

Witaj w Obiektowym Obozowisku!

Jest to kraina, w której wiedzie się spokojne, powolne życie...

1. Mieszkaniec wioski

- Stwórz klasę mieszkańca wioski - Villager. Każdy mieszkaniec nosi własne imię (name) oraz posiada konkretny wiek (age). Zastanów się jakich konstruktorów/konstruktorów będzie potrzebowała klasa Villager i stwórz odpowiednie.
- Każdy z mieszkańców wioski potrafi się przedstawić w sposób - Greetings traveler... I'm Staszek and I'm 52 years old, gdzie Staszek to imię osadnika, a 52 to jego wiek. Utwórz metodę sayHello() klasy Villager drukującą odpowiedni komunikat na ekranie.
- W klasie Main w metodzie main utwórz 7 osadników.
 - Kashya 30 lat,
 - Akara 40 lat,
 - Gheed 50 lat,
 - Deckard Cain 85 lat,
 - Warriv 35 lat,
 - Flawia 25 lat.
- Wywołaj metodę sayHello() dla każdego z mieszkańców wioski.

2. Niezwykły mieszkaniec wioski.

Okazuje się, że niektórzy osadnicy posiadają niezwykle umiejętności przydatne dla Wędrowców. Zwróć uwagę, że klasa Villager nie posiada odpowiedniego pola do zaprezentowania tych umiejętności.

- Stwórz klasę ExtraordinaryVillager dziedziczącą po klasie Villager.
- Stwórz wewnętrzny publiczny enum Skill w klasie ExtraordinaryVillager z dostępnymi umiejętnościami.
 - IDENTIFY("I will identify items for you at no charge.")
 - SHELTER("I can offer you poor shelter.")
- Dodaj pole Skill skill w klasie ExtraordinaryVillager.
- Zmodyfikuj odpowiednio konstruktor, tak aby przyjmował jako kolejny argument utworzony enum.
- Zmień typ obiektów reprezentujących Deckarda Cain'a oraz Akarę na ExtraordinaryVillager.
- Nadpisz metodę sayHello() w klasie ExtraordinaryVillager, tak aby drukowany komunikat mówił również o posiadanej umiejętności. Tak, aby:
 - deckardCain.sayHello() drukowało Greetings traveler... I'm Deckard Cain and I'm 85 years old. I will identify items for you at no charge.
 - akara.sayHello() drukowało Greetings traveler... I'm Akara and I'm 40 years old. I can offer you poor shelter.

3. Potwory atakują!

Nasze spokojne obozowisko atakują potwory!

- Utwórz interfejs Fighter posiadający metody:
 - void attack(Fighter victim);
 - void takeHit(int damage);

- Utwórz klasę abstrakcyjną `Monster` implementującą interfejs `Fighter`.
Klasa ma posiadać pola
 - `int health` - pomyśl nad odpowiednim zakresem widoczności pola. Zastanów się, czy przydatny będzie tu konstruktor inicjalizujący tę wartość.
 - `int damage` - jak wyżej
 - Utwórz niezbędny getter/gettery.
- Utwórz klasę `Monsters`. W klasie mają się znaleźć dwa potwory (instancje klasy `Monster` w postaci anonimowych podklas).
 - `andariel` z życiem o wartości 10 oraz zadawanymi obrażeniami o wartości 70
 - `blacksmith` z życiem o wartości 100 oraz zadawanymi obrażeniami o wartości 25
 Potwory mają zostać utworzone jako obiekty `static final` za pomocą klas anonimowych. Klasa ma zawierać statyczne pole `monstersHealth` zawierające sumę punktów życia `Andariel` i `Blacksmith'a`.

Wywołanie metody `takeHit(damage)` dla:

- `andariel` - ma zmniejszać życie obiektu o `damage` oraz zmniejszać `monstersHealth` o `damage`.
- `blacksmith` - ma zmniejszać życie obiektu o `5 + damage` oraz zmniejszać `monstersHealth` o tę samą wartość

Wywołanie metody `attack(victim)` ma powodować zadanie obrażeń obiektowi `victim` o wartości `damage` aktualnego obiektu. Utwórz gettery dla pól.

4. Przygotuj osadników!

- Klasa `Villager`
 - Zaimplementuj interfejs `Fighter`.
 - Dodaj pole `health` reprezentujące punkty życia. Każdy osadnik ma mieć ich 100.
 - Ilość zadawanych obrażeń ma wynikać ze wzoru $(100 - \text{age} * 0.5) / 10$. Wzoru nie można modyfikować. Jeśli zajdzie konieczność należy zastosować rzutowanie. Utwórz niezbędne gettery.
- Klasa `ExtraordinaryVillager`
 - Nadpisz metody:
 - `void attack(Fighter victim);`
 - `void takeHit(int damage);`
 Tak, aby `ExtraordinaryVillager` nie zadawał żadnych obrażeń i ginął od jednego ciosu.
- Ukryj `ExtraordinaryVillager'ów` jako obiekty, na zasadzie:
 - `Object objectDeckardCain = deckardCain;`
 - `Object objectAkara = akara;`
 Czy na `objectDeckardCain` lub `objectAkara` można wywołać metody klasy `ExtraordinaryVillager`?

5. Obroń osadę!

- Przeprowadź epicką bitwę, która kończy się gdy suma żyć potworów jest mniejsza, bądź równa 0.
- W pętli `while()` atakuj naprzemiennie osadnik -> potwór, potwór -> osadnik, osadnik -> potwór, potwór -> osadnik... Możesz dodać dodatkowe metody w celu uniknięcia duplikacji kodu. Nie atakuj martwych jednostek (`health <= 0`). Możesz dodać dodatkowe pole w klasach informujące o tym, że jednostka jest już martwa. W każdym obiegu pętli informuj, o tym ile życia pozostało potworom wyświetlając komunikat:
 - Potwory posiadają jeszcze <ilość punktów życia potworów> punkty życia oraz o imieniu aktualnie walczącego osadnika wyświetlając komunikat:
 - Aktualnie walczący osadnik to <imię osadnika> Dodaj niezbędne gettery.
- Po zwycięskiej walce wypisz na ekran Obozowisko ocalone!

- Na koniec (po zakończonej walce) spraw aby "obiektowi" ExtraordinaryVillager'owie odzyskali swe moce przez zrzutowanie ich na klasę ExtraordinaryVillager, na zasadzie:
`deckardCain = (ExtraordinaryVillager) objectDeckardCain;`
- Sprawdź czy można wywołać metody z klasy ExtraordinaryVillager.