**შესავალი:**

მისტერ ბერნსს თითქმის არაფერი აინტერესებს ... მხოლოდ საკუთარი თავი და მისი ფული. ამრიგად, როდესაც მისი ელექტროსადგურიდან რადიოაქტიური ნივთიერება გაიჟონა, რამაც ველური ბუნების ფაუნის რამდენიმე სახეობის სიცოცხლე შეიწირა. იგი ძალზედ შეშფოთდა იმის გამო თუ როგორ აღიქვამდა ამას საზოგადოება, რადგან საზოგადოების აზრი მის შემოსავალზე იმოქმედებდა, გაჩნდა ჭორები გამოჩნდა სპრინგფილდის ბირთვული ელექტროსადგურის გარშემო. ერთ-ერთი მათგანია შეშფოთება შეეხებოდა Springfield- ის იშვიათი მდინარის იხვნისკარტების მუტაცისს შესაძლო შემთხვევებს. ველურ გარემოში ხომ თითქმის 50-ზე ცოტა მეტი დარჩა, გაისმა გადაკრული სიტყვა "გადაშენება". ამრიგად, მედიის შესაჩერებლად, მისტერ ბერნსსმა 30 მათგანი ჩაიგდო ხელში რომ გაეანალიზებინა მუტაციის ნიშნები. მან აღმოაჩინა, რომ მუტაციის მაჩვენებელი ყოველთვიურად 2% -ს შეადგენს, მაგრამ როდესაც მუტაცია ხდება, ეს იწვევსიმას რომ იხვნისკარტები სტერილური ხდებიან და ვერ ახდენენ რეპროდუცირებას. არსებული ინფორმაციის საშუალებით, მას სურს შექმნას კომპიუტერული სიმულაციები, რომ პრესას ძალიან უყვარს. აქ იწყება თქვენი დავალება!

**სპეციფიკაციები:** ამ დავალებაში თქვენ შექმნით კლასს, რომელსაც ეწოდება platypus . ქვემოთ, ჩვენ აღწერს, თუ რა განსაზღვრავს platypus. თქვენ ასევე შექმნით main ფუნქციას, რომელშიც შექმნით platypus ტიპის ობიექტებს თქვენი ახალი მომხმარებლის მიერ განსაზღვრული ტიპის ფუნქციონირების შესამოწმებლად. სხვათა შორის, ამ ტიპის მაგისტარს დრაივერი ეწოდება, რადგან იგი გამოიყენება მხოლოდ რაღაცის (ახალი) შესამოწმებლად. ამ დრაივერში, თქვენ შექმნით თქვენი ახალი ტიპის (ებ) ის საკმარის ობიექტებს, რათა ადეკვატურად შეამოწმოთ მათი ფუნქციები. თუროგორ მოახერხებთ ამას - თქვენი მოსაფიქრებელ/გადასაწყვეტია.

თქვენი პლატიპუსის კლასი შეიცავს შემდეგ

**წევრ მონაცემებს(ცვლადებს):**

* + float weight
  + short age (months)
  + char name (initial)
  + char gender
  + bool alive (ცოცხალია თუ მკვდარი)
  + bool mutant (მუტანტია თუ არა)

**წევრ ფუნქციებს:**

* + უპარამეტრო კონსტრუქტორი (default constructor) რომელიც ქმნის მკვდარ იხვნისკარტას.
  + პარამეტრიანი კონსტრუქტორი რომელსაც გადასცემთ მნიშვნელობებს იმისათვის რომ განსაზღვროთ ობიექტის gender, weight, age, და name ცვლადების მნიშვნელობები, აგრეთვე default-ად ობიექტი უნდა იყოს alive(ცოცხალი) და not mutant(არა-მუტანტი).
  + ფუნქცია print რომელიც გამოიტანს იხვნისკარტას შესახებ ინფორმაციას ლამაზი და ადვილად აღსაქმელი ფორმითა და ფორმატით.
  + ფუნქცია age\_me რომელიც არაფერს აბრუნებს, მაგრამ ცვლის ობიექტის ასაკს. მასში უნდა იყოს გათვალისწინებული 2% შანსი იმისა რომ ობიექტი შესაძლოა გახდეს მუტანტი. ყოველ-ჯერზე age\_me ფუნქციის გამოძახებისას ობიექტს აქვს შანსი რომ მოკვდეს და ეს შანსი ტოლია 10-ჯერ იხვნისკარტის წონისა, ანუ 5 კგ იხვნისკარტას აქვს 50% შანსი რომ მოკვდეს. 10 და მეტ კილოიანი იხნისკარტას სიკვდილის ალბათობა 100%-ია.
  + ფუნქცია fight- რომელიც პარამეტრად იღებს სხვა პლატიპუსს. რომელიც ახდენს იმ იხვნისკარტას შებრძოლებას რომელზეც ხდება ფუნქციის გამოძახება (calling\_platypus ) მეორესთან რომელიც პარამეტრად გადაეცემა (other\_platypus). გამარჯვებული ვლინდება ძალების შეფარდებიდან ( "fight ratio") გამომდინარე (calling\_platypus წონა /other\_platypus წონა) \* 50. აიღება შემთხვევითი რიცხვი 1 დან 100-მდე. თუ ძალთა შეფარდება მეტია ამ შემთხვევით რიცხვზე calling\_platypus - იმარჯვებს, თუ არადა გამარჯვებულია მეორე.
  + ფუნქცია eat ზრდის იხვნისკარტას წონას რამდომად მისი მიმდინარე წონის 0.1% - 5.0% -ით.
  + ფუნქცია hatch(ახალშობილი) რომელიც ანიჭებს ახალდაბადებულ პლატიპუსს შემდეგ პარამეტრებს alive=true, mutant=false, და age=0. სქესი 'm' ან 'f' ერთნაირი ალბათობით. წონა რანდომად 0.1 დან - 1.0 კგ. სახელი მომხმარებელს თავად შეუძლია განუსაზღვროს ან შესაძლებელია სახელი იყოს რაიმე შემთხვევითად განსაზღვრული სიმბოლო (character chosen randomly).

**შენიშვნა:** ძალიან კარგად დაფიქრდით ზემოთ ჩამოთვლილი ფუნქციების დაწერის დროს იმაზე თუ როგორ უნდა იქნას გამოყენებულნი ისინი. ნამდვილად არსებობს გარემოებები, როდესაც ზოგიერთი ფუნქცია არ უნდა შესრულდეს. მაგალითად, მკვდარი იხვნისკარტა არაფერს ჭამს.

**კლასის ტესტირება:** თქვენს დრაივერ პროგრამაში უნდა გატესტოთ თქვენი პლატიპუსის კლასი სრულყოფილად. მასში უნდა გამოიძახოთ ყველა წევრ-ფუნქცია. ეკრანზე უნდა გამოიტანოთ რას აკეთებს და ამ ცვლილებების შედეგად რა ხდება, რა მოსდის შესაბამის ობიექტს(იხვნისკარტას/პლატიპუსს). ბრძოლის ფუნქციას დასჭირდება ორი პლატიპი: ერთი რომელზეც უნდა გამოიძახოთ საბრძოლო ფუნქცია და მეორე უნდა გადასცეთ როგორც პარამეტრი ამ ფუნქციისა.

შეფასებული იქნებება კონსტანტების გამოყენების ადეკვატურობა:ფუნქციების პარამეტრების რომლებიც არ შეცვლის ფუნქციას, უნდა იყოს გამოცხადებული const-ის გამოყენებით (გაითვალისწინეთ კლასი დრო(Time)-ს დიზაინი). დაფიქრდით მუტაციის პარამეტრის კონსტანტად გამოცხადებაზე, თუ ის შემდგომ შეიცვლება, კოდის გადაკეთება უფრო მარტივი იქნება...გარდა ამისა, წევრის ცვლებს თან უნდა ახლდეს შესაბამისი გეთერები და სეთერები