

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту

Кафедра математичного моделювання та аналізу даних

КОНТРОЛЬНА РОБОТА №1

на тему: «»

Виконав: студент 3 курсу групи НОМЕР
КС32

Спеціальності 122 «Комп'ютерні науки»

ПІБ Бабенко Андрій Андрійович

Варіант 13

Теоретичні питання

1. Як видалити елемент з масиву у Ruby?

`delete` видаляє всі елементи, що дорівнюють значенню `value`.

```
array = [1, 2, 3, 4, 2]
array.delete(2)
# => [1, 3, 4]
```

2. Що таке масиви масивів (двовимірні масиви) у Ruby? Як їх використовувати?

Масиви масивів — це двовимірні масиви або масиви, де кожен елемент є іншим масивом.

```
matrix = [
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9]
]
```

Для доступу до елементів використовуються подвійні індекси

```
puts matrix[0][1] # => 2 (елемент з першого рядка і другого стовпця)
```

```
puts matrix[2][0] # => 7 (елемент з третього рядка і першого стовпця)
```

3. Як реалізується множинне спадкування за допомогою модулів у Ruby?

```
module Flyable
  def fly
    puts "I can fly!"
  end
end
```

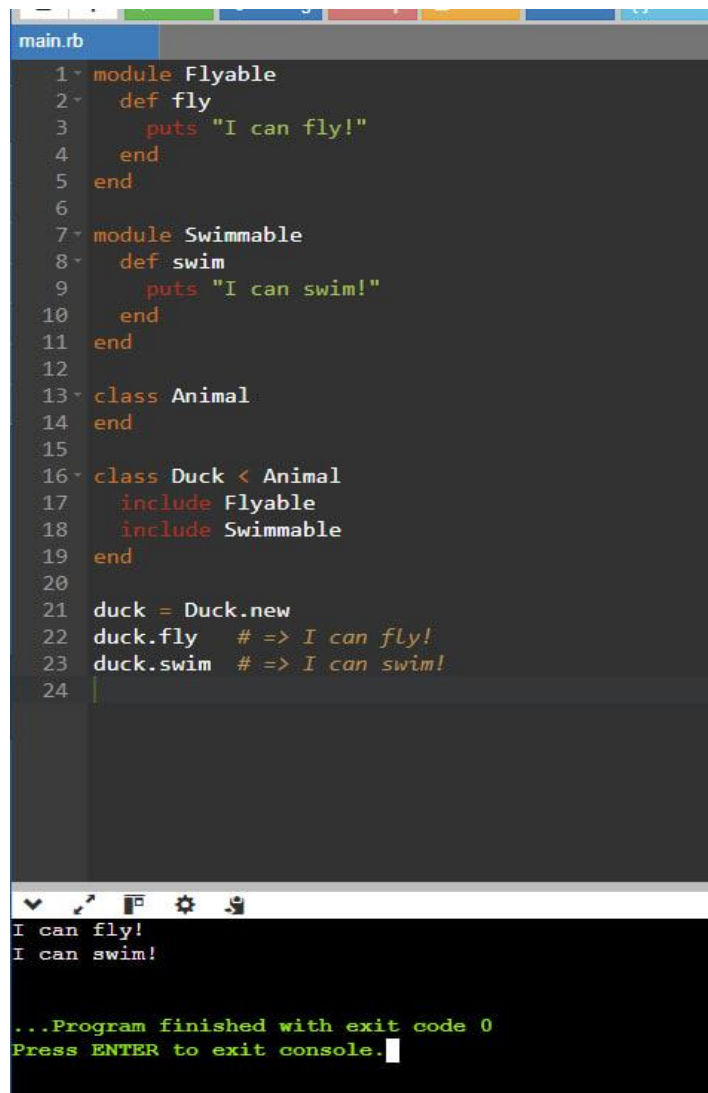
```
module Swimmable
  def swim
    puts "I can swim!"
  end
end

class Animal
end

class Duck < Animal
  include Flyable
  include Swimmable
end

duck = Duck.new
duck.fly # => I can fly!
duck.swim # => I can swim!
```

Лістинг 1 – вхідний код



The image shows a screenshot of a Ruby script execution. The top part is a code editor window titled 'main.rb' showing the following code:

```
1 module Flyable
2   def fly
3     puts "I can fly!"
4   end
5 end
6
7 module Swimmable
8   def swim
9     puts "I can swim!"
10  end
11 end
12
13 class Animal
14 end
15
16 class Duck < Animal
17   include Flyable
18   include Swimmable
19 end
20
21 duck = Duck.new
22 duck.fly # => I can fly!
23 duck.swim # => I can swim!
24
```

The bottom part is a terminal window showing the output of the script:

```
I can fly!
I can swim!

...Program finished with exit code 0
Press ENTER to exit console.
```

Малюнок 1 – результат

У цьому прикладі клас `Duck` успадковує поведінку з обох модулів `Flyable` та `Swimmable`.

4. Що таке інстансні методи та змінні? Як вони використовуються у Ruby?

Інстансні методи — це методи, які доступні на рівні окремого об'єкта класу. Вони визначають поведінку конкретного об'єкта. Для їх виклику потрібно створити об'єкт класу.

```
class Dog

  def bark

    puts "Woof!"

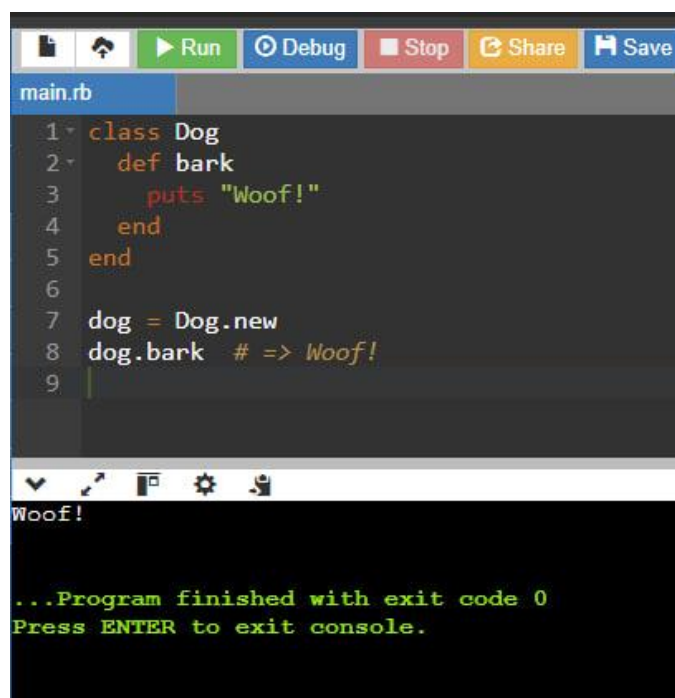
  end

end

dog = Dog.new

dog.bark #=> Woof!
```

Лістинг 2 – вхідний код програми



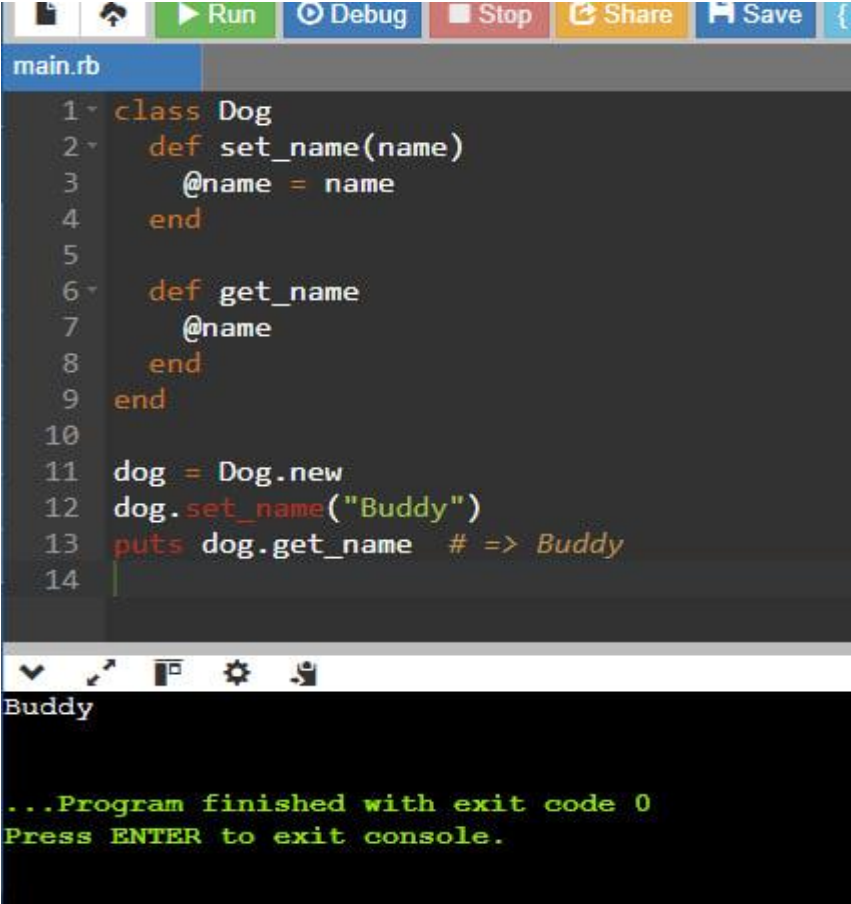
```
main.rb
1 class Dog
2   def bark
3     puts "Woof!"
4   end
5 end
6
7 dog = Dog.new
8 dog.bark #=> Woof!
9

Woof!

...Program finished with exit code 0
Press ENTER to exit console.
```

Малюнок 2 – результат

Інстансні змінні — це змінні, які доступні тільки для конкретного об'єкта і мають префікс @



```
main.rb
1 class Dog
2   def set_name(name)
3     @name = name
4   end
5
6   def get_name
7     @name
8   end
9 end
10
11 dog = Dog.new
12 dog.set_name("Buddy")
13 puts dog.get_name # => Buddy
14

Buddy

...Program finished with exit code 0
Press ENTER to exit console.
```

Малюнок 3 – результат

Використовуються для зберігання стану об'єкта.

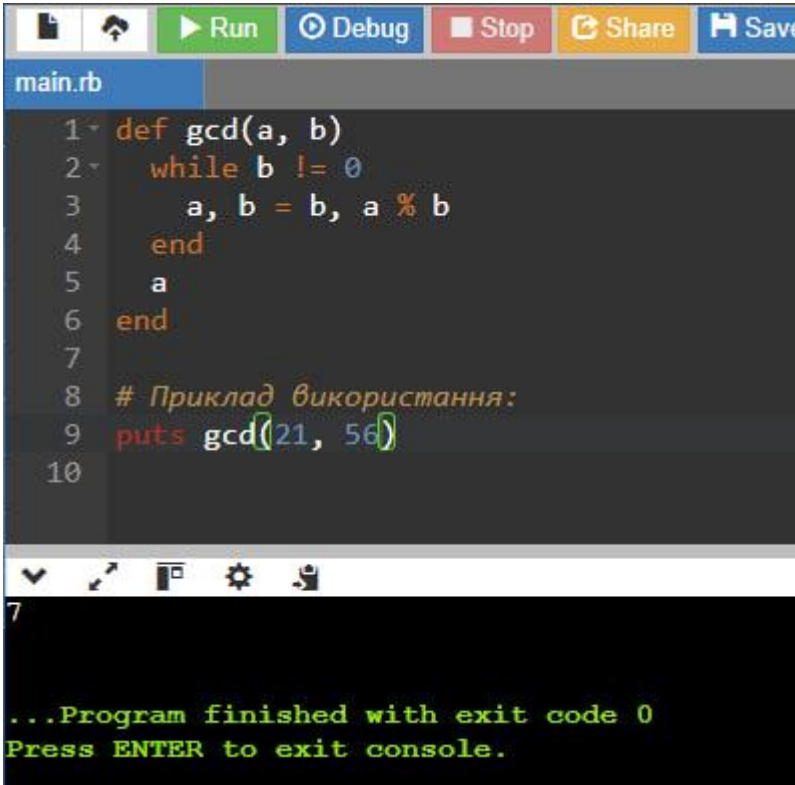
Практичні завдання

1. Напишіть метод, який знаходить найбільший спільний дільник (НСД) двох чисел.

```
def gcd(a, b)
  while b != 0
    a, b = b, a % b
  end
  a
end
```

```
# Приклад використання:
puts gcd(48, 56)
```

Лістинг 1 – вхідний код програми



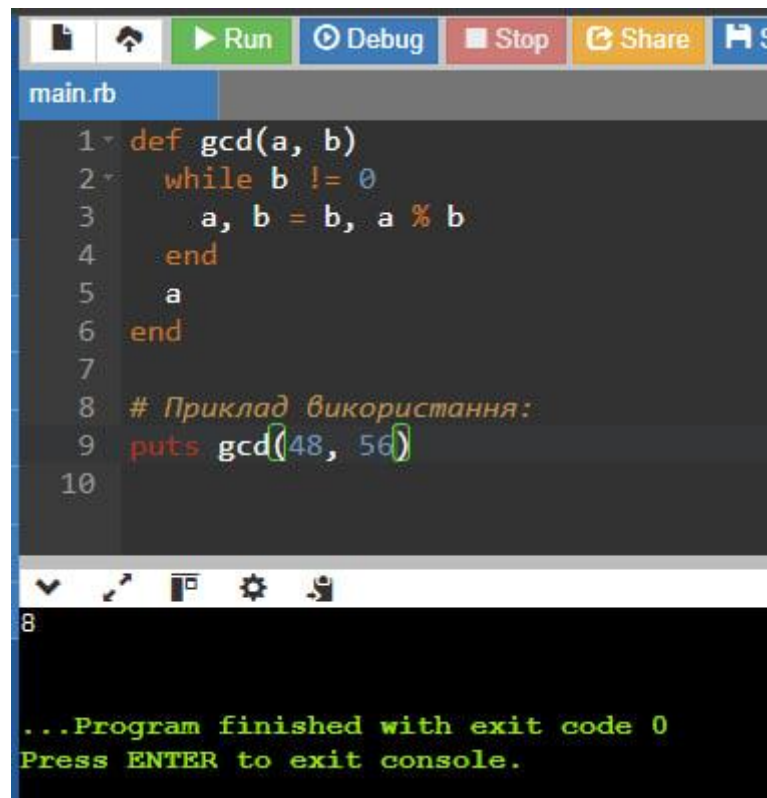
The image shows a screenshot of a Ruby IDE. The top toolbar includes icons for file operations and buttons for 'Run', 'Debug', 'Stop', 'Share', and 'Save'. The editor window, titled 'main.rb', contains the following code:

```
1 def gcd(a, b)
2   while b != 0
3     a, b = b, a % b
4   end
5   a
6 end
7
8 # Приклад використання:
9 puts gcd(21, 56)
10
```

Below the editor is a console window showing the output of the program:

```
7
...Program finished with exit code 0
Press ENTER to exit console.
```

Малюнок 1



The screenshot shows a Ruby IDE window titled 'main.rb'. The code defines a `gcd` function using a `while` loop and the modulo operator. It then calls `puts gcd(48, 56)`. The console output shows the result '8' and a message indicating the program finished with exit code 0.

```
main.rb
1 def gcd(a, b)
2   while b != 0
3     a, b = b, a % b
4   end
5   a
6 end
7
8 # Приклад використання:
9 puts gcd(48, 56)
10

8

...Program finished with exit code 0
Press ENTER to exit console.
```

Малюнок 2
Малюнок (1,2) – результати програми

2. Створіть клас `Student`, що містить ім'я студента і список його оцінок.
Додайте метод для обчислення середнього балу.

```
class Student
  attr_accessor :name, :grades

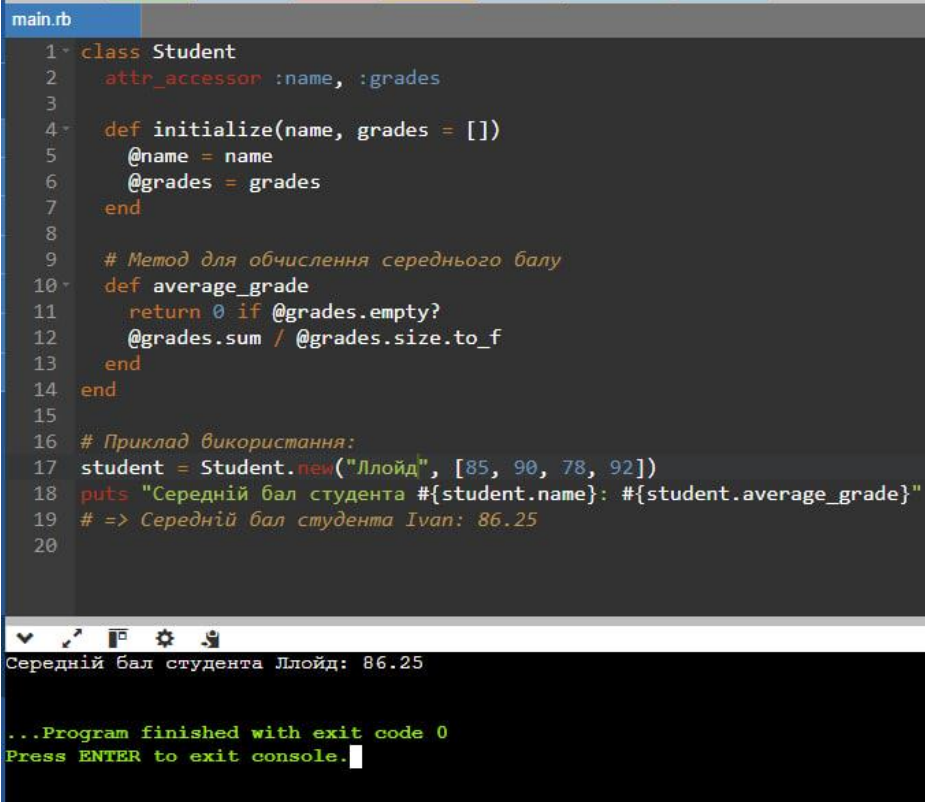
  def initialize(name, grades = [])
    @name = name
    @grades = grades
  end

  # Метод для обчислення середнього балу
  def average_grade
    return 0 if @grades.empty?
    @grades.sum / @grades.size.to_f
  end
end

# Приклад використання:
student = Student.new("Ентоні", [50, 90, 58, 72])
```

```
puts "Середній бал студента #{student.name}: #{student.average_grade}"  
# => Середній бал студента Ivan: 86.25
```

Лістинг 2 – вхідний код програми



```
main.rb  
1 class Student  
2   attr_accessor :name, :grades  
3  
4   def initialize(name, grades = [])  
5     @name = name  
6     @grades = grades  
7   end  
8  
9   # Метод для обчислення середнього балу  
10  def average_grade  
11    return 0 if @grades.empty?  
12    @grades.sum / @grades.size.to_f  
13  end  
14 end  
15  
16 # Приклад використання:  
17 student = Student.new("Ллойд", [85, 90, 78, 92])  
18 puts "Середній бал студента #{student.name}: #{student.average_grade}"  
19 # => Середній бал студента Ivan: 86.25  
20  
Середній бал студента Ллойд: 86.25  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Малюнок 3


```
main.rb
1 class Student
2   attr_accessor :name, :grades
3
4   def initialize(name, grades = [])
5     @name = name
6     @grades = grades
7   end
8
9   # Метод для обчислення середнього балу
10  def average_grade
11    return 0 if @grades.empty?
12    @grades.sum / @grades.size.to_f
13  end
14 end
15
16 # Приклад використання:
17 student = Student.new("Ентоні", [50, 90, 58, 72])
18 puts "Середній бал студента #{student.name}: #{student.average_grade}"
19 # => Середній бал студента Іван: 86.25
20
```

Середній бал студента Ентоні: 67.5

...Program finished with exit code 0
Press ENTER to exit console.

Малюнок 4
Малюнок(3,4) – результати програми