть воздействие на защиту, вернуть воздействие на HP, вернуть воздействие на уровень атаки. Для последних трех методов собственное значение параметра персонажа задается в качестве параметра.
    2. Напишите приложение, которое позволит вам создать два артефакта и надеть его на персонажа. У персонажа заданы характеристики: HP, атака, защита, сумма денег. У каждого артефакта сделайте набор кнопок: по три кнопки на каждый параметр артефакта. Одна из кнопок увеличивает значение на 1, другая – уменьшает. Третья кнопка фиксирует значение параметра, не позволяя редактировать дальше. Цена вводится с клавиатуры. Добавьте кнопку «Купить» возле каждого из артефактов. После этого параметры персонажа (в том числе и сумма денег) должны измениться. Взять можно только один артефакт, после этого забирать больше артефакты нельзя.
    3. Добавьте кнопку «Очистить», которая вернет параметры персонажа на место и уничтожит оба артефакта.
15. 1. Опишите класс «Денежная сумма», позволяющий хранить информацию о денежном вкладе: числовое значение (предполагается, что в рублях), процент по вкладу. Методы: начислить процент, снять указанную пользователем сумму, положить указанную пользователем сумму.
    2. Напишите приложение, которое позволит вам управлять двумя вкладами. У каждого вклада есть набор кнопок: открыть вклад (возможно только если еще вклад не открыт, при этом кладется на счет указанная сумма), добавить сумму, снять сумму, начислить процент. Действия со вкладом возможны только в том случае, если он открыт.
    3. Добавьте кнопку «Объединить» два вклада. В этом случае все деньги второго вклада объединяются с первым, в качестве процента выбирается наибольший. При этом второй вклад закрывается.
16. 1. Опишите класс «За́мок», позволяющий хранить следующую информацию о замке: название замка, имеющаяся в нем сумма, суточный доход замка, суточный расход на замок. Методы: переключить день (изменить сумму с учетом дохода и расхода), ограбить замок, принести в замок награбленное (добавить деньги).
    2. Напишите приложение, которое позволит вам отразить противостояние двух замков. Названия и все остальные параметры замков задаются пользователем. Есть кнопка «Переключить день» во время которой изменяются денежная сумма в замке. Есть кнопка «Атака», во время которой владельцы замков атакуют (грабят) противника. При этом атака может закончиться любой суммой от 0 до половины всех денег, имеющихся на данный момент в замке. После атаки награбленное относится в свой замок. После ограбления возле каждого замка необходимо вывести сообщение: «Замок такой-то ограблен на столько-то».
    3. Добавьте кнопку, нажав которую, можно узнать, на сколько изменилась сумма денег в замке с начала «отчетного периода». После нажатия этой кнопки начинается новый период.
17. 1. Опишите класс «Набор рун», позволяющий хранить информацию о наборе: число рун в наборе, число бросаемых рун за один бросок, общее число бросков, число совершенных бросков. Методы: бросить руны (изменить число совершенных бросков и число рун в наборе), вернуть все руны обратно (число возвращаемых рун передается как параметр).
    2. Напишите приложение, которое позволит вам выполнять гадание на рунах. У пользователя должна быть возможность ввести все данные о рунах. Должны быть предусмотрены кнопки: сделать бросок (уменьшить число оставшихся бросков, которое всегда должен видеть пользователь), вернуть все руны на место (число возвращаемых рун определяется с помощью числа оставшихся бросков).
    3. Добавьте кнопку, которая позволит узнать, сколько всего бросков было произведено. После нажатия этой кнопки статистика обнуляется.
18. 1. Опишите класс «Судно», позволяющий хранить информацию о судне: название, максимальная грузоподъемность, вес уже имеющегося груза, скорость (без учета ветра, задается в км/ч). Методы: добавить груз, вычислить скорость с учетом скорости ветра (скорость ветра задается как параметр в м/с).
    2. Напишите приложение, которое позволит вам выбрать судно (одно из двух) для того, чтобы быстрее доставить свой груз. Пользователем вводятся параметры первого и второго судна, вводится размер груза и скорость ветра. Возле каждого судна есть кнопка «Загрузить», рядом есть общая кнопка «Проверить». «Загрузить» груз можно только после проверки. После проверки остаются активными только те кнопки, где есть свободное место для груза. Далее для каждого судна вычисляется скорость с учетом скорости ветра. При нажатии кнопки «Загрузить» груз добавляется на выбранное судно.
    3. Добавьте кнопку, которая позволит указать значение скорости, которую будет иметь самый быстрый из кораблей с учетом скорости ветра.
19. 1. Опишите класс «Конь», позволяющий хранить информацию о коне: кличка, максимально возможное число HP, текущее число HP, скорость. Методы: подвергнуть коня атаке (снять HP, снимается случайное число, но не превышающее половины остатка, уменьшить скорость, скорость имеет формульную зависимость от процента HP от максимально возможного), подлечить (добавляется указанное в параметре число HP, меняется скорость).
    2. Напишите приложение, которое позволит владеть двумя лошадьми. У владельца есть определенная сумма денег. Возле каждого из коней есть по кнопке «Использовать» и «Подлечить». После каждого использования коня конь получает повреждения, но при этом владелец зарабатывает деньги (случайное число). После каждого лечения здоровье коня улучшается, однако лечение стоит денег (фиксированная сумма, введенная пользователем). Оформите процент здоровья каждого из коней с помощью прогресс-баров
    3. Добавьте кнопку, которая позволит увидеть наибольшее из значений скоростей коней.
20. 1. Опишите класс «Разбойник», позволяющий хранить информацию о разбойнике: сумма денег у разбойника, максимально возможное число HP, текущее число HP, максимально возможный уровень атаки. Методы: встретиться (при этом разбойник атакует противника, но при этом сам теряет HP в зависимости от уровня атаки противника: от 0 до максимума противника; после атаки разбойник получает деньги по аналогичному принципу; при реализации метода уровень атаки противника и имеющаяся у него сумма денег передаются как параметры), подлечиться (добавить HP, но израсходовать часть денег, данные о числе добавляемых HP и расходуемых денег передаются в метод как параметры).
    2. Напишите приложение, в котором пользователем задаются все необходимые параметры. У разбойника будет кнопка «Напасть», при нажатии которой он будет атаковать противника, данные о котором введены заранее пользователем. Также реализуйте кнопку «Лечить».
    3. Добавьте кнопку, которая позволит узнать средний доход разбойника за все нападения с начала отчетного периода. После нажатия кнопки вся статистика обнуляется, начинается новый период.
21. 1. Опишите класс «Питомец», позволяющий хранить информацию о питомце: имя, настроение (числовое значение в процентах), уровень сытости (проценты), силы (проценты). Методы: приласкать (улучшается настроение и увеличиваются силы), поиграть (улучшается настроение, уменьшаются силы и сытость), покормить (увеличивается сытость и настроение), дрессировать (уменьшаются все параметры). Увеличение/уменьшение – случайная величина от 1 до х, где х – параметр. Для каждой характеристики этот параметр свой в каждом из случаев.
    2. Напишите приложение, в котором реализуете игру-тамагочи с питомцем. Для питомца будут кнопки, соответствующие тому или иному действию. Если хотя бы одна из характеристик питомца обратиться в 0, все кнопки становятся неактивными – питомец умирает.
    3. Добавьте кнопку, которая позволит заново активировать все кнопки и завести нового питомца.
22. 1. Опишите класс «Монстр», позволяющий хранить информацию о монстре: максимальное число HP, текущее число HP, вес монстра. Методы: атаковать, отрезать кусок от монстра (вес монстра уменьшается на случайное число, от 1до 10 процентов его веса; процент может быть дробным, с одним знаком после запятой), восстановить HP (случайное число, но такое, чтобы не превысить максимально допустимое значение).
    2. Напишите приложение, в котором реализуете битву с монстром. Пользователем вводятся все параметры монстра, а также ваш максимальный уровень атаки. У вас должны быть две кнопки: «Атаковать» и «Резать». Атакуя, вы снимаете с монстра случайное число HP, которое может позволить себе ваш персонаж. При этом с вероятностью 1/n (n задается пользователем) после атаки монстр может восстановить некоторое число HP. После нажатия кнопки «Резать» вы отрезаете от монстра кусок мяса, при этом его HP уменьшается на столько процентов, сколько процентов от его веса вы отрезали.
    3. Добавьте кнопку «Продать мясо монстра», при нажатии которой вы сможете узнать, сколько денег вы выручите за все отрезанное мясо (цена задается заранее пользователем). После нажатия этой кнопке мяса у вас уже не остается.
23. 1. Опишите класс «Юнит в войске», позволяющий хранить информацию о юните: максимальный уровень HP, текущий уровень HP, уровень атаки, уровень защиты. Методы: атаковать юнитом (при этом юнит как атакует, так и получает повреждения), подвергнуться атаке (кроме снятия HP, юнит отвечает противнику). Уровень атаки – это урон, который каждый раз наносит юнит во время атаки, уровень защиты – максимально возможный урон, который он наносит, когда атакуют его. Реальный урон может быть меньше (определяется случайным образом в пределах максимально возможного).
    2. Напишите приложение, которое позволит вам реализовать сражение двух разных юнитов. У каждого юнита есть кнопка «Создать» (запоминаются данные юнита) и «Атаковать», во время которой он атакует противника. Юниты атакуют друг друга строго по очереди. Как только у юнита HP окажется равно 0, кнопка атаки должна стать неактивной.
    3. Добавьте кнопку «Очистить», которая восстановит все параметры, которые пользователь задал ранее.
24. 1. Опишите класс «Аптечка», позволяющий хранить информацию об аптечке, которой может воспользоваться персонаж: число HP всей аптечки, число лекарственных средств, стоимость всей аптечки. Методы: определение числа HP в одном средстве (у всех средства аптечки одинаковое), воспользоваться (сколько будет взято лекарственных средств, задается как параметр, стоимость аптечки после использования пропорционально уменьшается), пополнить (задается число средств, которым будет пополнена аптечка, стоимость аптечки при этом пропорционально растет).
    2. Напишите приложение, которое позволит вам реализовать использование аптечки. Все параметры (стоимость, общее число HP, число средств, число добавляемых средств, число средств для использования) задаются пользователем. У аптечки есть кнопки «Использовать» и «Добавить», при этом должно быть реализовано изменение параметров аптечки.
    3. Добавьте кнопку «Очистить», которая восстановит все параметры, которые пользователь задал ранее.
25. 1. Опишите класс «Источник силы», позволяющий хранить информацию об источнике силы: число единиц атаки, на которое поднимется уровень атаки за одно использование, максимально возможное число использований, текущее число использований. Методы: воспользоваться (при этом персонаж получает определенное число единиц атаки, текущее число использований при этом увеличивается), восстановить.
    2. Напишите приложение, в котором персонаж с заданным уровнем атаки будет пользоваться источником силы. Параметры источника вводятся пользователем. Можно пользоваться источником силы только в том случае, если он не пуст. Возле источника есть кнопки «Испить силы» и «Восстановить источник». Степень наполненности источника оформите в виде прогресс-бара.
    3. Добавьте кнопку, которая позволит вывести, сколько раз воспользовались источником за «отчетный период». После нажатия кнопки статистика обнуляется, период начинается снова.
26. 1. Опишите класс «Темница», позволяющий хранить информацию о темнице замка: число камер, число узников, размер камеры (в м2). Методы: обновить число узников (каждый раз случайным образом определяется, сбежали ли узники, и если сбежали, то число человек от 1 до 3), добавить узников (число добавляемых задается как параметр), определить число м2 на одного узника.
    2. Напишите приложение, которое позволит вам вести учет узников в темнице. Пользователь изначально задает все параметры, затем может пользоваться кнопками: «Добавить» (добавить узников), «Обновить». Каждый раз после изменения числа узников вычисляется, сколько м2 приходится на одного узника.
    3. Добавьте кнопку, которая позволит вывести, сколько всего человек было добавлено, и сколько сбежало за отчетный период. После нажатия кнопки статистика обнуляется, период начинается снова.
27. 1. Опишите класс «Лавка артефактов», позволяющий хранить информацию о лавке артефактов: число артефактов, стоимость одного артефакта, сумма денег в лавке. Методы: купить артефакт, продать артефакт, определить уровень скидки на артефакт (данный метод вызывается методом продажи, если владелец лавки согласен сделать скидку; вероятность такого согласия – 0,n, где n – число артефактов в лавке; при этом вероятность округляется с точностью до 2 знаков после запятой; размер скидки определяется случайным образом в диапазоне от 1 до 5 процентов от стоимости товара, процент скидки может быть дробным, дробная часть задается с точностью до одного знака после запятой).
    2. Напишите приложение, которое позволит вам покупать и продавать артефакты в лавке. Пользователем задаются все необходимые параметры. У лавки есть кнопки «Купить» и «Продать». Продать артефакт можно только в том случае, если в лавке достаточно для этого денег.
    3. Добавьте кнопку, которая позволит вывести, сколько раз предоставлялась скидка за «отчетный период». После нажатия кнопки статистика обнуляется, период начинается снова.
28. 1. Опишите класс «Таверна», позволяющий хранить информацию о таверне: число мест, число посетителей, число денег в кассе. Методы: определить число свободных мест, зайти в таверну компанией в n человек (число n передается как параметр, при этом каждый оплачивает стоимость посещения, которая тоже передается как параметр), ограбить (взять случайное число денег, не превышающее содержимое кассы).
    2. Напишите приложение, которое позволит вам то посещать, то грабить таверну. Все параметры заранее вводятся пользователем. Есть кнопки «Открыть» (инициализация всех данных), «Зайти» (активна только в том случае, если число свободных мест соответствует числу человек в компании), «Ограбить».
    3. Добавьте кнопку, которая рассчитает, сколько приходится денег на каждого из компании с учетом награбленного за весь «отчетный период». После нажатия кнопки статистика обнуляется, период начинается снова.
29. 1. Опишите класс «Местность на карте», позволяющий хранить информацию о местности: тип местности, координаты левого верхнего и правого нижнего углов (местность является прямоугольной), коэффициент скорости (может как увеличивать скорость, так и уменьшать). Методы: вычислить площадь местности, определить скорость персонажа на данной местности (скорость передается как параметр), определить длину пути через местность (предполагается, что движение идет по основной диагонали из угла в угол).
    2. Напишите приложение, которое позволит вам реализовать проход по двум разным участкам местности. Все параметры задаются пользователем. Предполагается, что расстояние в одну координатную единицу – это 1 км. При нажатии кнопки возле каждой из местностей необходимо рассчитать площадь местности и время, которое затратит персонаж на прохождение через нее (с учетом заданной скорости).
    3. Добавьте кнопку, которая рассчитает, сколько всего времени затратит персонаж на проход по двум участкам местности.
30. 1. Опишите класс «Дерево», позволяющее хранить информацию об игровой единице «дерево»: название, число ударов для рубки, число оставшихся ударов (дерево могли уже рубить), число плодов на дереве. Методы: рубить (уменьшить число оставшихся ударов), потрясти (получить случайное число плодов, не превышающее число имеющихся), полить (добавить число оставшихся ударов), удобрить (добавить число оставшихся ударов и количество плодов (от 10 до 50 процентов от уже имеющихся)).
    2. Напишите приложение, реализующее взаимодействие с деревом. Все начальные параметры вводятся пользователем, после чего нажимается кнопка «Создать» (дерево). Далее пользователь может нажимать кнопки «рубить», «потрясти» (активна только если есть плоды), «полить» (активна только если дерево не максимального уровня), «удобрить». Если дерево срублено, то все кнопки становятся неактивными. Степень срубленности дерева оформить с помощью прогресс-бара.
    3. Добавьте кнопку, которая рассчитает, сколько всего получено плодов с дерева. После нажатия кнопки статистика обнуляется.
31. 1. Опишите класс «Лабиринт», позволяющий хранить информацию о лабиринте: число уровней, число добавляемых на каждом уровне препятствий, базовое число очков. Методы: рассчитать число препятствий для уровня, рассчитать число очков для уровня (зависит от базы, общего числа препятствий).
    2. Напишите приложение, реализующее анализ двух лабиринтов. Все начальные параметры задаются пользователем. Необходимо показать для введенного пользователем уровня число препятствий и число очков для каждого лабиринта.
    3. Добавьте кнопку, после нажатия которой будет выдан номер лабиринта с максимальным числом очков на заданном уровне.
32. 1. Опишите класс «Напиток» (как игровой объект), позволяющий хранить информацию о напитке: воздействие на HP (числовое значение, отрицательное или положительное, в процентах), воздействие на скорость персонажа (аналогично HP), воздействие на уровень атаки (аналогично HP), цена. Методы: вернуть результирующий HP персонажа (базовый задан как параметр), аналогично вернуть итоговую скорость, итоговую атаку.
    2. Напишите приложение, которое реализует употребление двух типов напитков (все параметры задаются пользователем). Кроме того, у персонажа есть определенная сумма денег (задается пользователем). У каждого напитка есть кнопка инициализации, кнопки «Купить» и «Выпить». Число имеющихся напитков должны быть показаны, меняются при покупке и при использовании. При этом параметры персонажа и имеющиеся деньги меняются. Если денег у персонажа недостаточно, то кнопка «Купить» блокируется. Аналогично невозможно выпить напиток, если его нет.
    3. Добавьте кнопку, которая покажет, сколько денег было потрачено персонажем от начала игры.
33. 1. Опишите класс «Ресурсная шахта», позволяющий хранить информацию о шахте: название ресурса, количество ресурса, добываемого за сутки одним рабочим, число рабочих. Методы: рассчитать количество ресурса за сутки, определить прибыль от шахты за сутки (зарплата одному рабочему и стоимость ресурса на рынке задаются как параметры), нанять рабочих (число нанимаемых рабочих задается как параметр), уволить рабочих.
    2. Напишите приложение, которое реализует владение ресурсной шахтой. Все параметры при этом задаются пользователем. У владельца есть некоторая сумма денег, также заданная пользователем. У шахты есть кнопки: «Нанять рабочих», «Уволить рабочих», «Переключить сутки».
    3. Добавьте кнопку, которая покажет, сколько всего денег было заработано на шахте с начала игры.
34. 1. Опишите класс «Книга», позволяющий хранить информацию о книге: название книги, число страниц, число заполненных страниц. Методы: дописать (число страниц для записи задается как параметр), определить степень заполненности книги (в процентах), уничтожить книгу (стереть все страницы).
    2. Напишите приложение, которое реализует взаимодействие с книгой. Все параметры при этом задаются пользователем. У книги есть кнопки «Инициализировать», «Дописать», «Уничтожить». Нельзя ничего дописать в книгу, если она уже полностью заполнена. Степень заполненности оформить в виде прогресс-бара.
    3. Добавьте кнопку, которая покажет, сколько всего страниц написано с начала игры.
35. 1. Опишите класс «Деревня», позволяющий хранить информацию о принадлежащей персонажу деревне: название деревни, число домов, доход с одного дома за сутки. Методы: определить доход от деревни за сутки, напасть на деревню (забрать часть возможного дохода и уничтожить часть домов, часть – определяется случайным образом в пределах возможного).
    2. Напишите приложение, которое реализует взаимодействие с двумя деревнями. Первоначальные данные о деревнях вводятся пользователем. Отображается сумма денег, имеющаяся в деревне на данный момент (задается пользователем). Предполагается, что первая деревня принадлежит самому игроку. Он не может на нее нападать, но может собрать доход (не весь, указывается конкретная сумма). На вторую он может нападать и собирать часть дохода.
    3. Добавьте кнопку, которая дает возможность пройти суткам. За это время добавляется случайное число домов в каждую деревню и к сумме денег добавляется суточный доход.
36. 1. Опишите класс «Университет», позволяющий хранить информацию об университете, в котором персонаж может прокачивать навыки: прокачиваемый навык, степень увеличения навыка персонажа за один «курс» (в процентах к существующему уровню), стоимость курса. Методы: определить значение навыка персонажа после изучения курса (текущий уровень передается как параметр), вернуть стоимость курса со скидкой (будет ли скидка, определяется случайным образом с вероятностью ½), размер скидки определяется как нецелое случайное число от 5 до 20 с точностью до одного знака после запятой (это – процент скидки).
    2. Напишите приложение, которое реализует прокачку навыка персонажа в университете. Для университета заданы: название навыка, стоимость курса, текущая сумма денег на счету университета. Для персонажа заданы: имя, текущая сумма денег, текущее значение навыка. Все данные вводятся пользователем, после чего необходимо нажать кнопку инициализации (имеется как для персонажа, так и для университета). Название навыка персонажа зависит от того, какой навык указан в университете. Кнопка, позволяющая учиться, должна быть активна только в том случае, если у персонажа достаточно денег, чтобы обучиться без скидки. После обучения меняются данные: сумма денег у персонажа, уровень навыка у персонажа, сумма на счету университета.
    3. Добавьте копку, после нажатия которой будет показано: «Персонаж такой-то повысил свой навык такой-то на столько-то». Значение указывается не в процентах, а в единицах навыка.
37. 1. Опишите класс «Здание», которое хранит информацию о построенном в городе игрока здании: уровень здания, расходы на содержание за сутки на один уровень, доход со здания за сутки на один уровень. Подразумевается, что с повышением уровня значение дохода/расхода будет увеличиваться. Методы: вычисление суммы денег за сутки, которую игрок получит от здания (в случае, если здание «дотационное», эта сумма окажется отрицательной), определение того, является ли здание доходным (т.е., его доход больше 0), является ли здание дотационным (доход меньше 0).
    2. Напишите приложение, которое реализует взаимодействие со зданием. Информация о здании: уровень, доход с одного уровня, расход с одного уровня. После нажатия кнопки инициализации (одна для обоих объектов) должны быть показаны доход и расход за сутки. Для игрока показана имеющаяся сумма денег. Все данные для инициализации вводятся пользователем. Должна быть кнопка прохождения суток, после которой имеющаяся у игрока сумма поменяется. Кнопка активна только в том случае, если у игрока достаточно для этого денег. Также реализуйте кнопку «Повысить уровень», после чего уровень здания увеличится на 1.
    3. Добавьте кнопку, после нажатия которой будет выдано True, если здание приносит доход, False, если не приносит, и ничего, если здание не является ни дотационным, ни доходным.
38. 1. Опишите класс «Турнир», которое хранит информацию о рыцарском турнире: число участников, число денег, необходимое для выплаты призовых, сумма, выданная для выплаты призовых спонсорами. Методы: подсчитать, какой процент обеспечен спонсорами, подсчитать размер взноса с участника (в случае, если денег спонсоров не хватает, недостающую сумму оплачивают участники как взнос для участия в турнире).
    2. Напишите приложение, которое реализует запись рыцаря на турнир. Данные о турнире и рыцаре (имя, сумма денег) введены пользователем. Кроме вводимых данных, показан номер турнира (до объявления каких-либо турниров это 0). Рыцарь может записаться на турнир, нажав специальную кнопку, которая активна только в том случае, если у него для этого достаточно денег и если он еще не записывался на этот турнир. При расчете следует учесть, что рыцарь будет новым добавляемым участником турнира (т.е., после записи число участников будет больше). Должна быть также кнопка «Объявить турнир», после нажатия которой будут показаны все вычисляемые данные о турнире а также его номер (каждый раз увеличивается на 1).
    3. Добавьте кнопку, после нажатия которой будет показано: «Рыцарь такой-то записан на столько-то турниров».
39. 1. Опишите класс «Блюдо», которое хранит информацию о блюде: название блюда, его вес, калорийность, стоимость. Методы: пересчитать калорийность для другого веса (новый вес задается как параметр), пересчитать новую стоимость блюда для другого веса, подсчитать, сколько килограммов добавит блюдо к весу того, кто его съест (вес блюда задается как параметр).
    2. Реализуйте приложение, которое позволяет отобразить поправку и похудение игрока. Данные по блюду вводятся пользователем, вес персонажа и сколько веса снимает «упражнение» – тоже вводится пользователем. У игрока есть кнопка «Есть» (предварительно он должен ввести вес поедаемого блюда), после чего его килограммы пересчитываются. Также есть кнопка «Худеть», которая предполагает, что персонаж выполняет упражнение, которое уменьшит часть его веса.
    3. Добавьте кнопку, после нажатия которой будет показано, какую сумму денег игрок израсходовал на еду.
40. 1. Опишите класс «Клан», который хранит информацию об игровом клане: название клана, число игроков в клане, общий уровень атаки клана. Методы: добавить игрока (уровень его атаки задается как параметр), удалить игрока (уровень атаки задается как параметр), подсчитать среднее арифметическое уровня атаки игроков.
    2. Реализуйте приложение, которое показывает данные о клане (изначальные данные вводятся пользователем) и позволяет добавлять или удалять игроков с заданными характеристиками (характеристики задаются пользователем). Для этого реализуйте кнопки «Вступить в клан» и «Выйти из клана» (в случае, если игроков в клане нет, эта кнопка должна быть неактивна).
    3. Добавьте кнопку «Статистика», после нажатия которой будет выдано, сколько игроков вступило в клан и вышло из клана за «отчетный период». После нажатия кнопки статистика обнуляется.
41. 1. Опишите класс «Страна», который хранит информацию о стране: название, население, площадь (в км2). Методы: подсчитать число человек на 1 км2, подсчитать, сколько км2 приходится на одного человека, поучаствовать в войне (при этом передается как параметр, на сколько при этом изменяется численность населения и площадь).
    2. Напишите приложение, которое реализовывает модель существования двух стран (изначальные данные вводятся пользователем). Реализуйте кнопку инициализации данных и кнопку «Воевать». Победитель определяется случайным образом. При этом убыль населения у каждой страны определяется случайным образом. Победителю прибавляется часть территории проигравшего (вычисляется случайным образом), при этом меняются данные о площади стран и численности населения (предполагается, что населения распределено по территории равномерно).
    3. Добавьте кнопку «Статистика», после нажатия которой будет выдано среднее арифметическое числа погибших за все прошедшие с начала игры войны.
42. 1. Опишите класс «Город», который хранит информацию о городе, который собирается посетить путешественник: название, численность населения, площадь, число достопримечательностей, стоимость проживания за день. Методы: подсчитать число человек на 1 км2, подсчитать число дней на посещение города (число достопримечательностей, которое может посетить путешественник за день, задается как параметр).
    2. Напишите приложение, которое реализует путешествие туриста. Задаются данные о городе и о туристе (имя, сумма денег, сколько достопримечательностей посещает за день). Есть кнопка «Новый город», после нажатия которой запоминаются поля, введенные ранее пользователем, и выводятся значения вычисляемых данных. Есть кнопка «Посетить», после нажатия которой с туриста снимается требуемая сумма денег. Эта кнопка активна только в том случае, если у туриста достаточно денег для посещения.
    3. Добавьте кнопку «Статистика», после нажатия которой будет показана средняя сумма денег, которую турист тратил в каждом городе.
43. 1. Опишите класс «Достопримечательность», который хранит информацию о достопримечательности города: название, год появления, оптимальное время на посещения, стоимость посещения. Методы: вычислить возраст, сформировать строку с информацией о достопримечательности в нескольких строках: первая строка – название, вторая – оптимальное время на посещение, третья - стоимость.
    2. Напишите приложение, которое реализует составление маршрута по достопримечательностям. Информация о достопримечательности вводится пользователем. Пользователем нажимается кнопка «Добавить в маршрут», и в список достопримечательностей добавляется новая со всей информацией.
    3. Добавьте кнопку «Максимальный возраст», после нажатия которой будет выдан возраст самой древней достопримечательности из выбранных.
44. 1. Опишите класс «Планета», который хранит информацию о планете: название, масса, радиус, длительность суток в земных часах. Методы: посчитать объем, посчитать плотность, посчитать площадь поверхности (предполагается, что планета представляет собой шар), посчитать, сколько земных суток пройдет на планете за указанное земное время (это время передается как параметр).
    2. Напишите приложение, которое будет рассчитывать информацию о двух планетах. Все исходные данные при этом вводятся пользователем. Расчет производится по кнопке «Рассчитать».
    3. Добавьте кнопку, по которой будут рассчитываться среднее арифметическое масс, объема и площади поверхности планет.
45. 1. Опишите класс «Инопланетянин», который хранит информацию об инопланетянине: имя, планета, минимально возможный рост, максимально возможный рост, рост данного существа. Методы: вычислить отклонение от среднего роста (среднее между максимумом и минимумом), изменить вес на заданное отклонение (изменяется только в возможных границах, отклонение задается как параметр).
    2. Напишите приложение, которое будет рассчитывать информацию о двух инопланетянах: необходимо показать отклонение от среднего. Остальные данные вводятся пользователем. Реализовать кнопку изменения веса. При этом вес должен меняться на случайное число в заданных пользователем пределах (не совпадает с пределами веса, заданными для инопланетян). После изменения веса выдается реальная разница (если, например, вес оказывался бы вне возможных границ).
    3. Добавьте кнопку, по которой будет рассчитываться среднее реальное отклонение от веса для каждого из инопланетян.
46. 1. Опишите класс «Космический корабль», который хранит информацию о космическом корабле: название, скорость, число дней от последнего ТО, число отсеков. Методы: определить время пути в днях (расстояние задается как параметр), определить, сколько дней осталось до ТО (периодичность ТО задается как параметр), определить, возможен ли полет (если до ТО осталось меньше дней, чем время полета, то полет невозможен).
    2. Напишите приложение, которое будет реализовывать отправку космического корабля в полет. Все необходимые параметры вводятся пользователем, после чего нажимается кнопка инициализации. Кнопка отправки в полет активна только в том случае, если число дней полета не превышает число дней до ТО. Реализуйте также кнопку «Пройти ТО». В этом случае число дней от последнего ТО обнуляется.
    3. Добавьте кнопку «Статистика», которая выдаст общее расстояние, которое преодолел корабль, и его общее число дней в полете.
47. 1. Опишите класс «Космонавт», который хранит информацию о космонавте: имя, возраст, число полетов, число проведенных в полетах дней. Методы: перевести число дней полетов в часы, посчитать сколько часов длился каждый полет (усредненное значение), добавить полет.
    2. Напишите приложение, которое будет реализовывать отправку космонавта в полет. Данные о полете и начальные данные о космонавте вводятся пользователем. Для отправки в полет необходимо нажать кнопку, после которой число полетов увеличится, добавится число дней в полете, изменится усредненное значение.
    3. Добавьте кнопку, по которой будет выдана информация о том, сколько часов провел космонавт в полете.
48. 1. Опишите класс «Комета», который будет хранить информацию о комете: название, скорость, частота появления (сколько раз за 10 лет приближается к Земле на минимальное расстояние), минимальное расстояние до Земли. Методы: рассчитать, сколько осталось до минимального расстояния (текущее состояние задается как параметр), рассчитать, за какое время комета достигнет минимального расстояния.
    2. Напишите приложение, которое будет реализовывать расчет параметров двух комет. Все необходимые исходные данные вводятся пользователем. Необходимо после кнопки «Рассчитать» выдать информацию о том, за сколько какая комета приблизится к Земле, выдать номер кометы, которая приблизится раньше.
    3. Добавьте кнопку, которая будет показывать номер кометы, которая чаще приближается к земле.
49. 1. Опишите класс «Организм», который будет хранить информацию об организме, живущем где-то на просторах Вселенной: название, вес, объем. Методы: дать новое название, вычислить плотность, увеличить вес на указанное число процентов (процент задается как параметр).
    2. Напишите приложение, которое будет реализовывать контакт с этим организмом. Все необходимые исходные данные задаются пользователем и запоминаются по кнопке инициализации. Также есть специальные кнопки переименования и изменения веса. При этом плотность должна пересчитываться. До инициализации «игры» прочие кнопки нажимать невозможно.
    3. Добавьте кнопку, по которой будет показано: «Организм такой-то изменил свой вес на столько-то» (результат указывается не в процентах, а в единицах измерения веса).
50. 1. Опишите класс «Наперсточник», хранящий информацию о наперсточнике: имя, число предлагаемых в игре «наперстков», номер «наперстка» с шариком, вероятность сжульничать (задается с двумя знаками после запятой). Методы: вычислить вероятность отгадать где шарик, спрятать шарик (номер наперстка вычисляется случайно), реализовать жульничество (перепрятать наперсток, если наперсточник решил сжульничать).
    2. Напишите приложение, которое будет реализовывать игру с наперсточником. Изначальные данные (число наперстков, вероятность жульничества и т.д.) вводятся пользователем и сохраняются после нажатия кнопки инициализации. До нажатия этой кнопки игра невозможна. Пользователь указывает номер наперстка и нажимает «Играть». Наперсточник либо прячет шарик один раз, либо перепрятывает (если выпало по вероятности). Если номер наперстка, указанный пользователем, совпал с тем, под каким сейчас шарик, то выдать True, иначе выдать False.
    3. Добавьте кнопку «Статистика», после нажатия которой будет показано, сколько раз пользователь у наперсточника выиграл, сколько проиграл.
51. 1. Опишите класс «Межзвездный маршрут», хранящий информацию о маршруте: число остановок, расстояние между остановками (предполагается, что оно одинаково), число населенных планет среди остановок. Методы: подсчитать процент населенных планет среди остановок, подсчитать общее расстояние, подсчитать, за какое время маршрут пройдет корабль с заданной скоростью (скорость задается через параметр).
    2. Напишите приложение, в котором можно задавать параметры маршрута. Возле параметров «число остановок» и «число населенных планет» сделайте три кнопки: увеличить на 1, уменьшить на 1, зафиксировать (нельзя уменьшить на 1 число 1, нельзя задать число населенных планет больше, чем число остановок). Расстояние между остановками задается в млн.км и в дробных частях с точностью до двух знаков после запятой. Целая и дробная часть могут увеличиваться отдельно (по кнопке добавить 1 и отнять 1 возле каждой части). При этом дробная после 99 становится равной 00, меньше 0 сделать целую часть нельзя.
    3. Добавьте кнопку, после нажатия которой будет выведено время, которое будет затрачено на весь маршрут и процент планет с жизнью.
52. 1. Опишите класс «Сообщение», хранящий информацию о сообщении: общее число символов, число различных символов. Методы: рассчитать, какое минимальное число бит понадобится для кодирования одного символа, рассчитать, сколько целых байт понадобится для кодировании всего сообщения, если каждый символ кодируется минимально возможным числом бит.
    2. Напишите приложение, которое по параметрам сообщения будет высчитывать объем закодированного сообщения (в байтах). Приложение должно работать для двух сообщений.
    3. Добавьте кнопку, по нажатию которой будет выводиться номер сообщения, которое займет наибольший объем памяти.
53. 1. Опишите класс «Игра», хранящий информацию об игре «Виселица»: число букв, число отгаданных букв, допустимое число ошибок, сколько сделано в ошибок. Методы: посчитать число ошибок, которое еще может сделать игрок, не оказавшись «повешенным», подсчитать процент ошибок в данном сеансе (общее число сделанных ходов передается как параметр).
    2. Напишите приложение, которое по исходным параметрам игры, введенным пользователем, высчитывает для двух игроков процент ошибок и процент удачных ходов в игре, а также сколько букв осталось отгадать, сколько ошибок может еще сделать каждый из игроков.
    3. Добавьте кнопку, по нажатию которой будет выдаваться номер игрока, у которого больше шансов выиграть (сумма оставшихся возможных ходов и не отгаданных букв минимальна).
54. 1. Опишите класс «Игра», хранящий информацию об игре угадывания слова по буквам из набора: изначально заданное число букв, имеющееся на текущий момент число букв, число букв в слове, число уже отгаданных букв. Методы: посчитать число неразгаданных букв, подсчитать процент отгаданных букв, подсчитать процент израсходованных букв.
    2. Напишите приложение, которое по исходным параметрам игры (число букв – в соответствии с полями класса), введенным пользователем, высчитывает для игрока все остальные параметры, которые можно вычислить с помощью методов.
    3. Добавьте кнопку, по нажатию которой будут генерироваться новые входные параметры (случайным образом).
55. 1. Опишите класс «Рецепт», хранящий информацию о трех ингредиентах, которые входят в рецепт: вес первого, вес второго, вес третьего. Методы: пересчитать вес всех ингредиентов по заданному весу первого (новый вес задается как параметр), аналогично по методу для второго и для третьего ингредиента, подсчитать общий вес блюда.
    2. Напишите приложение, которое будет давать возможность задавать рецепт из трех ингредиентов и пересчитывать его составляющие по нажатию кнопки. При этом должна быть возможность указать, какой ингредиент поменял свой вес в блюде.
    3. Добавьте кнопку, по нажатию которой будет рассчитываться общий вес блюда.
56. 1. Опишите класс «Загадчик», хранящий информацию о персонаже, загадывающем загадки игроку: имя, число загадок в серии загадок, число правильно отгаданных загадок, число загаданных загадок. Методы: вычислить, сколько осталось загадать загадок, определить процент правильно отгаданных загадок, подсчитать число баллов игрока (число баллов за одну отгаданную загадку задается как параметр).
    2. Напишите приложение, реализующее статистику отгадывания загадок (самих загадок пока нет). Пользователем вводится общее число загадок в серии. Затем пользовать может нажать кнопку «Отгадать», после чего будут изменены данные: число отгаданных загадок, число оставшихся загадок, число баллов, а также выданные остальные сведения, получаемые методами класса. При этом отгадал игрок загадку или нет, вычисляется случайным образом с вероятностью ½. Отгадывать можно только тогда, когда еще есть загадки.
    3. Добавьте кнопку, которая выдаст, сколько ответов из скольких было отвечено правильно.