

- 01 - SquareRectangle inherit Rectangle
- 02 - Animal part 1
- 03 - Animal part 2
- 04 - Animal part 3
- 05 - Animal part 4

01 SquareRectangle inherit Rectangle

Recent submission on 16:52 17 Mar 2020 Result: PASSED [PPPPPP]

Current   Recent   All

SquareRectangle inherit Rectangle

สี่เหลี่ยมผืนผ้า (Rectangle) คือ สี่เหลี่ยมที่มีมุมภายในทุกมุมเป็นมุมฉาก ประกอบด้วยด้านสูง (height) และด้านกว้าง (width)

เราสามารถสร้างคลาส Rectangle เพื่อแทนสี่เหลี่ยมมุมฉาก ที่เก็บข้อมูลด้านสูงและด้านกว้างของสี่เหลี่ยม และมี method การหาพื้นที่ (area) และความยาวรอบรูป (circumference)

สี่เหลี่ยมจัตุรัส (Square Rectangle) คือ สี่เหลี่ยมผืนผ้าที่มีด้านสูงเท่ากับด้านกว้าง

ดังนั้น เราสามารถสร้างคลาส SquareRectangle เพื่อแทนสี่เหลี่ยมจัตุรัส ที่สืบทอดคุณลักษณะและพฤติกรรมของสี่เหลี่ยมมุมฉากได้

จงเติมส่วนของโปรแกรมด้านล่าง ให้ SquareRectangle สืบทอดคุณลักษณะและพฤติกรรมของ Rectangle และ implement การทำงานของคลาสทั้งสอง ให้ทำงานอย่างถูกต้อง

[hide line #]

```
#include <iostream>
#include <cmath>
using namespace std;

class Rectangle
{
private:
    double width;
    double height;
public:
    Rectangle(double width, double height);
    double area();
    double circumference();
    double getWidth();
    double getHeight();
};

class SquareRectangle
{
private:
    double side;
public:
    SquareRectangle(double side);
    double getSide();
};

int main()
{
    int tc;
    double width, height;
    double side;
    Rectangle rectangle(0, 0);
    SquareRectangle sqRec(0);
    cin >> tc;
    switch (tc) {
        case 1:
            cin >> width;
            cin >> height;
            rectangle = Rectangle(width, height);
            cout << "Rectangle" << endl;
            cout << "Width: " << rectangle.getWidth() << endl;
            cout << "Height: " << rectangle.getHeight() << endl;
            cout << "Area: " << rectangle.area() << endl;
            cout << "Circumference: " << rectangle.circumference() << endl;
            break;

        case 2:
            cin >> side;
            sqRec = SquareRectangle(side);
            cout << "Square Rectangle" << endl;
            cout << "Width: " << sqRec.getWidth() << endl;
            cout << "Height: " << sqRec.getHeight() << endl;
            cout << "Side: " << sqRec.getSide() << endl;
            cout << "Area: " << sqRec.area() << endl;
            cout << "circumference: " << sqRec.circumference() << endl;
            break;
    }
}
```

Submit

- Lab 11:
- CPP 113 Lab 11
- 01 - SquareRectangle inherit Rectangle
- 02 - Animal part 1
- 03 - Animal part 2
- 04 - Animal part 3
- 05 - Animal part 4

02 Animal part 1

Recent submission on 22:47 18 Mar 2020 Result: FAILED [PPP---PPPP]

CurrentRecentAll

Class Animal

จงเขียน Implementation ของ Class Animal ซึ่งเป็นคลาสแทนสัตว์

Animal ประกอบด้วย attributes ดังนี้

- happiness คือ ค่าความสุข มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100
- energy คือ ค่าพลังงาน มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100
- fullness คือ ค่าความอึด มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100

Animal มี methods ต่อไปนี้

- Constructor ที่กำหนดค่าเริ่มต้นของ happiness, energy และ fullness ตามลำดับ
- limitStat() ใช้ควบคุมค่า happiness, energy และ fullness ไม่ให้เกินขอบเขตของช่วง 0 ถึง 100
- getter ของ attributes ทั้งสาม
- eat(int food) สัตว์มีพฤติกรรมการกินอาหาร โดยรับค่าพลังงานของอาหาร มาเพิ่ม fullness ตามจำนวนค่าพลังงานอาหารที่รับเข้ามา แต่ถ้าฝืนกินจนเกินค่าความอึดสูงสุด จะทำให้ความสุขลดลง 10 หน่วยด้วย
- play(int hour) สัตว์มีพฤติกรรมการเล่น โดยแต่ละครั้งสัตว์จะเล่นเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้พลังงานลดลง 10 หน่วยต่อชั่วโมง ทำให้ความอึดลดลง 20 หน่วยต่อชั่วโมง แต่ทำให้ค่าความสุขเพิ่มขึ้น 5 หน่วยต่อชั่วโมง แต่หากในครั้งนั้น ค่าพลังงาน หรือค่าความอึด เป็น 0 ก็จะไม่เล่น
- sleep(int hour) สัตว์มีพฤติกรรมการนอน โดยแต่ละครั้งสัตว์จะนอนเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้ค่าพลังงานเพิ่มขึ้น 10 หน่วยต่อชั่วโมง และค่าความอึดลดลง 10 หน่วยต่อชั่วโมง

[hide line #]

```
#include <iostream>
using namespace std;

class Animal
{
private:
    int happiness;
    int energy;
    int fullness;
    const int MAX_HAPPINESS = 100;
    const int MAX_ENERGY = 100;
    const int MAX_FULLNESS = 100;
    void limitStat();
public:
    Animal(int, int, int);
    int getHappiness();
    int getEnergy();
    int getFullness();
    void eat(int);
    void play(int);
    void sleep(int);
};

int main()
{
}
```

Submit



Lab 11:  
CPP 113 Lab 11

01 - SquareRectangle inherit Rectangle
02 - Animal part 1
03 - Animal part 2
04 - Animal part 3
05 - Animal part 4

04 Animal part 3

CurrentRecentAll

Class Pet inherit Animal

จงเขียน Implementation ของ Class Animal ซึ่งเป็นคลาสแทนสัตว์ และ Class Pet ซึ่งเป็นคลาสแทนสัตว์เลี้ยง

Animal ประกอบด้วย attributes ดังนี้

- happiness คือ ค่าความสุข มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100
- energy คือ ค่าพลังงาน มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100
- fullness คือ ค่าความอึด มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100

Animal มี methods ต่อไปนี้

- Constructor ที่กำหนดค่าเริ่มต้นของ happiness, energy และ fullness ตามลำดับ
- limitStat() ใช้ควบคุมค่า happiness, energy และ fullness ไม่ให้เกินขอบเขตของช่วง 0 ถึง 100
- getter ของ attributes ทั้งสาม
- eat(int food) สัตว์มีพฤติกรรมกินอาหาร โดยรับค่าพลังงานของอาหาร มาเพิ่ม fullness ตามจำนวนค่าพลังงานอาหารที่รับเข้ามา แต่ถ้าฝืนกินจนเกินค่าความอึดสูงสุด จะทำให้ความสุขลดลง 10 หน่วยด้วย
- play(int hour) สัตว์มีพฤติกรรมการเล่น โดยแต่ละครั้งสัตว์จะเล่นเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้พลังงานลดลง 10 หน่วยต่อชั่วโมง ทำให้ความอึดลดลง 20 หน่วยต่อชั่วโมง แต่ทำให้ค่าความสุขเพิ่มขึ้น 5 หน่วยต่อชั่วโมง แต่หากในครั้งนั้น ค่าพลังงาน หรือค่าความอึด เป็น 0 ก็จะไม่เล่น
- sleep(int hour) สัตว์มีพฤติกรรมการนอน โดยแต่ละครั้งสัตว์จะนอนเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้ค่าพลังงานเพิ่มขึ้น 10 หน่วยต่อชั่วโมง และค่าความอึดลดลง 10 หน่วยต่อชั่วโมง

Pet ประกอบด้วย attributes ที่สืบทอดจากสัตว์ และมี attribute ต่อไปนี้เพิ่มจากสัตว์

- name ชื่อของสัตว์ เนื่องจากเจ้าของสัตว์เลี้ยง มักจะตั้งชื่อให้สัตว์เลี้ยงด้วย

Pet มีพฤติกรรมเดียวกับสัตว์ และเพิ่มพฤติกรรมต่อไปนี้

- Constructor ของ Pet จะระบุ name, happiness, energy และ fullness เป็นค่าเริ่มต้น ตามลำดับ
- getter และ setter ของ name

[hide line #]

```
#include <iostream>
using namespace std;

class Animal
{
protected:
    int happiness;
    int energy;
    int fullness;
    const int MAX_HAPPINESS = 100;
    const int MAX_ENERGY = 100;
    const int MAX_FULLNESS = 100;
    void limitStat();
public:
    Animal(int, int, int);
    int getHappiness();
    int getEnergy();
    int getFullness();
    void eat(int);
    void play(int);
    void sleep(int);
};

class Pet
{
private:
    string name;
public:
    Pet(string, int, int, int);
    string getName();
    void setName(string);
};



int main()
{
}
```

Submit

Icons from Sweetie.



Lab 11:  
CPP 113 Lab 11

01 - SquareRectangle inherit Rectangle 
02 - Animal part 1 
03 - Animal part 2
04 - Animal part 3
05 - Animal part 4

05 Animal part 4

CurrentRecentAll

Class OwnedBird multiple inheritance

จงเขียน Implementation ของ Class Animal ซึ่งเป็นคลาสแทนสัตว์

Implementation ของ Class Bird ซึ่งเป็นคลาสของนกที่สืบทอดคุณสมบัติและพฤติกรรมของสัตว์

Implementation ของ Class Pet ซึ่งเป็นคลาสของความเป็นสัตว์เลี้ยง (ตัดการสืบทอดจากสัตว์ออกไป)

และ Implementation ของ Class OwnedBird ซึ่งเป็นคลาสของนกที่มีเจ้าของ คลาสนี้สืบทอดคุณสมบัติและพฤติกรรมจากนกและ ความเป็นสัตว์เลี้ยง

Animal ประกอบด้วย attributes ดังนี้

•

happiness คือ ค่าความสุข มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100

•

energy คือ ค่าพลังงาน มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100

•

fullness คือ ค่าความอึด มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง 100

Animal มี methods ต่อไปนี้

•

Constructor ที่กำหนดค่าเริ่มต้นของ happiness, energy และ fullness ตามลำดับ

•

limitStat() ใช้ควบคุมค่า happiness, energy และ fullness ไม่ให้เกินขอบเขตของช่วง 0 ถึง 100

•

getter ของ attributes ทั้งสาม

•

eat(int food) สัตว์มีพฤติกรรมการกินอาหาร โดยรับค่าพลังงานของอาหาร มาเพิ่ม fullness ตามจำนวนค่าพลังงานอาหารที่ได้รับเข้ามา แต่ถ้าฝืนกินจนเกินค่าความอึดสูงสุด จะทำให้ความสุขลดลง 10 หน่วยด้วย

•

play(int hour) สัตว์มีพฤติกรรมการเล่น โดยแต่ละครั้งสัตว์จะเล่นเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้พลังงานลดลง 10 หน่วยต่อชั่วโมง ทำให้ความอึดลดลง 20 หน่วยต่อชั่วโมง แต่ทำให้ความสุขเพิ่มขึ้น 5 หน่วยต่อชั่วโมง แต่หากในครั้งนั้น ค่าพลังงาน หรือค่าความอึด เป็น 0 ก็จะไม่เล่น

•

sleep(int hour) สัตว์มีพฤติกรรมการนอน โดยแต่ละครั้งสัตว์จะนอนเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้ค่าพลังงานเพิ่มขึ้น 10 หน่วยต่อชั่วโมง และค่าความอึดลดลง 10 หน่วยต่อชั่วโมง

Bird มี attributes และ methods ที่สืบทอดจาก Animal และมี methods เพิ่มเติมดังนี้

•

fly(int hour) นกมีพฤติกรรมการบิน โดยการบินแต่ละครั้งจะบินเป็นจำนวนชั่วโมงติดต่อกัน ซึ่งแต่ละครั้งจะทำให้ค่าพลังงานลดลง 5 หน่วยต่อชั่วโมง

•

sing() นกมีพฤติกรรมการร้องเพลง โดยการร้องแต่ละครั้งจะทำให้ความสุขเพิ่มขึ้น 5 หน่วย

Pet ประกอบด้วย attribute name

•

name ชื่อของสัตว์ เนื่องจากเจ้าของสัตว์เลี้ยง มักจะตั้งชื่อให้สัตว์เลี้ยงด้วย

Pet มีพฤติกรรมต่อไปนี้

•

Constructor ของ Pet จะระบุ name เป็นค่าเริ่มต้น

•

getter และ setter ของ name

OwnedBird สืบทอด attributes, methods จาก Bird และ Pet และมีข้อกำหนดของ Constructor ดังนี้

•

Constructor จะระบุ name, happiness, energy และ fullness เป็นค่าเริ่มต้น ตามลำดับ

#include <iostream>

using namespace std;

class Animal

{

protected:

int happiness;

int energy;

int fullness;

const int MAX\_HAPPINESS = 100;

const int MAX\_ENERGY = 100;

const int MAX\_FULLNESS = 100;

void limitStat();

public:

Animal(int, int, int);

int getHappiness();

int getEnergy();

int getFullness();

void eat(int);

void play(int);

void sleep(int);

};

class Bird

{

public:

Bird(int, int, int);

void fly(int);

void sing();

};

class Pet

{

protected:

string name;

public:

Pet(string);

string getName();

void setName(string);

};

class OwnedBird

{

public:

OwnedBird(string, int, int, int);

};

```
int main()
{
    char action;
    int hour;
    int happiness, energy, fullness;
    cin >> happiness >> energy >> fullness;
    OwnedBird animal("Dobby", happiness, energy, fullness);
    while (1) {
        cin >> action;
        if (action == 'q') break;
        cin >> hour;
        switch (action) {
            case 'e':
                animal.eat(hour);
                break;
            case 'p':
                animal.play(hour);
                break;
            case 's':
                animal.sleep(hour);
                break;
            case 'f':
                animal.fly(hour);
                break;
            case 'i':
                for (int i = 0; i < hour; i++)
                    animal.sing();
                break;
            case 'n':
                if (hour < 5)
                    animal.setName("Debby");
                else
                    animal.setName("Shogun");
                break;
        }
        test(animal);
    }
}
```

Submit

Icons from Sweetie.