

Контрольная работа #2

Вам нужно написать программу-игру, которая имитирует битву(перестрелку) межгалактического героя и робота-террориста. Информация о ходе битвы должна выводиться в консоль. Программа пишется постепенно, каждый этап добавляет новые функции в ту же программу. Задание считается выполненным на 100%, если выполнены условия всех уровней (кроме бонуса). Бонус засчитывается, только если решены остальные этапы, поэтому добейтесь сначала работоспособности основного кода, а потом беритесь за бонусы, если время осталось. Все **указания на этапах** являются **обязательными**.

Исходные данные:

```
robot = {
    'hp': 1300,      # жизненная энергия, запас здоровья
    'defence': 120,  # защита, броня
    'gun': 300       # оружие
}
hero = {
    'hp': 2000,
    'defence': 100,
    'gun': 250,
    'protective_field': 150    # защитное поле
}
```

Исходные данные можно менять по своему желанию, чтобы проверить **разные** варианты работы программы. **При проверке значения также могут и будут меняться (!)**.

Общие указания:

- Игровой цикл реализуется с помощью бесконечного цикла
- Сообщения в консоли должны быть максимально информативны. Они могут содержать информацию о количестве НР у робота и героя, номер хода, кто ходит, кто сколько урона нанёс и т.д.
- Можно использовать разделители (строки типа "=====") и дополнительные переносы строк ("\n"), чтобы было понятно, когда закончился один ход и начался следующий.
- Аккуратное форматирование вывода приветствуется.

Этап 1

На первом этапе вам достаточно написать главный игровой цикл, за который производятся только ходы игрока. Каждый ход игрока выполняется по следующим правилам:

- Герой пытается попасть по роботу вероятность 75%:
 - Если попал
 - Урон = оружие героя - защита робота
 - У робота отнимается здоровье, равное урону
 - Отображается сообщение о нанесенном уроне.
 - Если не попал:
 - Отображается сообщение о том, что герой не попал
- В конце хода вывести сообщение об оставшемся здоровье **робота**.

Вероятность можно определять с помощью `random`-а, например:

```
hit_probability = random.randint(1, 100)

if hit_probability <= 30: ...    # 30%

else: ...    # 70%
```

Главный цикл должен повторять ходы героя до тех пор, пока у робота не останется 0 хп.

- Если у робота больше нет жизненной энергии:
 - Показать сообщение, что герой победил
 - Выйти из цикла.

Указания:

- Для вывода информации о роботе напишите отдельную функцию `display_robot_info()`. На первом этапе достаточно выводить его здоровье.
- Напишите функцию `modify_health()`, которая принимает два аргумента:
 - Персонажа, у которого нужно изменить показатель здоровья
 - Количество жизненной энергии, на которое нужно изменить этот показатель. Внутри себя функция всегда **прибавляет** этот показатель. Таким образом, если нужно уменьшить показатель здоровья, то в функцию нужно передать **отрицательное** число. Функция должна следить, чтобы показатель здоровья **не стал меньше 0**, даже если нанесённый урон больше оставшегося здоровья.

Функция отвечает **только** за изменение показателей здоровья персонажей и не должна выводить никаких сообщений.

- В сообщениях о нанесенном уроне урон должен выводиться **без знака минус**.

Пример работы функции:

```
some_dmg = -50          # для примера
modify_health(robot, some_dmg) # нанести 50 ед урона роботу
```

При выводе сообщения:

```
"Робот получил 50 ед. урона"    # без знака минус
```

Этап 2

На этом этапе вам нужно добавить ход робота. Ход робота выполняется после хода игрока и проверки на то, что робот ещё живой (hp робота больше 0). Робот может с шансом 1/3 выбрать одно из действий:

- Использовать самонаводящиеся ракеты и точно попасть по герою:
 - Урон = оружие робота + 30% от оружия робота - защита героя
 - Нанесённый урон вычитается из здоровья героя
 - Вывести сообщение о нанесенном уроне
- Использовать обычные патроны:
 - С вероятностью 50/50 робот попадает по герою
 - Если попал:
 - Урон = оружие робота - защита героя
 - Нанесённый урон вычитается из здоровья героя
 - Вывести сообщение о нанесенном уроне
 - Если не попал:
 - Отображается сообщение о том, что робот не попал
- Заклинить:
 - Вывести сообщение о том, что робот заклинил.
- В конце хода вывести сообщение об оставшемся здоровье **героя**.

После хода робота в основном цикле нужно проверять количество оставшегося здоровья у героя.

- Если у героя закончилась жизненная энергия:
 - Показать сообщение, что герой проиграл
 - Выйти из цикла.

Можно выводить сообщения о состоянии обоих персонажей: и робота, и героя, после каждого хода.

Указания:

- Для вывода информации о герое напишите отдельную функцию `display_hero_info()`
- Для вывода информации о роботе используйте функцию `display_robot_info()` из первого этапа.
- **Вынесите ходы робота и героя в отдельные функции.**
- По желанию вынесите проверку количества HP у персонажа (ещё жив или нет) в отдельную функцию.
- Используйте функцию `modify_health()`, написанную на 1 этапе, для нанесения урона герою.

В конце этого этапа главный цикл может выглядеть примерно так:

```
while True:
    hero_turn()    # ход героя
    # если робот умер: break
    robot_turn()   # ход робота
    # если герой умер: break
```

Этап 3

Во время хода героя добавьте возможность для игрока выбирать, что герой будет делать:

- Атаковать - в этом случае выполняется атака героя, как в 1 этапе.
- Защищаться
 - Если герой защищается, то значение защитного поля (`protective_field`) героя прибавляется к его защите (`defence`) до конца хода (на 1 ход).
 - В этот ход герой не атакует.

- После следующего хода робота герой должен снять защитное поле.
- Ничего не делать, пропустить ход - герой не атакует и не защищается.

Выбор совершается путем ввода одной из команд: `attack`, `pass`, `defence`.

Указания:

- Для каждого действия напишите отдельную функцию.
- Для каждого действия выводите сообщение о нём.
- В функции, отвечающей за ход героя, оставьте только выбор действия и ввод команды.
- Для реализации защиты напишите функции `equip_shield()` и `remove_shield()`, которые будут добавлять и отнимать у героя силу защитного поля к его защите.
 - Функция `equip_shield()` должна проверить, что защитное поле ещё не активировано, чтобы защита героя не прибавлялась бесконечно.
 - Функция `remove_shield()` должна проверить, что защитное поле активировано, чтобы защита героя не уменьшалась бесконечно.
 - Для проверок добавьте герою ключ `has_shield` с булевым значением, которое будет переключаться в функции `equip_shield()` на `True`, а в `remove_shield()` - на `False`.
- Вызывайте функцию `equip_shield()`, когда герой защищается.
- Вызывайте функцию `remove_shield()` после хода робота.
- Добавьте вывод **защиты** героя в функцию `display_hero_info()`.

Этап 4, бонусный

Этот этап состоит из двух дополнительных заданий по **+5 баллов** каждое. Прежде, чем браться за 4 этап, выполните как можно лучше задания предыдущих этапов.

Общие указания:

- Выносите код, описывающий действия робота и героя, в отдельные функции там, где это целесообразно.
- Не забывайте про информативные сообщения о действиях героя и робота и их результатах.

Задание 1

Добавьте роботу возможность "Бросить отравляющую гранату" (**+5 баллов**).

Робот может с вероятностью 25% бросить гранату. Граната наносит удвоенный урон от оружия робота, и отравляет героя прямо сквозь броню (защиту), но герой может защититься от нее используя защитное поле. Если робот бросил гранату больше он ничего не делает в свой ход.

- Урон = оружие робота * 2
- Если герой защищается в этот ход, то он отражает гранату защитным полем и не получает урона (урон = 0).

Указания:

- Выведите сообщение, когда робот кидает гранату.
- Выведите сообщение, когда герой отражает атаку робота защитным полем.

Задание 2

Добавьте герою действие "вколоть адреналин" (**+5 баллов**).

У героя в рюкзаке завалялись несколько шприцов с адреналином (по умолчанию только 1), которые он может вколоть в бою, чтобы подлечиться.

- Каждый шприц с адреналином вылечивает 500 единиц здоровья (изменяйте это число по своему желанию).
- Использование шприца с адреналином считается за ход. Герой не может атаковать или защищаться в этот ход.
- Если шприцы с адреналином кончились, игрок должен выбрать другое действие или пропустить ход (на ваш выбор, оба варианта правильные).
- Добавьте вывод информации о шприцах с адреналином в функцию `display_hero_info()`.

Указания:

- Используйте функцию `modify_health` для применения эффекта от вкалывания адреналина.
- Храните информацию о количестве и силе шприцев с адреналином в данных героя.